

What is Java?

- Java is a high-level programming language originally developed by Sun Microsystems and Firstly released in 1995. Java is the only one to **run on different platforms**, such as Windows, Mac , and UNIX.
- When a program runs , it requires an environment , which helps program to run, which is also called a platform. And **Java has its own runtime environment** or platform (JRE) and API.
- Java is the most popular programming language & is the **language of choice for Android programming WITH 9 MILLION DEVELOPERS WORKING ON IT.**

java सबसे populer language है यह एक high-level language है high level का मतलब हम simple english का use करते हुए java के program को develop कर सकते हैं java को सबसे पहले Sun Microsystems ने 1995 में इसे बनाया था java की सबसे बड़ी विशेषता यह है कि अगर java के program को एक बार type कर दे और अलग अलग operating system या platforms पर इसे चला सकते हैं यह program Windows, Mac, and UNIX सभी पर same तरह से चलेगी

java अलग अलग devices पर प्रयोग की जा सकती है जैसे कि हम android mobile का use करते हैं तो उसमे भी काफी सारे operating system को java के साथ बनाया गया है और काफी सारी apps java के साथ बनाई जाती है इसी तरह अगर Windows पर काम करते हैं तो काफी सारी application java के साथ काम करने के लिए इसमें भी मिलती है

जब एक program चलता है तो उसे run होने के लिए एक environment चाहिए होता है java पहले से ही बना हुआ environment हमे देता है जिसे उसका runtime environment कहते हैं JRE और कुछ API. API का अर्थ application programming interface से है काफी सारी files जो पहले से ही बनी होती है और हमे काम करने में मदद करती है

java most popular language रही है इस पर 9 मिलियन से ज्यादा developers इस पर आज भी काम कर रहे हैं इस तरह से java एक popular language है

Prerequisite of learning

- ❑ Better to have basic knowledge of C / C++ language.
- ❑ Knowledge of any basic text editor.
- ❑ Good to have knowledge of any compiling software .
- ❑ Basic internet knowledge to download JDK and other required software.

java language सीखने के लिए हम जिनके बारे में जानकारी होना जरूरी है हम उनके बारे में जानकारी लेंगे सबसे पहले यदि java से पहले किसी language को सीखा हुआ है तो वह हमारे लिए बहुत ही useful रहता है इन language का experience होना java सीखने में मदद करता है अगर C या C ++ की जानकारी नहीं है तो भी java को शुरू से सीख सकते हैं

इसके अतिरिक्त basic text editor की जानकारी होना जरूरी है ताकि java के program को आसानी से edit कर सके उसके लिए notepad का use कर सकते हैं

इसके लिए अगर compiling software की जानकारी हो तो वह भी useful रहता है

इसके अतिरिक्त internet की भी जानकारी होनी चाहिए ताकि हम आसानी से java compiler को या java development kits(JDK) को डाउनलोड कर पाए और अगर internet की जानकारी है तो नीचे available websites पर जाकर पहले से बने हुए programs को डाउनलोड कर सकते हैं जोकि java की excercises के रूप में दी गई है

Evolution OF JAVA IN LEVELS

“Java has various levels and with every level Java has included new features in it.”

1. Firstly name Oak and now called Java was developed by James Ghosling, Patrick Naughton and Mike Sheridan at Sun Microsystems Inc in 1991.
2. In the year 1996 Java Versions 1.0 and 1.1 was released .
3. In the year 1998 Java version 1.2 (also called java 2) was released.
4. in the year 2000 Java Version 1.3 was released.
5. in the year 2002 Java Version 1.4 was released.
6. in the year 2004 Java Version 1.5/Java SE 5 was released.
7. in the year 2006 Java Version 1.6/Java SE 6 was released.
8. In 2010 Sun Microsystems was acquired by Oracle.
9. in the year 2011 Java Version 1.7/Java SE 7 was released.
10. in the year 2015 Java Version 1.8 /Java SE 8 was released.
11. The latest version of Java is Version 8 With Updation and it was released in July 2016.
12. The next version Java 9 is scheduled to be released very soon by Oracle.



अब हम java के evolution के बारे में जानकारी लेंगे कि java किस तरह से विकसित हुई है जैसे जैसे java विकसित हुई इसमें new features भी जुड़ते चले गए

जब java को launch किया गया तो इसे 1991 में Sun Microsystems द्वारा launch किया गया जिसमें patrick naughton and mike sheridan का important रोल रहा है java का सबसे पहला name Oak था पर 1991 से 1996 तक जैसे जैसे इसमें changes किए गए इसी तरह काफी सारे version released हुए थे और अब जो latest version है वह java version 8 है जोकि सभी तरह की updation के साथ में Oracle के द्वारा july 2016 में released किया गया इसके बाद java का 9 version भी launch किया जाएगा अभी हम version 8 का use करेंगे

Benefits and features of JAVA

- Simple : If You know C/C++ , than it is **very simple** for you to learn java, because Java has almost similar concepts and syntax but with java's portable properties.
- Portable and Platform independent :- Java provides software-based platform and after code is compiled and Interpreted in byte codes, This BYTECODE is a platform-independent code because it can be run on different platforms and operating systems like Windows / Linux / Mac.
- Object-Oriented : With Object-oriented programming(OOPs) you can implement Object , Class , Inheritance , Polymorphism, Abstraction and Encapsulation concepts.
- Secured :Java has no pointer like C/C++. Java Programs run inside its own virtual machine sandbox and secure your data and program.



अब हम java के benefits और features के बारे में जानकारी लेंगे

java बहुत ही simple language है यह लगभग C, C++ के concept को follow करती है परन्तु यहा पर syntax different रहता है java का portable feature इसे C या C++ से अलग बनाता है

java का portable और platform का independent होना इसका सबसे बड़ा feature है इसकी एक सबसे बड़ी विशेषता virtual machine होना और दूसरा एक bytecode को तैयार करना है इन्ही के कारण यह अलग अलग platform पर आसानी से काम कर सकती है यहा पर platform से मतलब अलग अलग Windows या operating system से है अलग अलग hardware और software से है इसी के कारण यह सभी systems पर एक ही तरह के result available होता है एक तरह से produce करते है और code को run करती है

java का object oriented होना भी इसका एक बड़ा feature है object oriented programming अर्थात OOPs बहुत ही important feature है जिसमे object, class, inheritance, polymorphism, abstraction और encapsulation जैसे concept इसमें शामिल किए गए है इन सभी को java follow करता है

java का अगला feature इसका secured होना है इसका सबसे बड़ा feature इसका virtual machine होना है अर्थात JVM जोकि एक sandbox है अर्थात एक virtual box में कार्य करती है java में c, c++ की तरह pointer नहीं होते है अर्थात direct memory इंटरैक्शन java की नहीं रहती है virtual machine जोकि एक peace of software है जोकि हमारे कंप्यूटर में रहता है केवल इसी के अंतर्गत ही program को run करता है जिसके कारण यह system के data से इंटरैक्ट नहीं कर पाता है और हमारा data बिलकुल secure रहता है इसी लिए java experimental projects के लिए बहुत ज्यादा use किया जाता है

है

Benefits and features of JAVA

- Robust : Java is strong because of its memory management , automatic garbage collection, exception handling and type checking mechanism.
- Architecture Independent : C/C++ take different memory sizes for different Architecture. But java takes same memory sizes for different Architecture.
- Interpreted with High Performance : Java Byte codes are faster than traditional interpretation. With the comparison of C++ compiled codes Java byte codes are bit slower but with having feature of security and automatic garbage collection feature, they are best.
- Multithreaded : Threads are way to add multi-media, applications for webs. They are easily developed within Java with having common shared memory for all threads.
- Distributed: It is very easy to create distributed applications using RMI AND EJB.

java का एक feature इसका robust होना भी है java का सबसे बड़ा feature garbage collection है इसकी वजह से memory को free करवाना है तो यह कार्य java automatically कर देता है

java एक architecture independent है इसका मतलब कि java different architecture में चाहे Windows 32 बिट में या 64 बिट में काम करे तो यह एक ही तरह के memory size के variables को create करता है जिससे की अलग अलग platform पर यह चल पाती है

java का एक बड़ा feature interpreted with high performance होना है java byte codes produce करता है यह virtual machine से होते हुए hardware interaction करते है जोकि platform independent रहते है इस तरह से java interpreted और high performance language रहती है

java एक multithreaded language है इसका मतलब एक से ज्यादा program को एक साथ चलाने से है यह काम को आसानी से करवा सकते है जैसे कि अगर कुछ डाउनलोड भी करना चाहते है और कुछ टाइपिंग कार्य भी करना चाहते है तो java इस तरह के application को develop करना बहुत आसान रहता है

इसके अतिरिक्त java distributed भी है remote method invocation और enterpricesjava bins की सहायता से हम वैसी applications बना सकते है जिससे हम distibute आसानी से कर सकते है

इन सभी features के कारण java एक strong और important language बनकर उभरी है

Java and C++

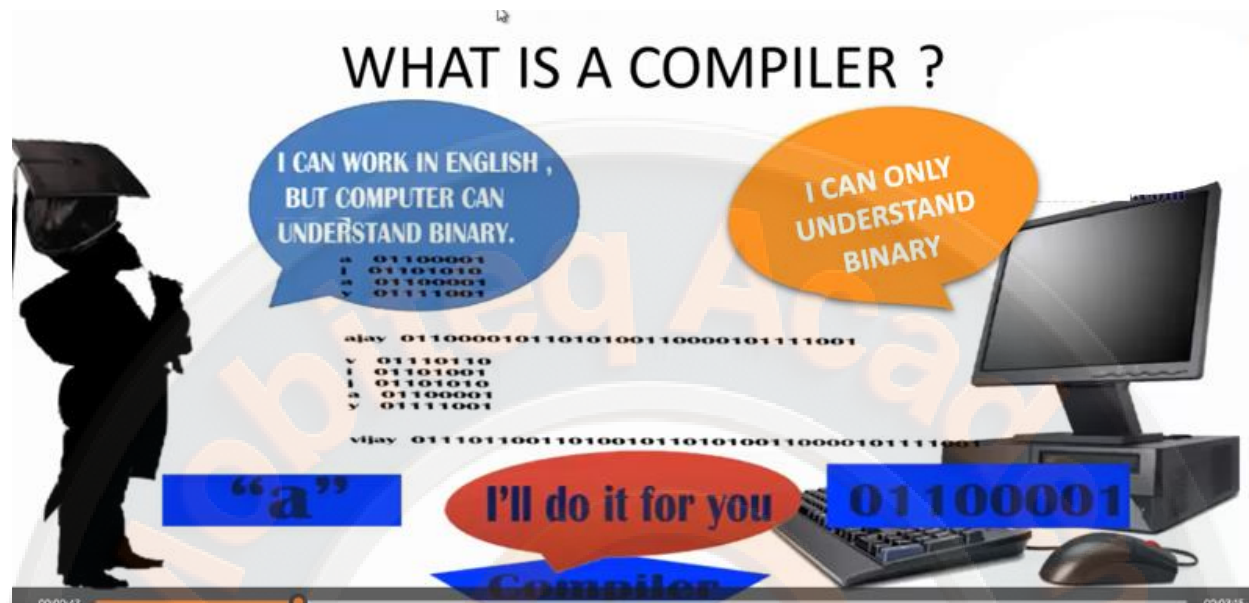
FEATURE	C++	JAVA
USAGE	MAINLY IN SYSTEM PROGRAMMING.	USED IN DESKTOP/WEB/MOBILE APPLICATIONS
DEPENDENCY	PLATFORM DEPENDENT.	PLATFORM INDEPENDENT.
COMPILER OR INTERPRETER	ONLY COMPILER	BOTH COMPILER AND INTERPRETER
INHERITANCE	MULTIPLE	SINGLE
POINTERS	YES	NO
VIRTUAL MACHINE	NO	YES
THREAD IMPLEMENTATION	NO	YES
STRUCTURE	YES	NO
UNION	YES	NO
OPERATOR OVERLOADING	YES	NO

यहा पर java और C++ language के बीच difference show किया गया है

Java and Java Script

FEATURE	JAVA SCRIPT	JAVA
USAGE	BROWSER BASED	USED IN DESKTOP/WEB/MOBILE APPLICATIONS
OBJECT	OBJECT BASED PROGRAMMING LANGUAGE	OBJECT ORIENTED PROGRAMMING LANGUAGE
DEPENDENCY	BROWSER BASED	PLATFORM INDEPENDENT.
COMPILER OR INTERPRETER	ONLY INTERPRETED BY BROWSER , NO INTERPRETER IS REQUIRED	BOTH COMPILER AND INTERPRETER
DATA TYPES	NO DATA TYPE IS DEFINED ONLY "VAR" KEYWORD WILL WORK.	STRONGLY DEFINED DATA TYPES.

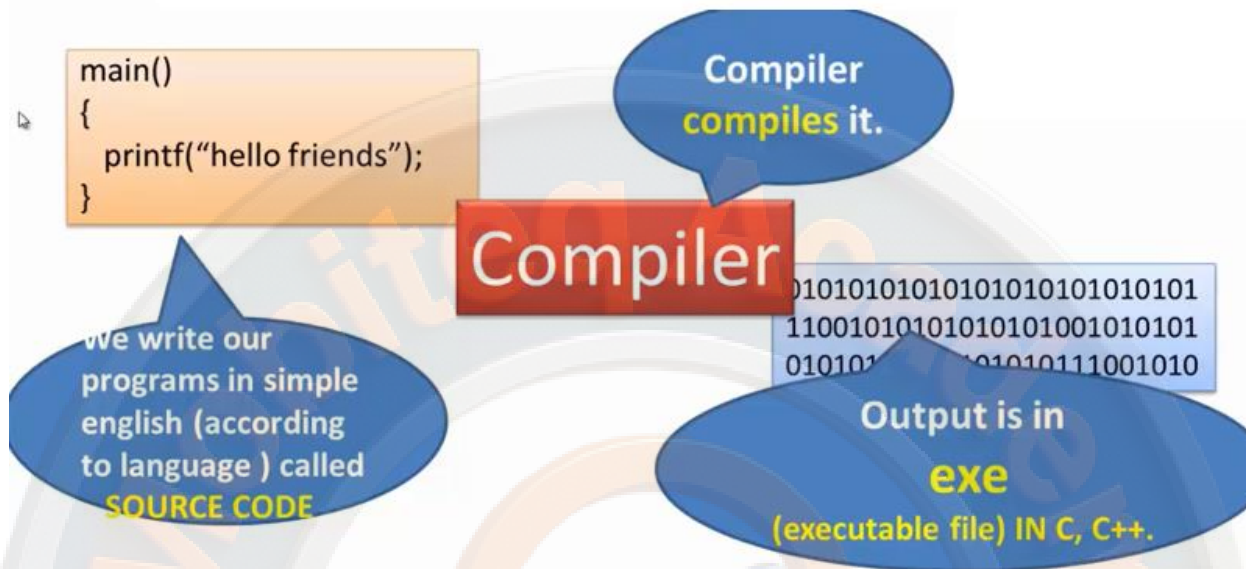
इसी तरह हम java और java script के बीच difference को show किया गया है



Compiler is a software which converts our simple english instructions to binary language for computer in c, c++ .

compiler एक software है जो हमारे कंप्यूटर में होता है जिससे हम आपने source code को binary या machine code में convert कर सकते हैं जिसमे हमने अपनी simple english में टाइप किया है यह हमने ajay का binary code तैयार किया है यहा पर 'a' का अलग binary code है ' j ' का अलग binary code है a का फिर से binary code है और y का अलग binary code है इस प्रकार ajay का binary code में लिखना बड़ा मुश्किल है और याद रखना भी मुश्किल है इसी प्रकार से vijay का binary code अलग है ऐसे में हम compiler का use करेगे हम आपनी english में type करेगे जिसको source code कहते हैं compiler source code को binary code में convert करता है

SOURCE CODE TO EXE FILES



यहा एक example है जिसमे हमने एक source code जोकि C language का code है जिसमे हमने अपनी simple english में टाइप किया और compiler ने उसे compiles करके कंप्यूटर की binary language में करके दिया इस तरह से compiler का कार्य हमारे द्वारा दिए गए source code को binary code या machine code में convert करना है



अगर इन सब को जोड़ दिया जाए तो programmer का कार्य programs बनाना है और program बनाते समय programming language का प्रयोग करना है जैसे C, C++, C# etc. इन language में जो instruction होते हैं वो simple language में होते हैं हर language के आपने rules और tools होते हैं इसमें compiler का मुख्य कार्य हमारे द्वारा दिए source code को binary या machine codes में convert करना है इस प्रकार हमारे पास programs बनकर तैयार है जो हमारे पास exe या executable फाइल format में available होते हैं परन्तु java में class files और java files बनती हैं इस तरह से compiler हमारे कार्य को करता है

Compiler and Interpreter

Source code

Line 1
Line 2
Line 3
Line 4
Line 5

Compiler

It compiles in one go and check for error (If there is any) and finally converts into machine code..

Source code

Line 1
Line 2
Line 3
Line 4
Line 5

Interpreter

It converts one statement of a program at a time.

हम compiler और interpreter के कार्यों का समझते हैं left side compiler है मान ले कि हमारे पास 5 line का source code है compiler एक ही बार में सभी lines को check करेगा कि कहीं कोई गलती तो नहीं है और फिर machine code में convert करके देगा और यह सब one go अर्थात एक ही बार में कर देगा

लेकिन interpreter line by line code को check करेगा और बारी बारी से उसे machine code में convert करेगा इन दोनों के अपने अपने feature हैं अब हम इनके बीच के difference को समझते हैं

Compiler and Interpreter

Interpreter	Compiler
<ul style="list-style-type: none"> • It converts one statement of a program at a time. • Its execution time is slower. Because it analyze the source code line by line. So it takes time to interpret the program into machine code until it reaches last line. • No intermediate object code is generated so It is memory efficient because no intermediate object code is require to generate. • It finds error line by line , so it is easy to work with interpreters. • Python, Ruby and use interpreters to execute programs. 	<ul style="list-style-type: none"> • It complies in one go and check for error (If there is any) and finally converts into machine code. • its execution comparatively faster but it takes large amount of time to analyze the source code. • It requires linking /loading and it generates intermediate object code which are used by linking and loading so it requires more memory to compile and execute programs. • It scans full program in one go and show us any error in program. Sometimes this makes typical to work with compiler • C, C++, C# require compiler to work.
<p>Java uses both compiler and interpreter</p>	

हम compiler और interpreter के difference के बारे में जानेगे

हमने अभी देखा कि interpreter one by one code को check करता है जबकि compiler एक ही बार में program को check करता है

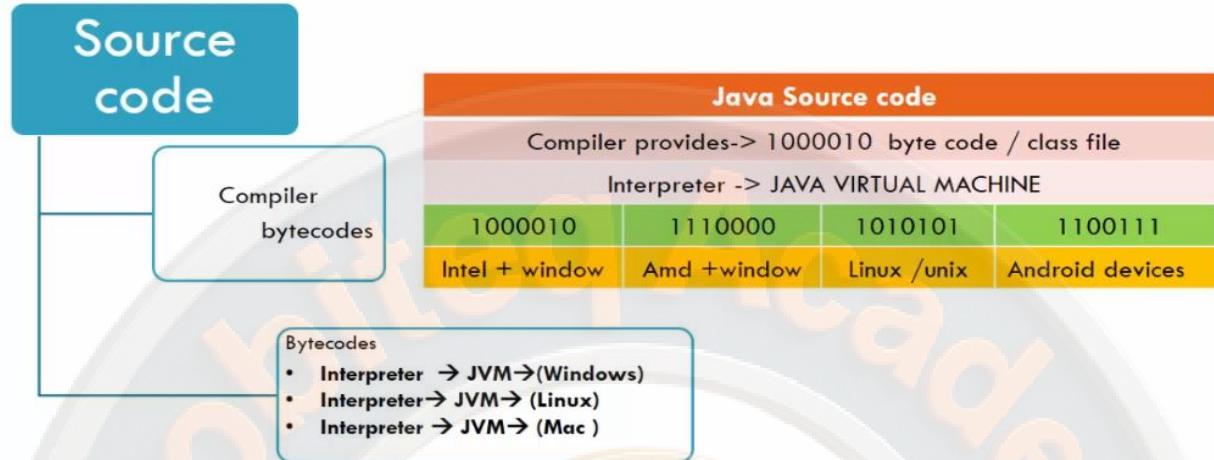
interpreter का execution time slow रहता है जबकि compiler का execution time fast रहता है

इसी तरह अगर हम memory efficiency की बात करे तो interpreter थोडा सा ज्यादा memory efficient है क्योंकि इसको intermediate object code तैयार करने की जरूरत नही रहती है इस कारण से यह ज्यादा memory नही लेता है जबकि compiler को intermediate code तैयार करना पड़ता है जिसको बाद में linking और loading के साथ use किया जाता है जिस कारण से compiler को ज्यादा memory चाहिए होती है

interpreter error को line by line check करता है जिसके कारण user आसानी से उस line में error ठीक कर सकता है जबकि compiler पुरे program को एक बार में scan करता है और काफी सारे error हमे show करता है तो हमे error को ढूंढ कर उसे ठीक करना होता है जिस कारण थोडा typical सा हो जाता है

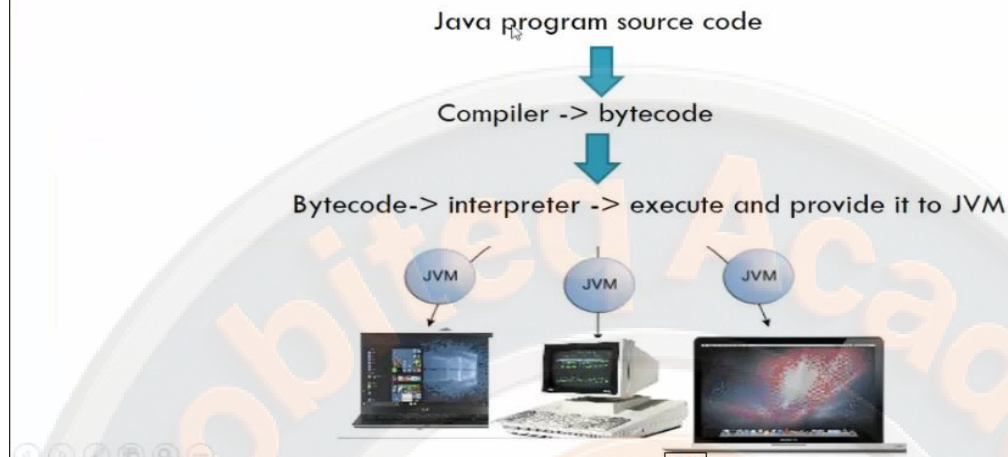
interpreter में program execute करने के लिए ruby और python जैसी language use की जाती है जबकि C, C++ OR C# जैसी language compiler का use करती है परन्तु java दोनों को use करती है

Java uses both Compiler and Interpreter



यहा पर source code java का है अगर compiler उसे compile करे तो वह bytecodes में convert हो जाते है जो एक तरह की class file होती है जिसका use interpreter के द्वारा किया जाता है यहा जो errors रहते है वह compiler के दवारा ठीक करवा लिए जाते है और execute करने के लिए java interpreter का use करता है तो इस तरह जो bytecode available रहता है उसे line by line interpreter execute करता है और java virtual machine एक software segment है जो कंप्यूटर में होता है उसे interpreter सारा code provide करता है और फिर java virtual machine उन instruction को code के रूप में create करती है जोकि हमारे platform के acording होता है जैसे कि अगर हम intel+window का architecture use कर रहे है तो उसके acording ही java virtual machine हमे code provide करेगी इसी तरह other architecture के अनुसार code provide करती है तो इस तरह java virtual machine एक sandbox की तरह रहता है जोकि platform को देखते हुए code को तैयार करता है जिससे की code आसानी से execute हो जाता है

JAVA VIRTUAL MACHINE

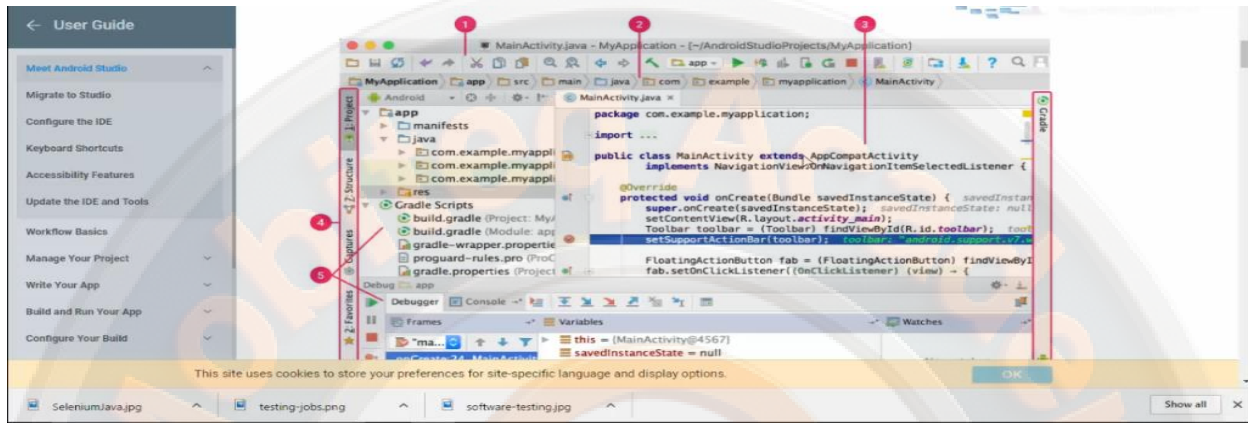


हम यहा पर एक ओर example देखते है जिसमे java का source code सबसे पहले compiler के पास जाता है compiler उसमे यदि कोई error है उसे ठीक कराने के बाद bytecodes में convert करता है इन्ही bytecode को interpreter line by line execute करता है और फिर JVM को प्रदान करता है और फिर JVM ऐसा code produce करता है जोकि हमारे platform code के according होता है तो इस तरह से java virtual machine, java का एक important पार्ट है जिससे की java के द्वारा create किए गए programs machine independent होते है

Where we use Java?

- ❑ **Android Apps**
- ❑ **Software testing**
- ❑ **Games development**
- ❑ **Scientific Applications**
- ❑ **Java web apps / E commerce /web application**
- ❑ **J2ME Apps for different hardware**

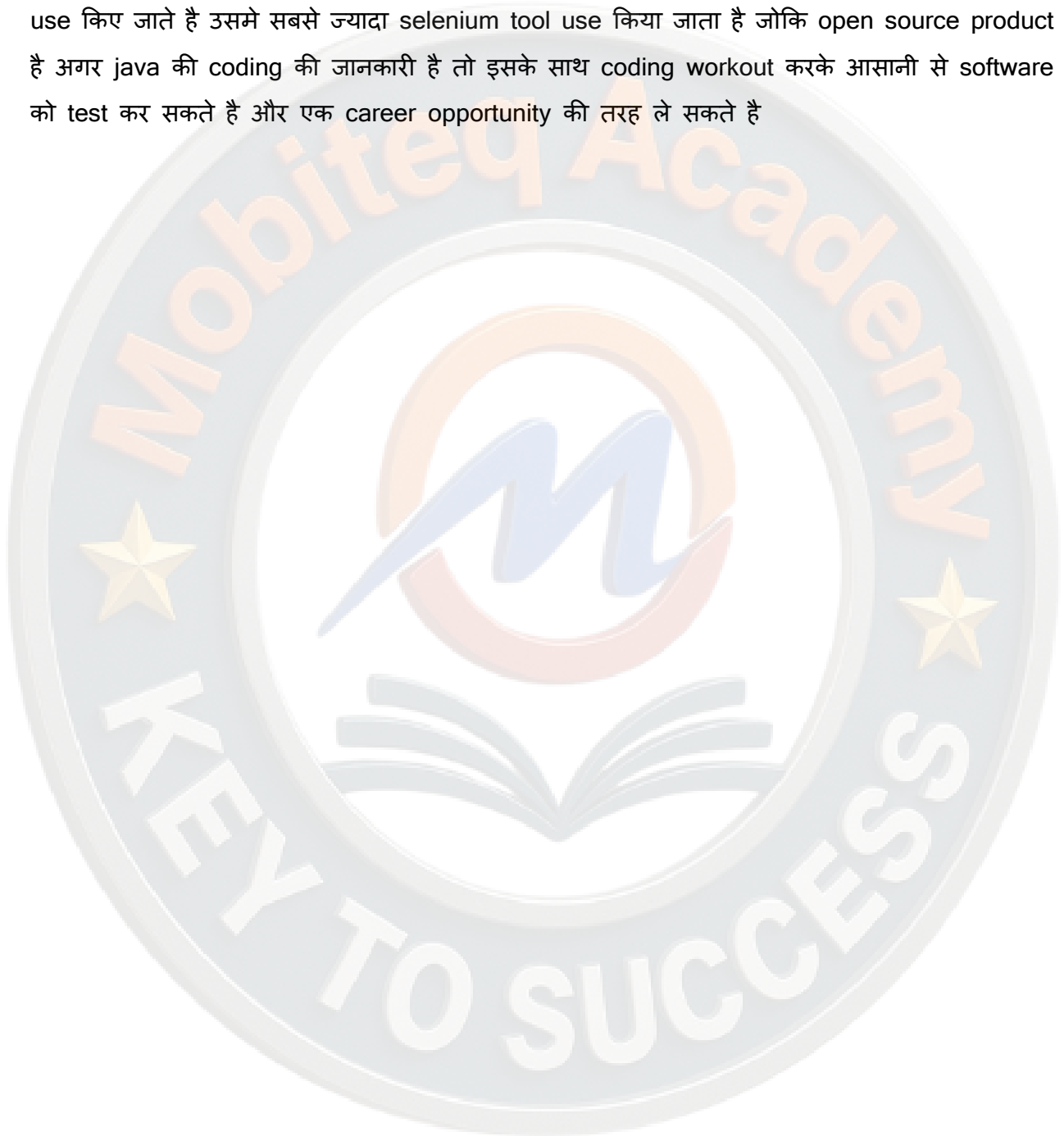
यहा पर java का use जहा पर किया जाता है उसके बारे में बताया गया है सबसे पहले हम android apps और software testing दो जगह इसका uses देखते है



आज सभी android के बारे में जानते है androids के back में इसका use किया जाता है android studio एक software है जिससे की android application को develop कर सकते है हमने यहा पर एक developer android.com एक URL open किया हुआ है जिसमे android studio का screenshot available है जिसमे हम android apps को develop कर सकते है यदि हम ऊपर देखे तो main activity.java लिखा गया है जब हम आपने program को save करते है तो हम .java का use करते है यहा पर java के सभी rules को follow किया जाता है जैसे की सभी class और उनके method को use किया गया है तो java एक practical usage android development में java का use होना भी



software testing -: यह एक बढ़िया career opportunity के रूप में उपलब्ध है यदि कोई student है जो जॉब section में जाना चाहता है तो वह software testing को as a जॉब career भी ले सकते हैं software testing में java का बहुत use किया जाता है यदि इंटरनेट पर check करे तो जो tools use किए जाते हैं उसमें सबसे ज्यादा selenium tool use किया जाता है जोकि open source product है अगर java की coding की जानकारी है तो इसके साथ coding workout करके आसानी से software को test कर सकते हैं और एक career opportunity की तरह ले सकते हैं





games development :- हमने net पर java games को सर्च किया तो हमारे पास काफी सारी games available हो गई है इसे भी एक career opportunity की तरह ले सकते है

scientific application :- java का use scientific application में बहुत किया जाता है क्योंकि इसमें security का ध्यान रखा जाता है इसमें JVM होता है और java में pointer नहीं रहते है जिसके कारण यह कंप्यूटर memory से direct intrect नहीं कर सकती है

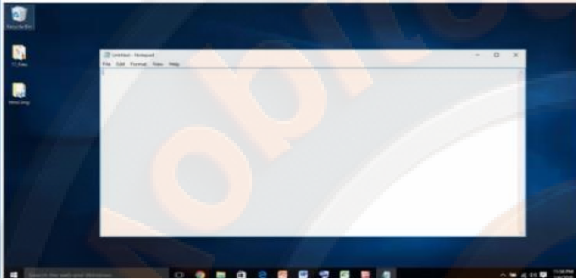
इसी तरह java का वेब apps, E commerce और वेब application में भी java का बहुत use किया जाता है

इसी तरह J2ME apps बनाने के लिए जोकि different hardware पर बनाए जाते है इसके लिए भी java का use किया जाता है

इस तरह से java का use बहुत कार्य में किया जाता है

We need two Softwares to run

- ▣ A Simple text editor to write java programs like notepad.

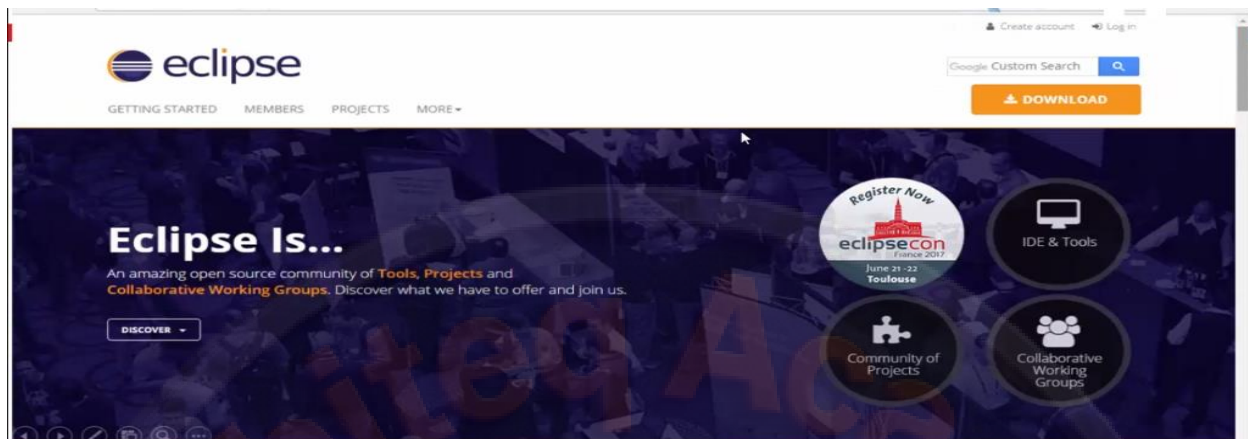


- ▣ JAVA SE DEVELOPMENT KIT

jdk



हम यहा पर जानते है कि java को run करने के लिए हमे क्या क्या चीजो की जरूरत रहती है इसमें हमारे पास दो software बहुत ही जरूरी रहते है यदि हम java को command prompt से run करना चाहते है इसमें हम एक normal text editor notepad का use कर सकते है इसके अतिरिक्त online jdk को डाउनलोड करना होता है

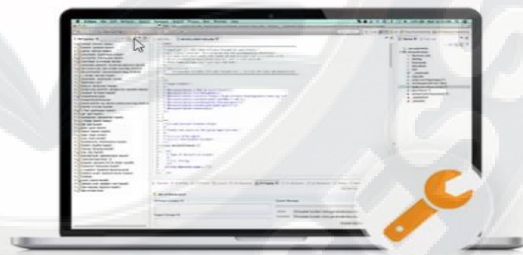


हमारे पास एक दूसरा तरीका भी है यहा पर eclipse के नाम से जो पहले से एक software available है जोकि open source software है इसको free में डाउनलोड कर सकते है इससे आसानी से java के program को बना सकते है

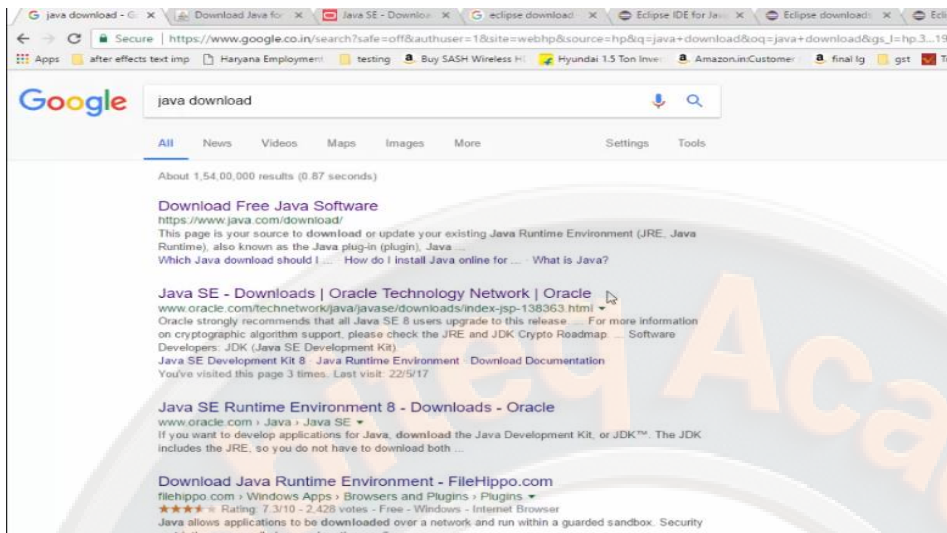
Eclipse is an Integrated development

Tools and IDEs

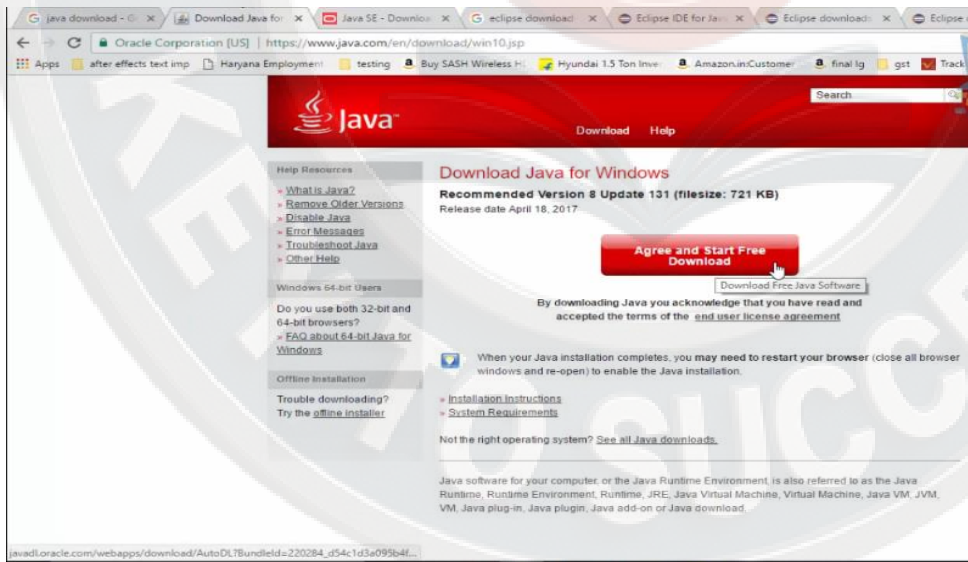
It has various tools and IDEs for Java, C/C++, JavaScript and PHP IDEs and it also covers many more languages.



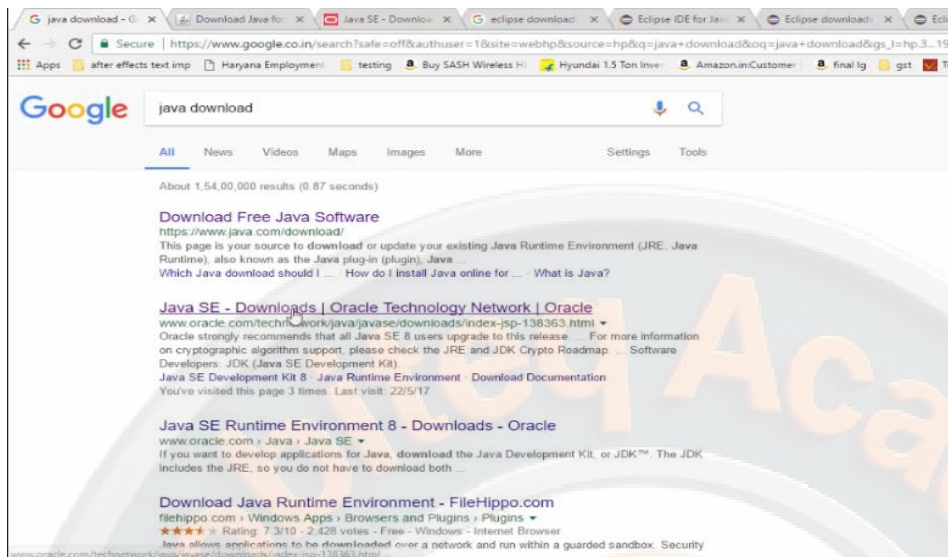
java में eclipse एक ID है जोकि intergrated development environment है जिसमे ना केवल java बल्कि ओर भी काफी सारी languages को run कर सकते है लेकिन हम इसे java के prospect से देखने वाले है



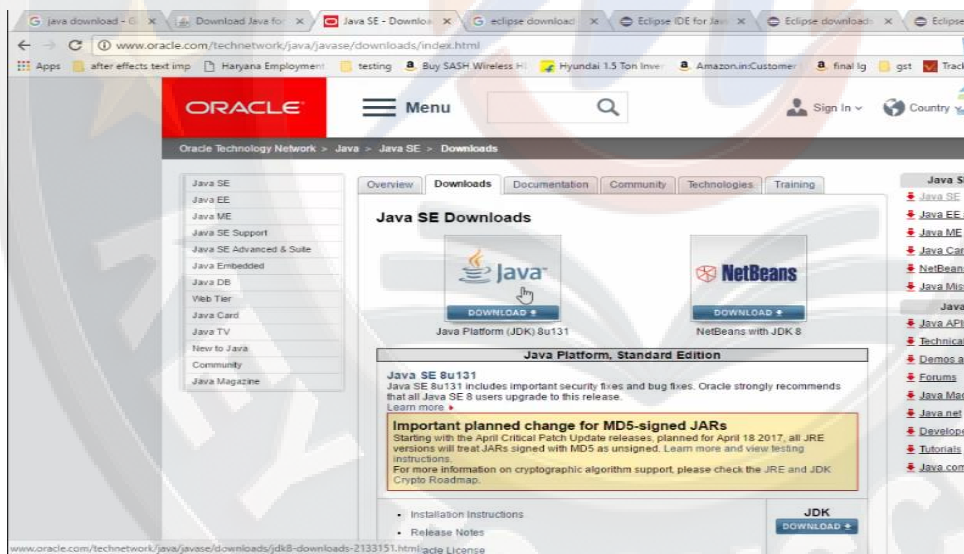
अब हम java को डाउनलोड करना सीखेंगे और उसको install करेंगे यहाँ पर हमने java download type करके search पर क्लिक किया तो हमारे पास काफी सारे result available हो गए हैं यहाँ पर हमने download free java software को एक new tab में open किया है



यहाँ से हम इस option पर क्लिक करके java को free में डाउनलोड कर सकते हैं



हमने second link को new tab में open किया



यहा से हमने java डाउनलोड पर क्लिक किया

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to Java.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u131 checksum

Java SE Development Kit 8u131

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.87 MB	jdk-8u131-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.81 MB	jdk-8u131-linux-arm64-vfp-hflt.tar.gz
Linux x86	164.86 MB	jdk-8u131-linux-i586.rpm
Linux x86	179.39 MB	jdk-8u131-linux-i586.tar.gz
Linux x64	162.11 MB	jdk-8u131-linux-x64.rpm
Linux x64	176.95 MB	jdk-8u131-linux-x64.tar.gz
Mac OS X	226.57 MB	jdk-8u131-macosx-x64.dmg
Solaris SPARC 64-bit	139.79 MB	jdk-8u131-solaris-sparcv9.tar.Z

हमारे पास यहा पर काफी सारे downloads की जानकारी available हो जाएगी यह latest version available है

Java SE Development Kit 8u131 Demos and Samples

You must accept the Oracle BSD License, to download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	9.84 MB	jdk-8u131-linux-arm32-vfp-hflt-demos.tar.gz

हमने इसमें accept license agreement पर क्लिक करके डाउनलोड कर सकते है

www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Thank you for downloading this release of the Java™ platform, standard Common Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select Subscriptions, expand Technology, and subscribe to Java.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u131 checksum

Java SE Development Kit 8u131

You must accept the Oracle Binary Code License Agreement for Java SE to download this software. Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.87 MB	jdk-8u131-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.81 MB	jdk-8u131-linux-arm64-vfp-hflt.tar.gz
Linux x86	164.66 MB	jdk-8u131-linux-i586.rpm
Linux x86	179.39 MB	jdk-8u131-linux-i586.tar.gz
Linux x64	162.11 MB	jdk-8u131-linux-x64.rpm
Linux x64	176.96 MB	jdk-8u131-linux-x64.tar.gz
Mac OS X	226.57 MB	jdk-8u131-macosx-x64.dmg
Solaris SPARC 64-bit	139.79 MB	jdk-8u131-solaris-sparcv9.tar.z
Solaris SPARC 64-bit	99.13 MB	jdk-8u131-solaris-sparcv9.tar.gz
Solaris x64	140.51 MB	jdk-8u131-solaris-x64.tar.z
Solaris x64	96.96 MB	jdk-8u131-solaris-x64.tar.gz
Windows x86	191.22 MB	jdk-8u131-windows-i586.exe
Windows x64	198.03 MB	jdk-8u131-windows-x64.exe

Java SE Development Kit 8u131 Demos and Samples Downloads

You must accept the Oracle BSD License, to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
----------------------------	-----------	----------

यहाँ पर हमारे पास अलग अलग bit के version available है हमें अपने कंप्यूटर के बिट के हिसाब से अलग अलग version को डाउनलोड करना पड़ सकता है हमारे पास Window 10 available है तो हम यहाँ से Windows का option choose करेंगे अगर आप अपने Windows की bit जानना चाहते हैं

www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Thank you for downloading this release of the Java™ platform, standard Common Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select Subscriptions, expand Technology, and subscribe to Java.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u131 checksum

Java SE Development Kit 8u131

You must accept the Oracle BSD License, to download this software. Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.87 MB	jdk-8u131-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.81 MB	jdk-8u131-linux-arm64-vfp-hflt.tar.gz
Linux x86	164.66 MB	jdk-8u131-linux-i586.rpm
Linux x86	179.39 MB	jdk-8u131-linux-i586.tar.gz
Linux x64	162.11 MB	jdk-8u131-linux-x64.rpm
Linux x64	176.96 MB	jdk-8u131-linux-x64.tar.gz
Mac OS X	226.57 MB	jdk-8u131-macosx-x64.dmg
Solaris SPARC 64-bit	139.79 MB	jdk-8u131-solaris-sparcv9.tar.z
Solaris SPARC 64-bit	99.13 MB	jdk-8u131-solaris-sparcv9.tar.gz
Solaris x64	140.51 MB	jdk-8u131-solaris-x64.tar.z
Solaris x64	96.96 MB	jdk-8u131-solaris-x64.tar.gz
Windows x86	191.22 MB	jdk-8u131-windows-i586.exe
Windows x64	198.03 MB	jdk-8u131-windows-x64.exe

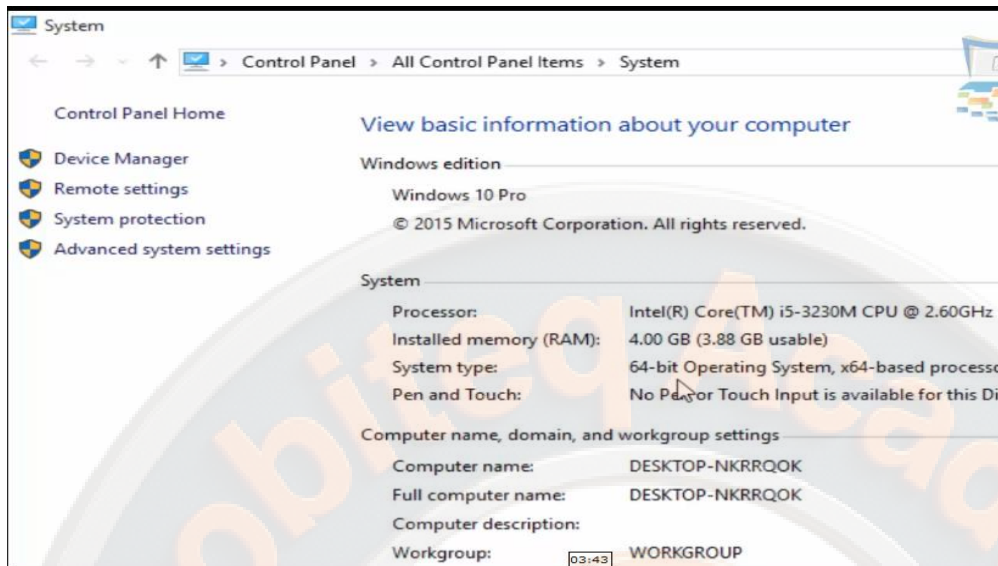
Java SE Development Kit 8u131 Demos and Samples Downloads

You must accept the Oracle BSD License, to download this software.

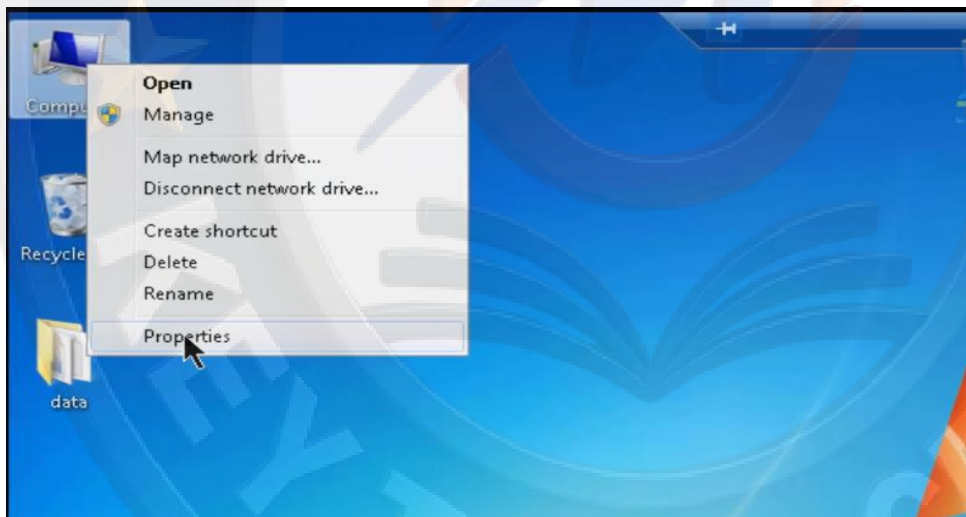
Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
----------------------------	-----------	----------

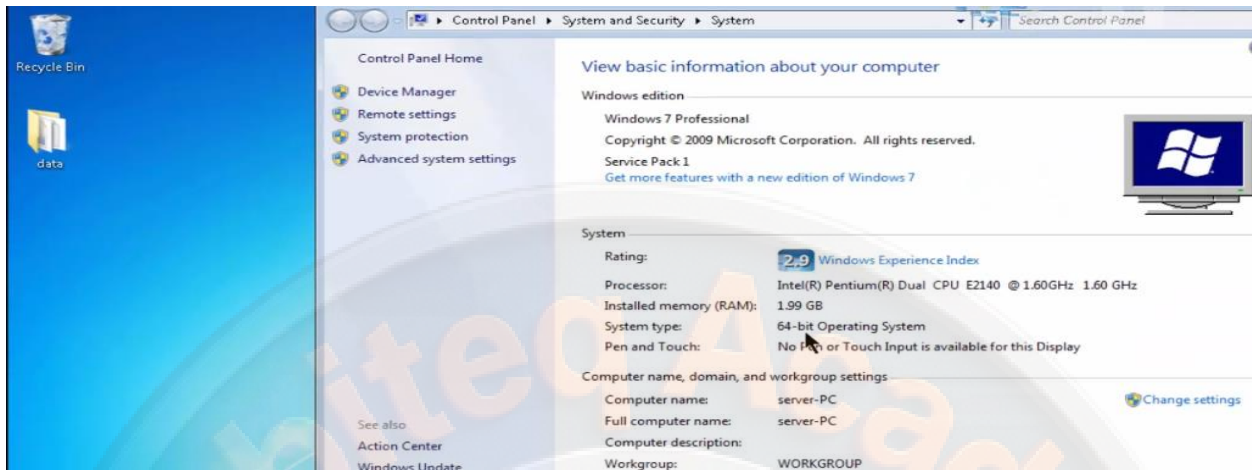
तो आप start बटन पर right क्लिक करके system option पर क्लिक करें



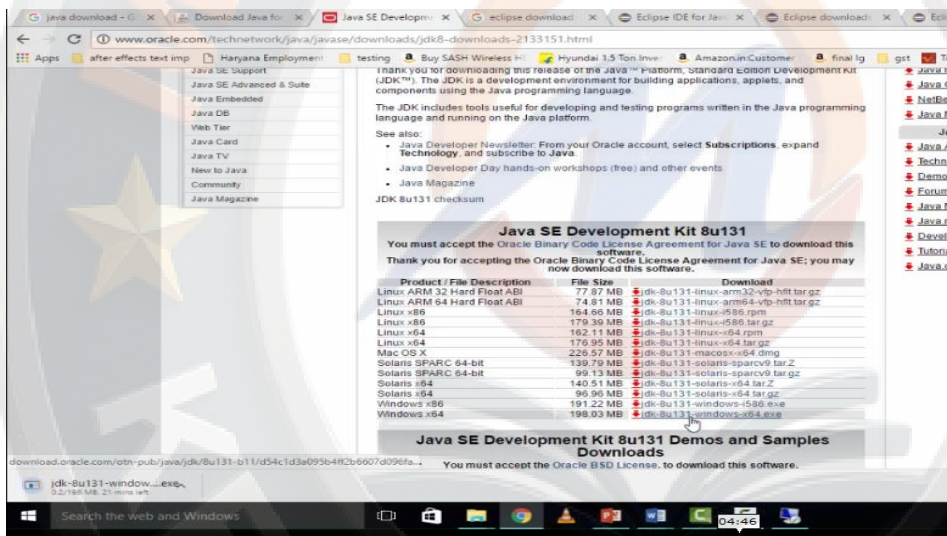
यहा पर हमे system type में बिट की जानकारी दी जा रही है हमारा operating system 64 बिट का है



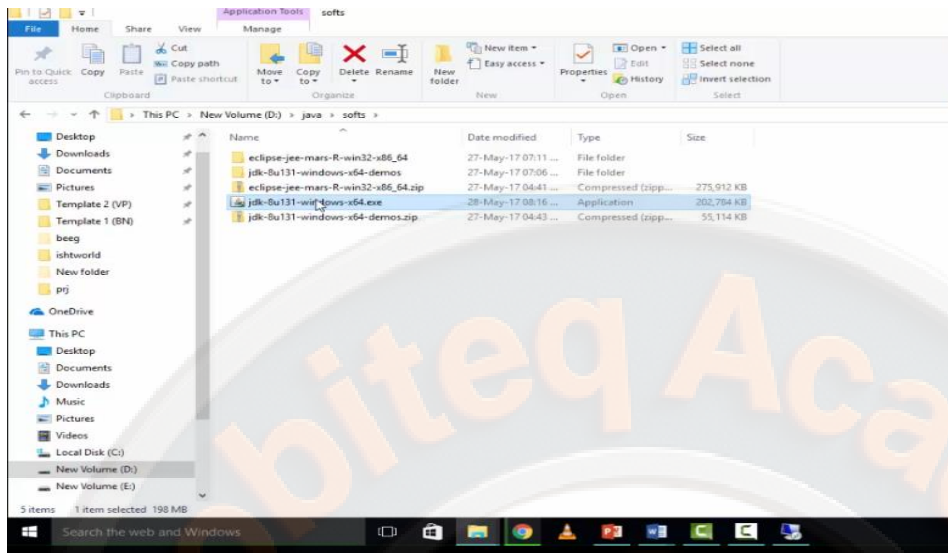
इसी तरह अगर Windows7 का use कर रहे है तो computer पर right क्लिक करके properties option पर क्लिक करे



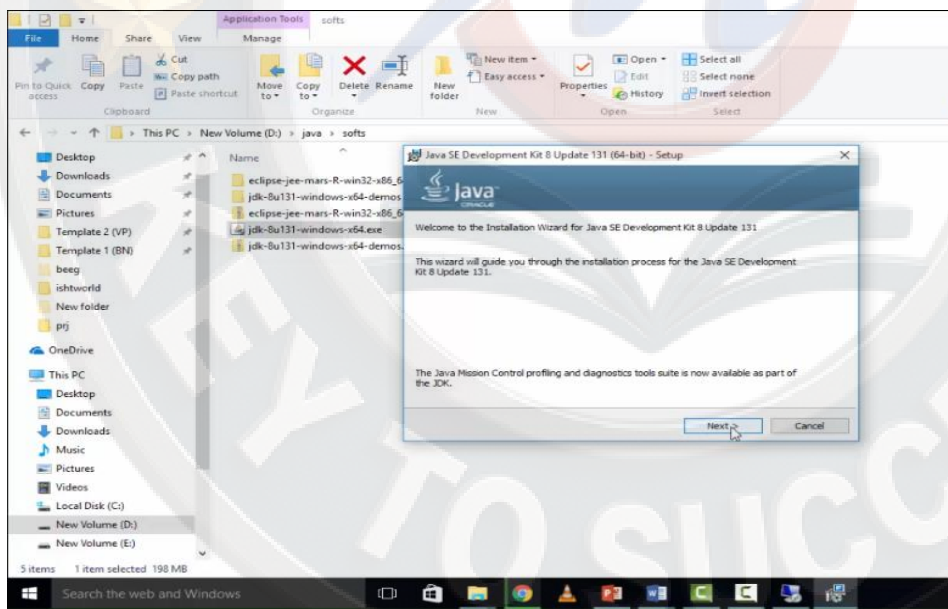
तो हमारे पास यहा पर Windows के बारे में काफी सारी information available हो जाएगी यहा से हम system type देख सकते है



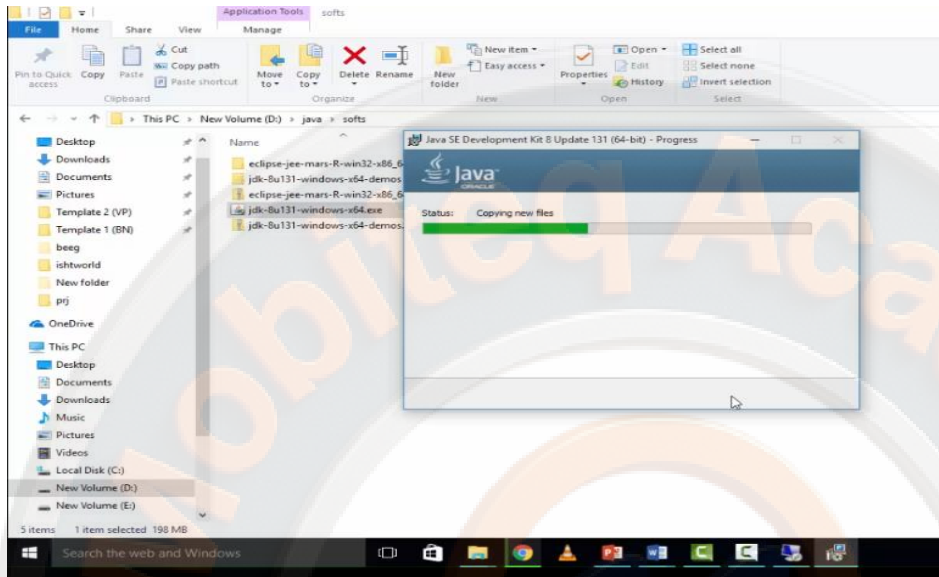
यहा से हमने Windows 64 पर क्लिक किया तो left side पर डाउनलोड होना शुरू हो जाएगा



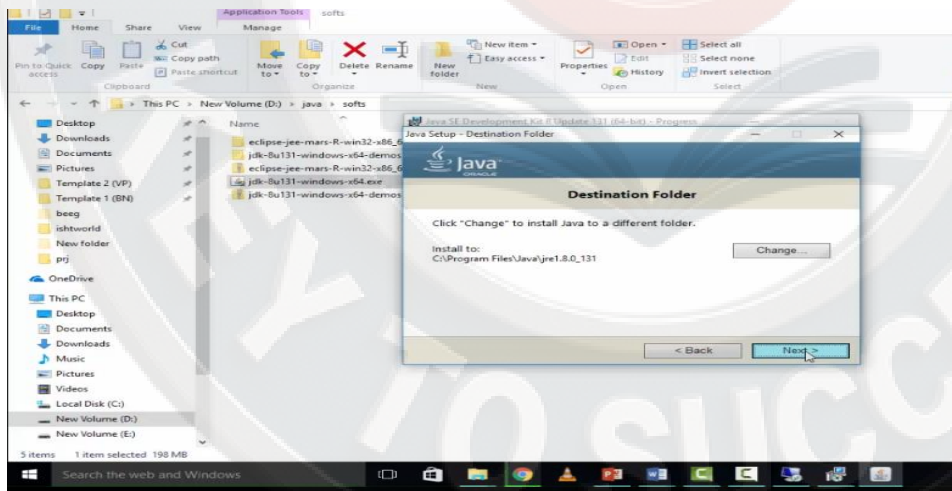
डाउनलोड होने के बाद हमारे पास यहा पर available है हमे यहा से क्लिक करके install करना है हमने double क्लिक किया यहा अगर कुछ allow का message आता है तो इसे allow करे



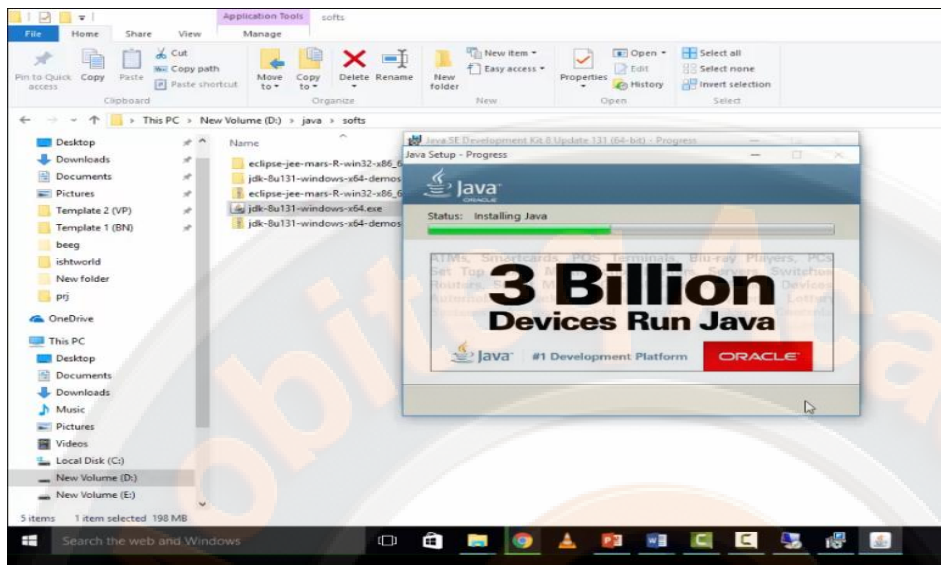
यहा से next पर क्लिक करे हमने next next पर क्लिक किया



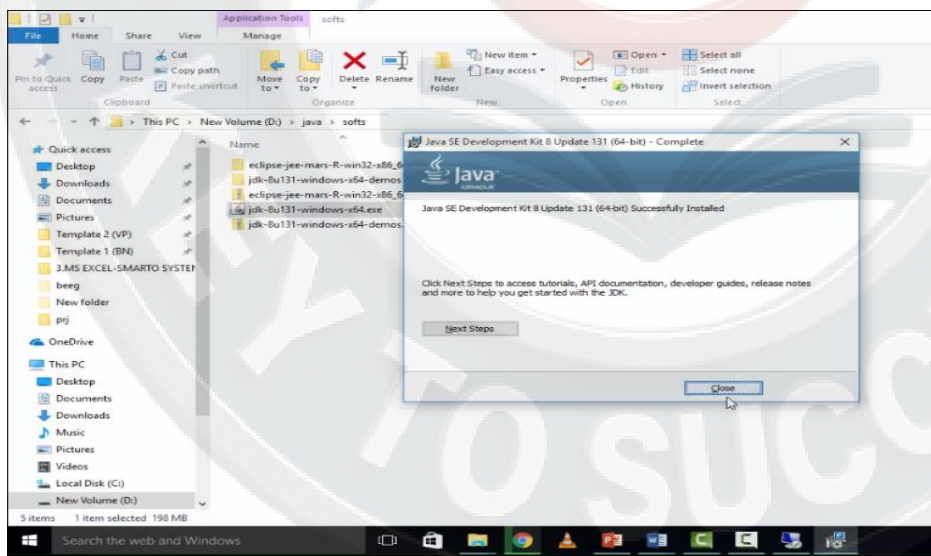
हमारे पास installation शुरू हो जाती है इस तरह आसानी से हम java को डाउनलोड कर सकते हैं



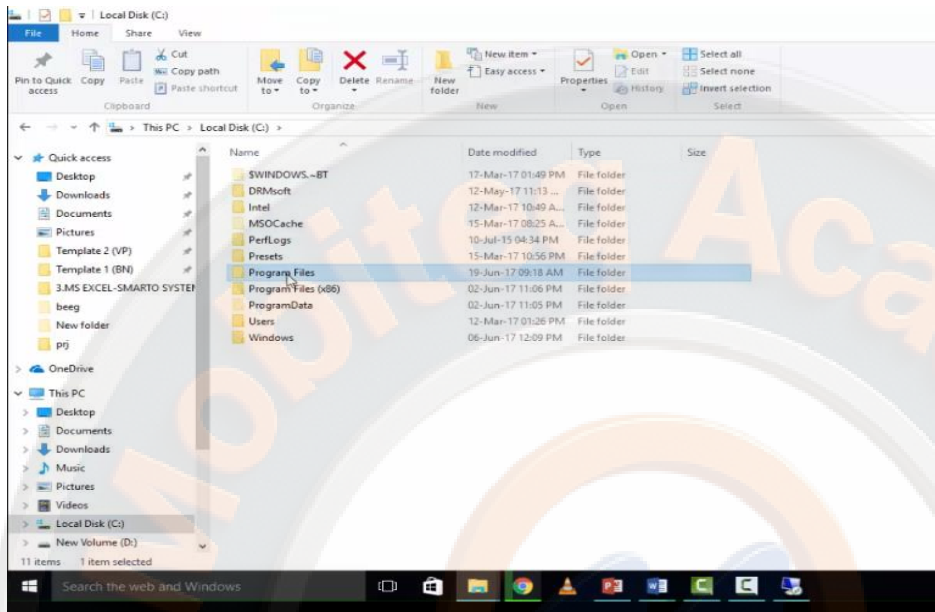
हमने यहा से next पर क्लिक किया



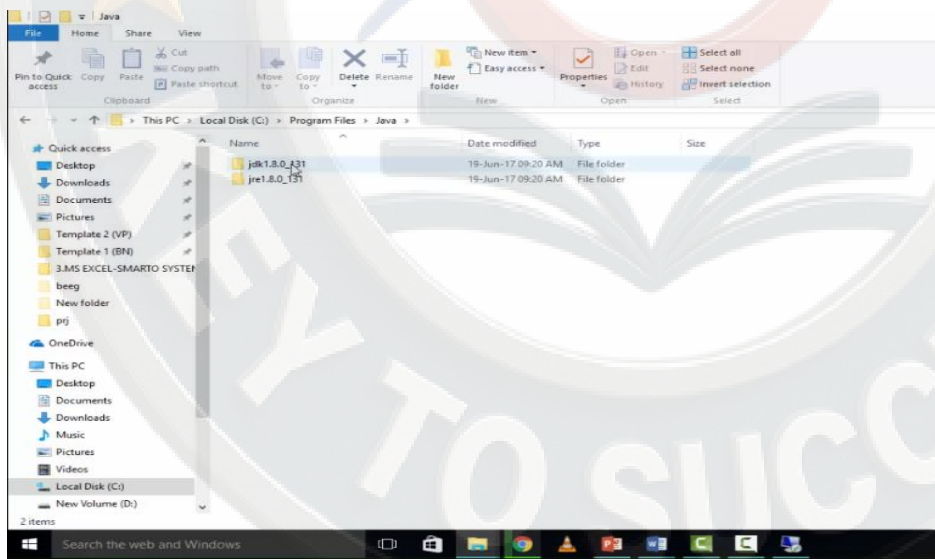
यहा पर install करते समय java की कुछ जानकारी available होती रहेगी



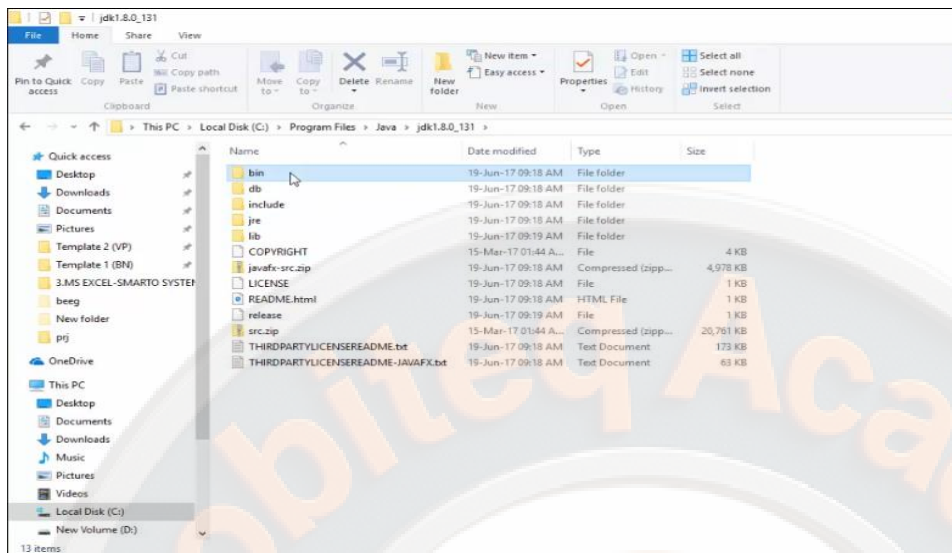
हमारे पास java successfully instaled हो गया है और close पर क्लिक किया



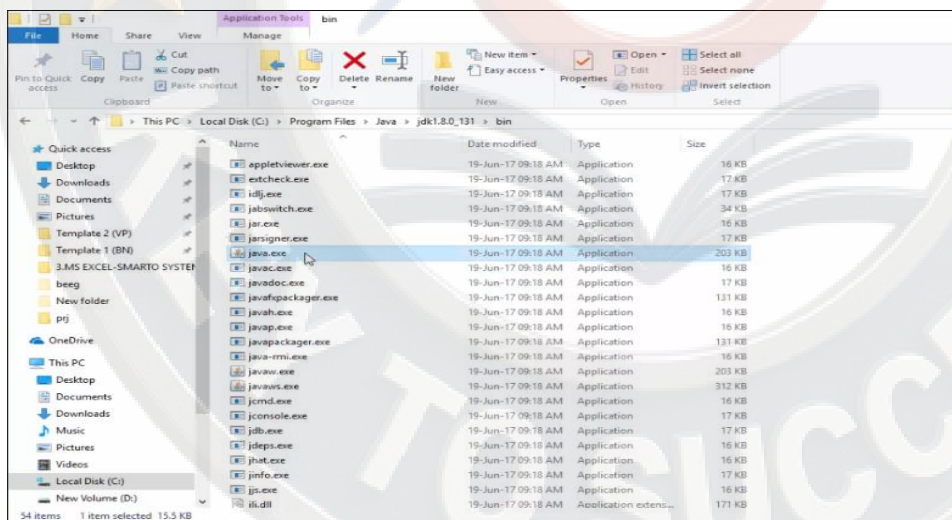
अगर आप देखना चाहते हैं तो c drive पर क्लिक करके program files पर क्लिक करे तो इसमें java का फोल्डर available है उस पर क्लिक करे



यहा पर java की installation files को रखा हुआ है अगर हम jdk पर क्लिक करे

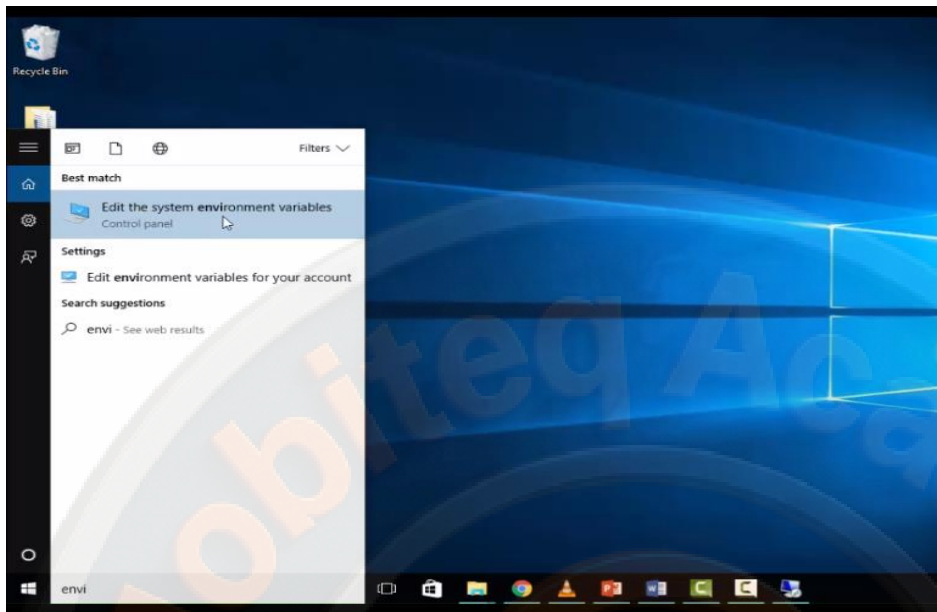


यहा पर काफी सारी files और folders available है जिनको अभी use किया गया है अगर यहा से bin पर क्लिक करे

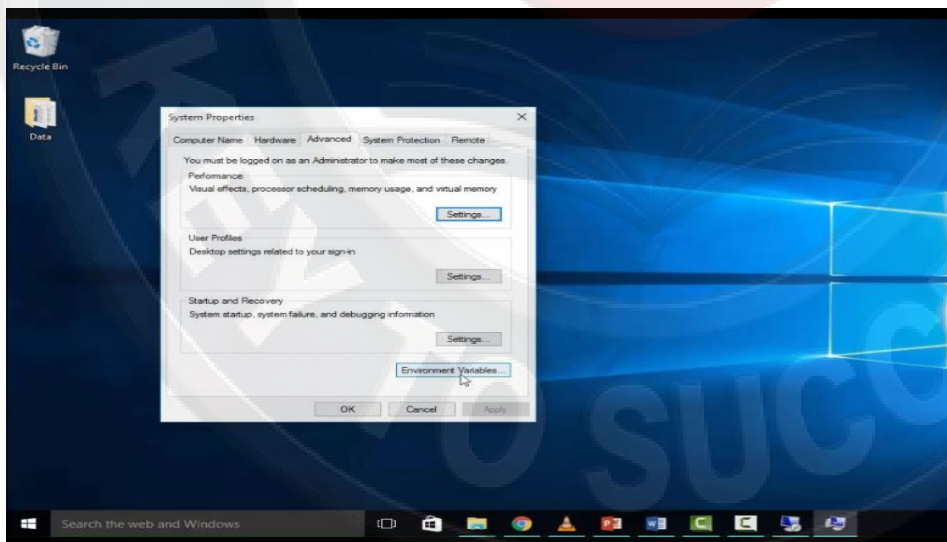


तो यहा पर हमारे पास java की exe files available है जो java में use होने वाली है जिनमे java और javac दो मुख्य रहेगी जिनका use हम करेगे

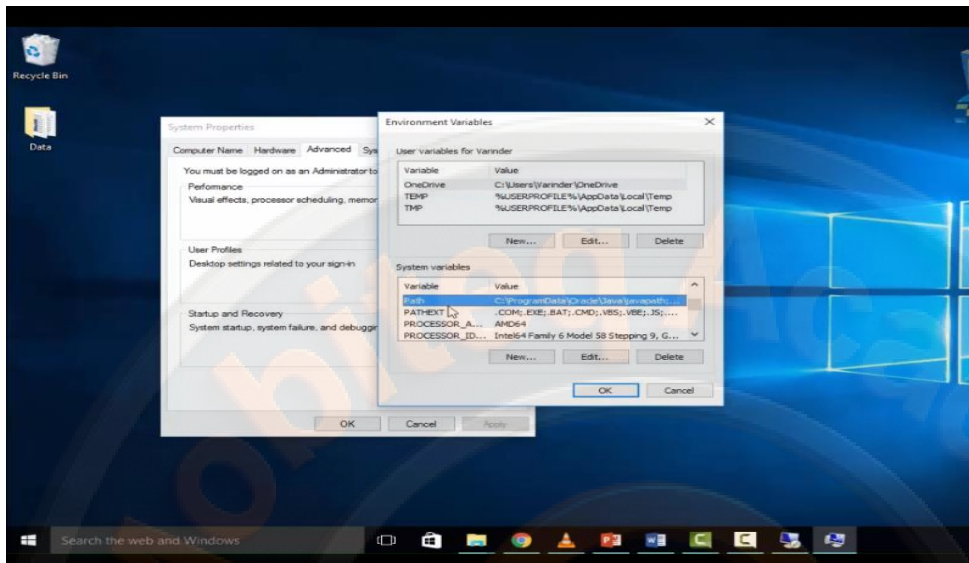
java को डाउनलोड करने के बाद हमे java variables की setting करनी होती है हम वह settings करेगे



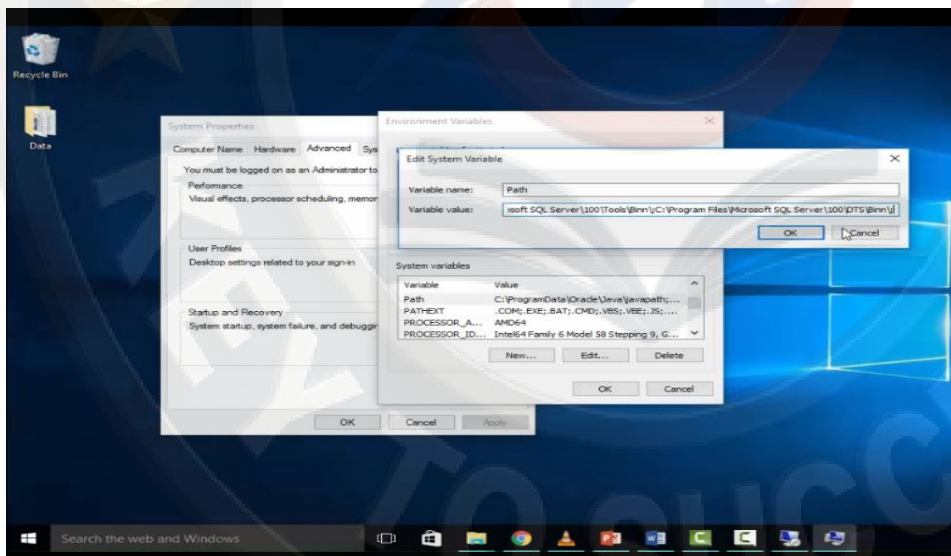
system environment variable set करने के लिए अगर आप Windows10 use कर रहे हैं तो search box में environment variables type करे तो काफी सारे option available हो जाएगे इस पर क्लिक करे



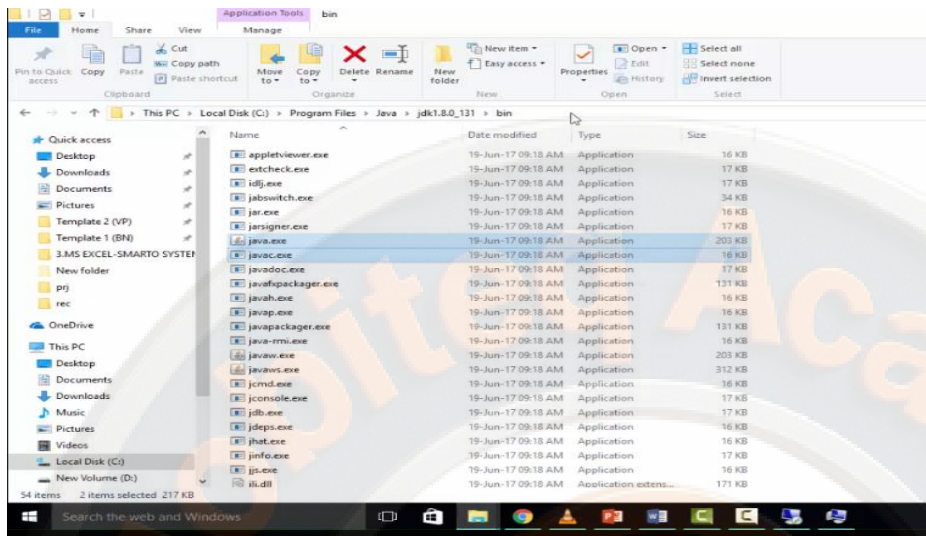
हमारे पास advance section open हो जाएगा यहा पर environment variables के option पर क्लिक करे path set करने का फायदा यह होगा कि java कही पर भी install हो तो java के program को कही भी save करके directlly ID और command prompt के साथ run कर सकते हैं



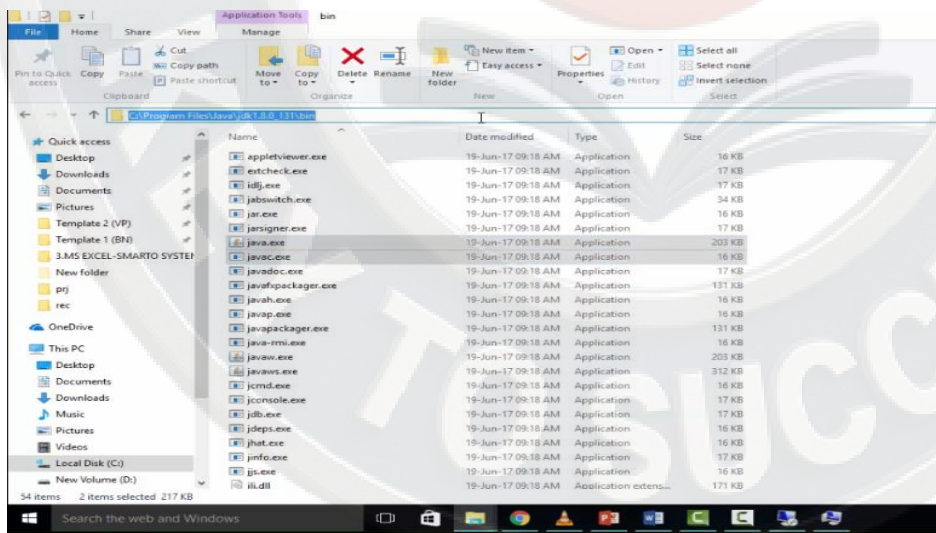
यहा से नीचे section में जाकर P key press करे तो path available होगा यहा से path पर क्लिक करके edit पर क्लिक करेगे



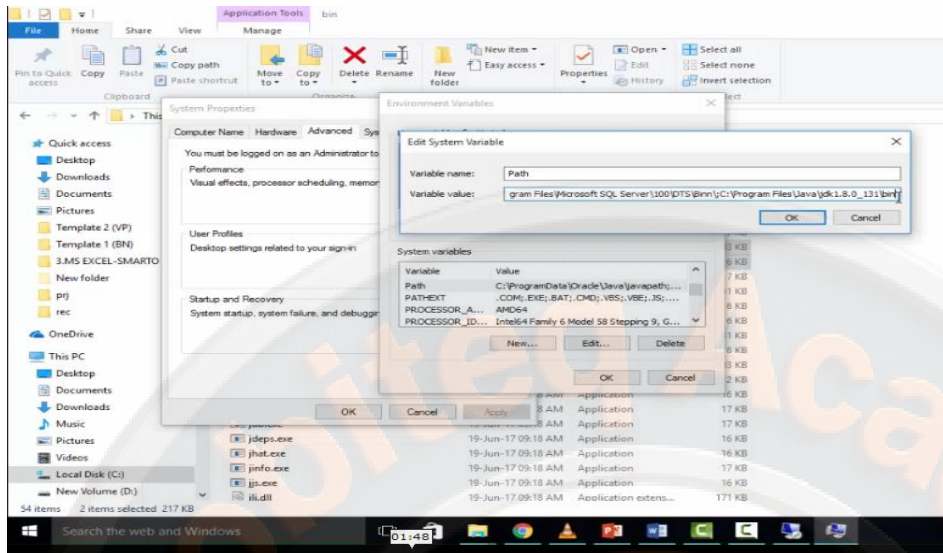
एक box available होगा जिसमे एक path available है हम इस path में कही भी क्लिक करके keyboard से end key को press करे तो हम last में पहुंच जाएगे path को set करने के लिए हमने जहा java को bin में install किया है



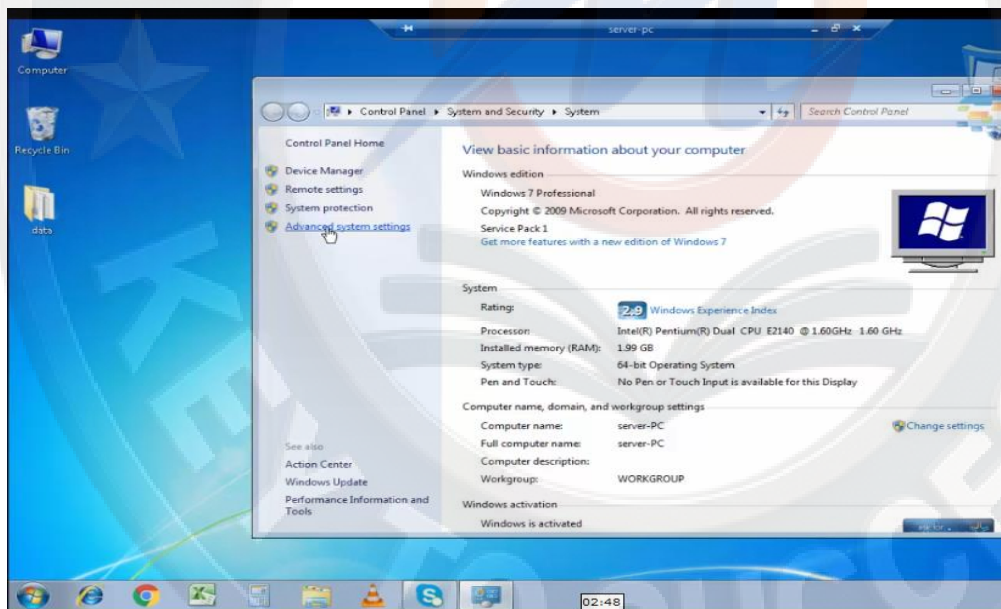
यहा से हमे directory के path को copy करना है



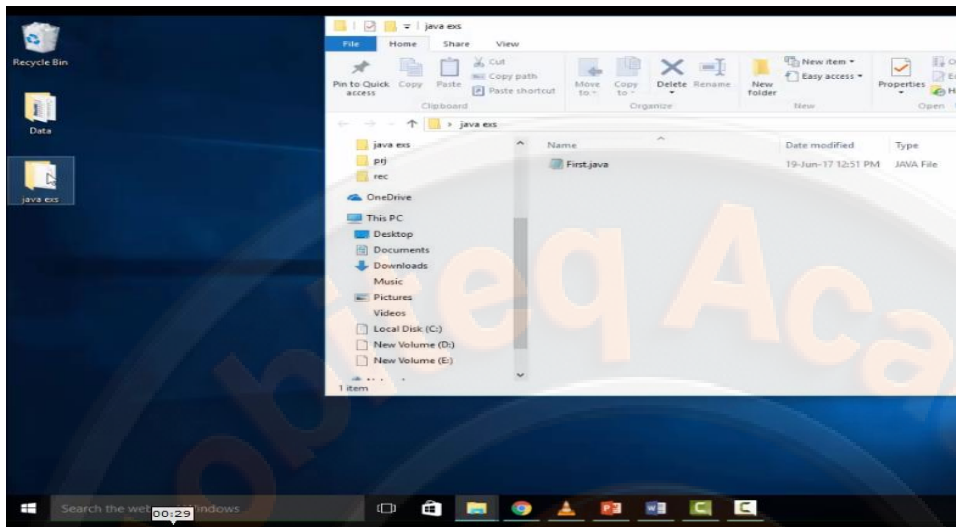
हमने last में क्लिक किया तो एक path पूरा select हो जाएगा हम right क्लिक करके copy करेगे



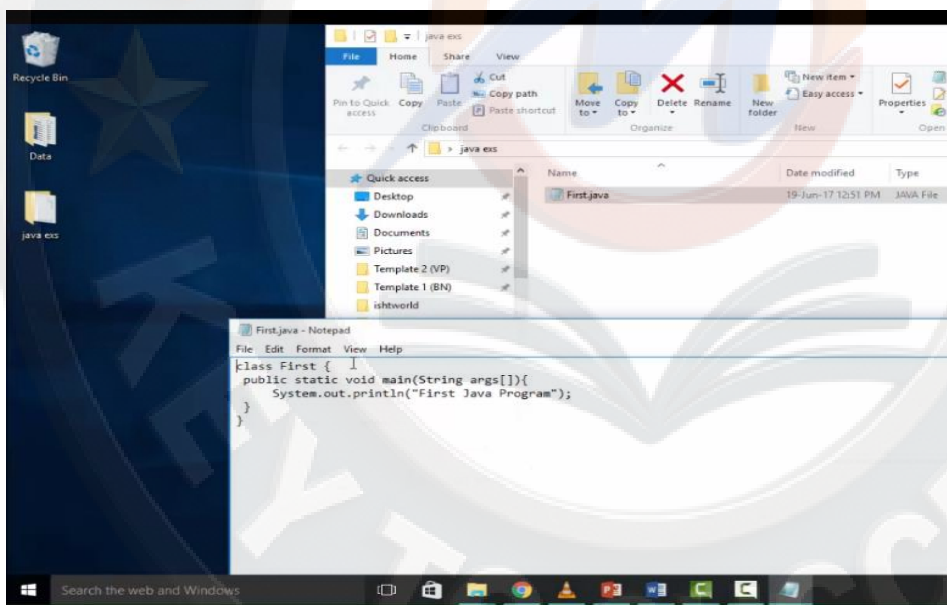
हमने path में जाकर Ctrl+V press करके paste किया और last में semicolon जरूर दे ताकि यह line end हो जाए हमने path को इस तरह से edit किया और ok किया और फिर ok किया अब हम java को आपने कंप्यूटर में use कर सकते है



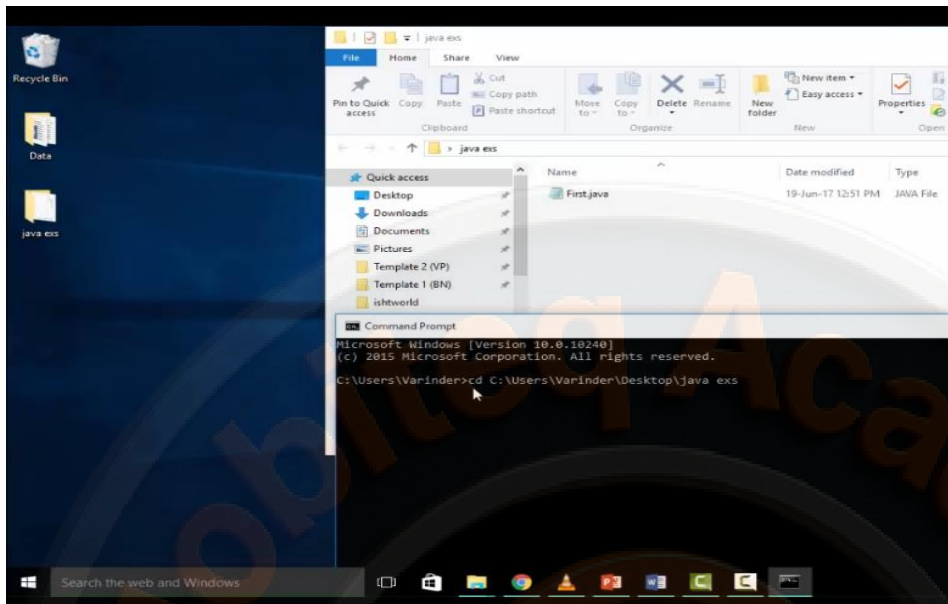
अगर Windows7 में setting करना चाहते है तो my computer पर right क्लिक करके properties option पर क्लिक करे तो box available होगा यहा से advance system setting पर क्लिक करके open करे तो एक box available होगा आगे का process Windows10 की तरह है इस तरह से Windows7 में path की setting कर सकते है



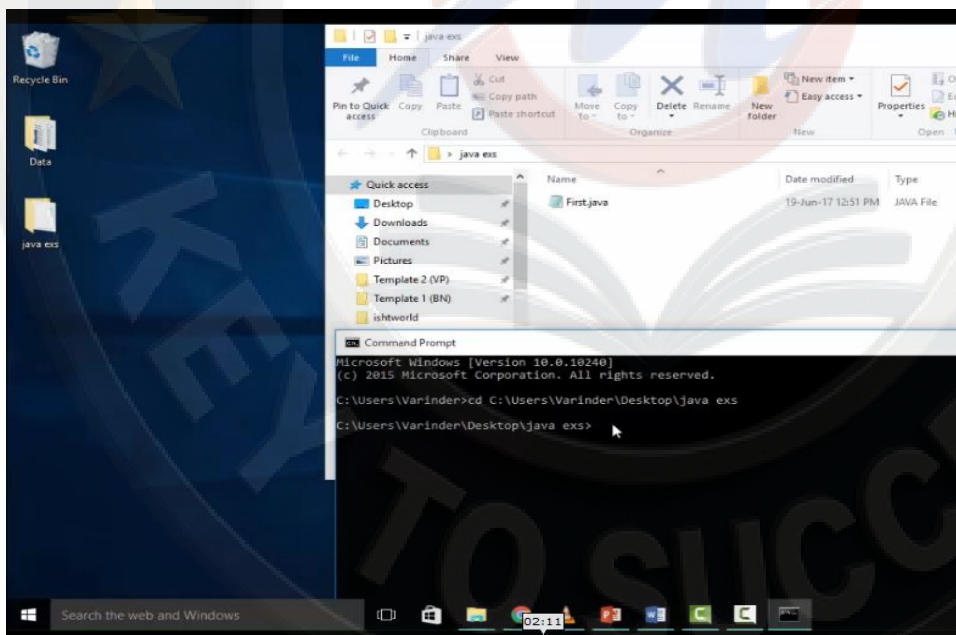
यहा पर हम program को run करने की जानकारी लेंगे हमारे पास java exe का एक फोल्डर available है हमने उसे क्लिक करके open किया इसमें हमारे पास first program available है



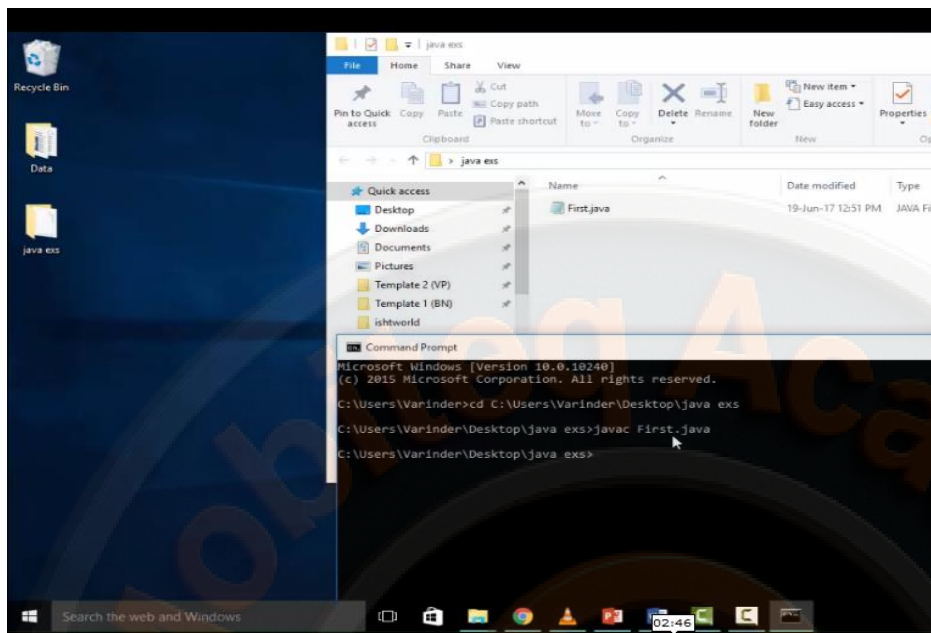
हम double क्लिक करके इसे देख सकते हैं यह एक simple program है जिसमे एक message print करवाया गया है हमने address bar पर क्लिक करके जो path select है उस path को copy किया है हमने program को run करने के लिए command prompt की जरूरत है हमने search box में command prompt type करके open किया



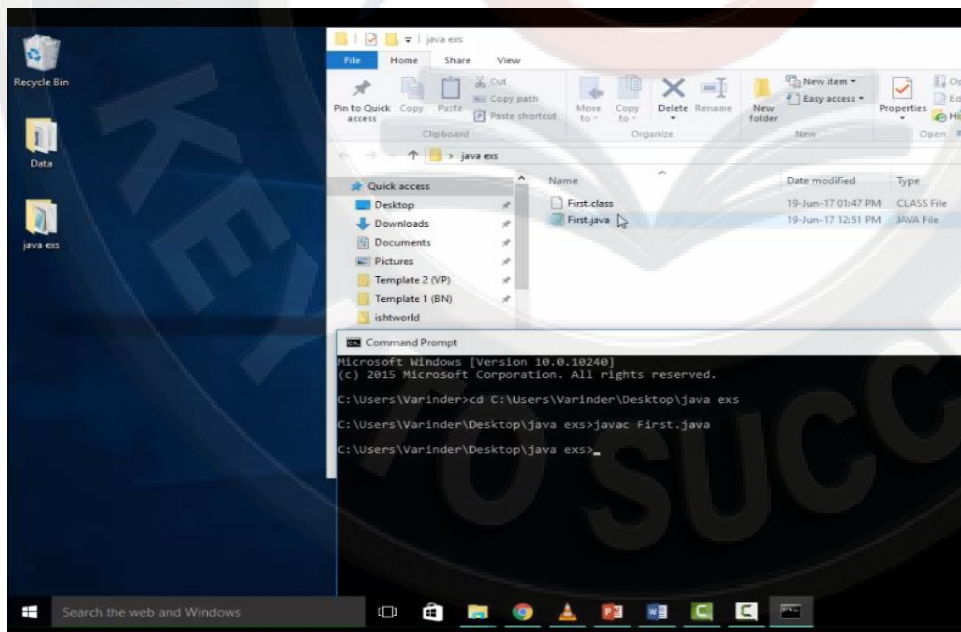
हमारे पास command prompt available है हमने यहा पर cd key को press किया है और Ctrl+V को press किया हम आपने ऊपर वाले फोल्डर में जाना चाहते है इस लिए हमने यह किया है इसलिए हमने address को यहा पर copy किया है हमने enter press किया



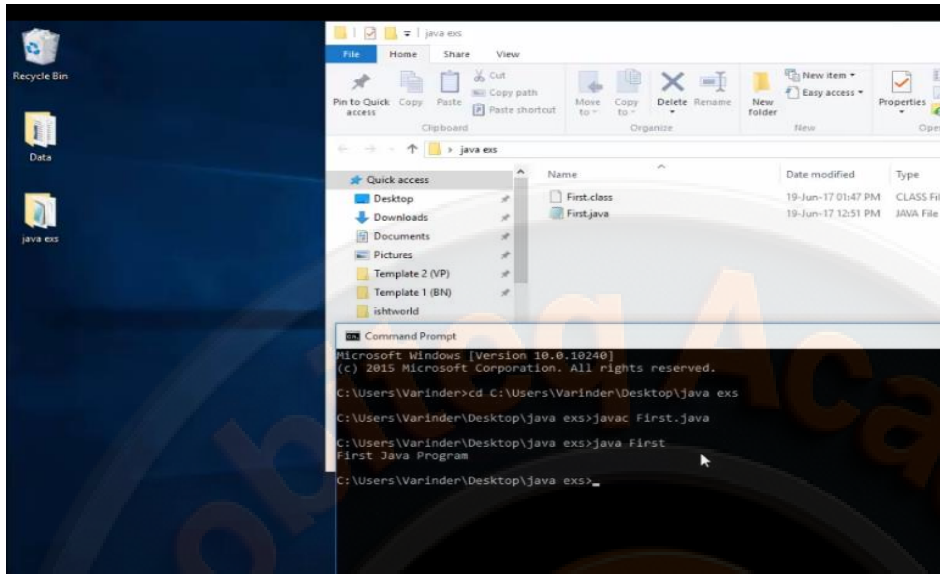
अब java फोल्डर के अंतर्गत आ चुका है java के program को run करने के लिए दो स्टेप में program को run करना होता है सबसे पहले program को compile करना जरूरी है उसके बाद interpreter के द्वारा run करना अर्थात execute करना जरूरी है अब हम program को compile करेगे इसके लिए हमने javac type किया जिसका अर्थ java compiler है उसके बाद जिस program को compile करना चाहते है उसका name देना है



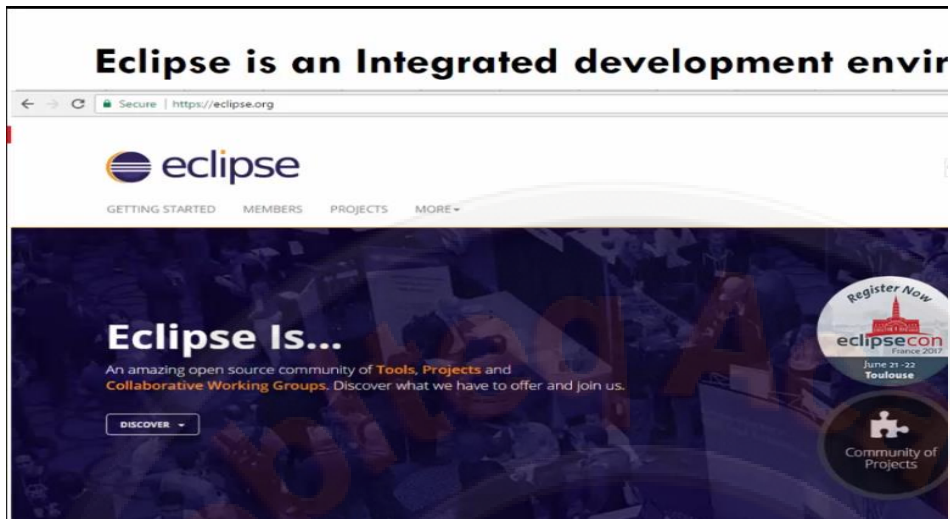
यहा पर हमारे program का नाम first. java है हमने type करके enter किया यदि हमारे पास कोई error नही है तो program सही compile हुआ है



अगर हम ऊपर देखे तो एक class file generate हुई है यह bytecode है अब जब हमने program को compile कर लिया है तो हमे इसे execute करना है



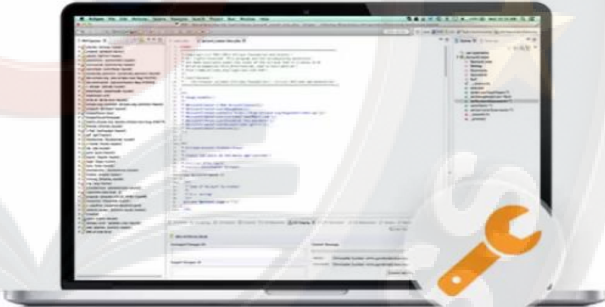
इसे execute करने के लिए java first type करके enter press किया हमें यहा पर extension का use नहीं करना है enter press करने के बाद हमारा प्रोगरण execute हो गया है जैसे की हमारी class file execute हो गई है जिसमे first java program का message available है इस तरह हम आसानी से program को compile करके execute कर सकते है हमने compiler और interpreter दोनों का use किया है हमे सबसे पहले javac से program को compile करना है उसके बाद एक class file generate होगी उसके बाद java first interpreter का use किया है enter करते ही program execute हो जाएगा हम इसके इलावा IDE से program को execute कर सकते है जोकि हम eclipse के रूप में प्रयोग करेगे



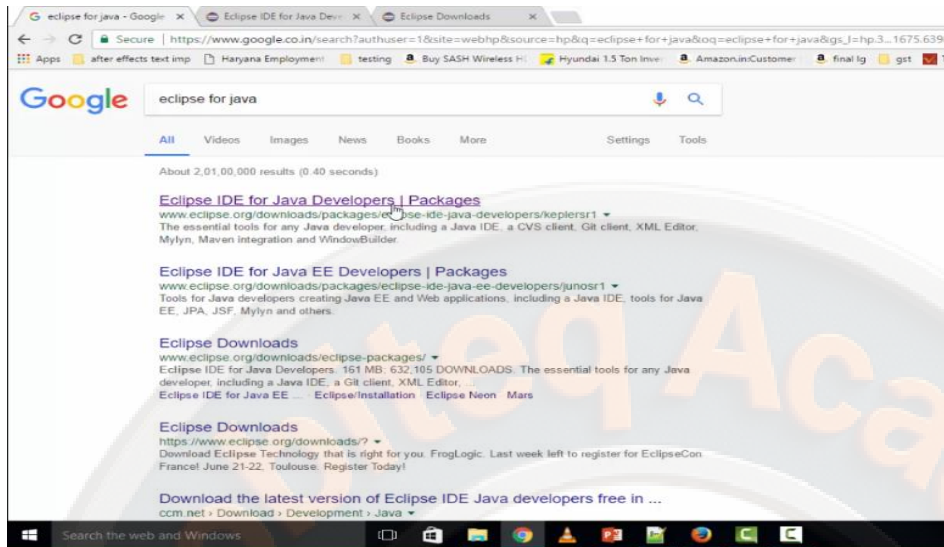
हम आपने java program को run करने के लिए IDE का use कर सकते हैं इसमें eclipse best है eclipse एक software है यह एक open source है जिसमें java के program को run कर सकते हैं यह बिलकुल free open source download available है

Tools and IDEs

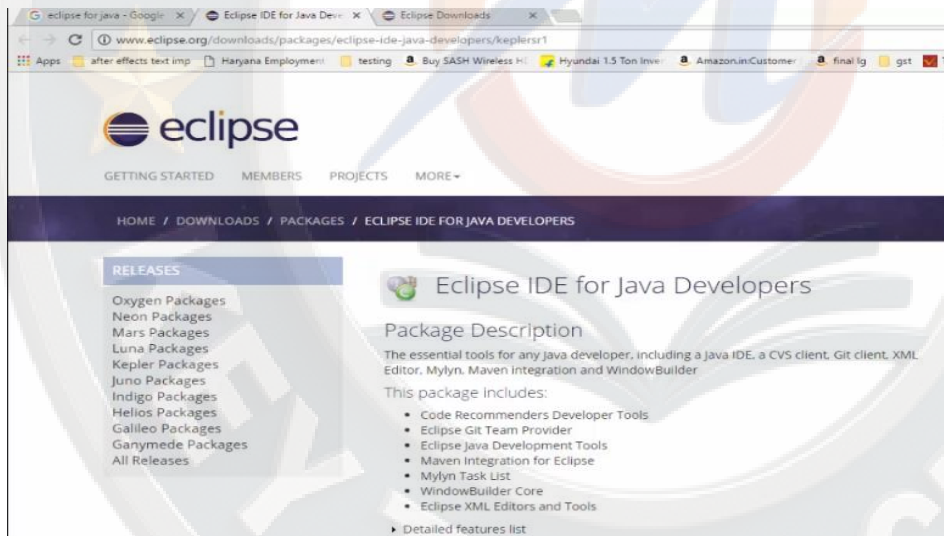
It has various tools and IDEs for Java, C/C++, JavaScript and PHP IDEs and it also covers many more languages.



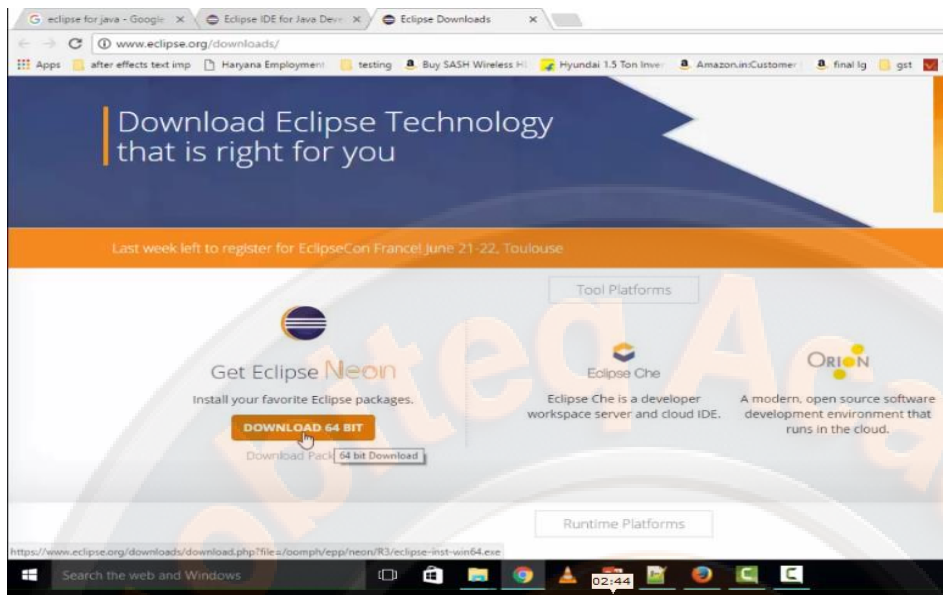
eclipse हमें बहुत सारे tools provide करता है जिससे हम आसानी से java program को run कर सकते हैं यह केवल java के लिए ही नहीं बल्कि other language के लिए भी available है



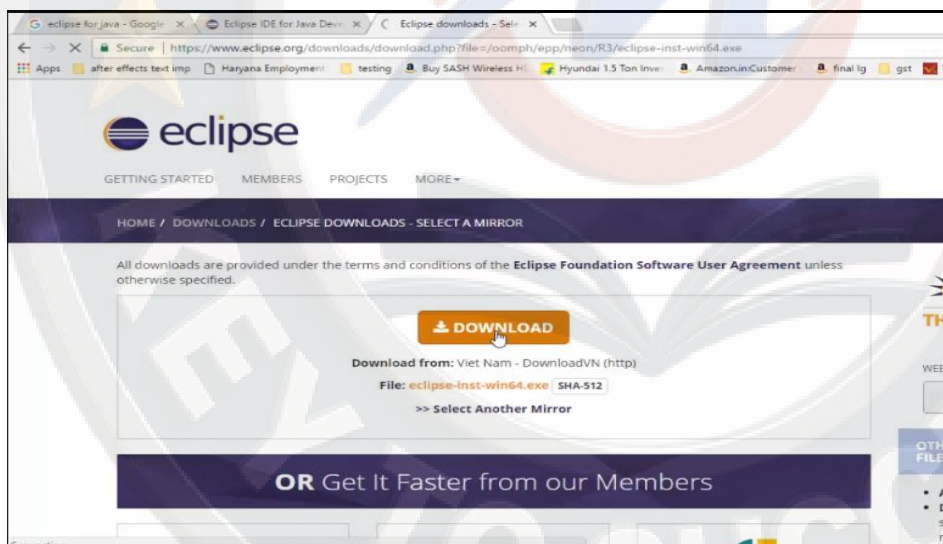
हमने eclipse को डाउनलोड करने के ली eclipse for java type करके search किया तो हमारे पास रिजल्ट available हो गए है हमने 1st link पर क्लिक किया



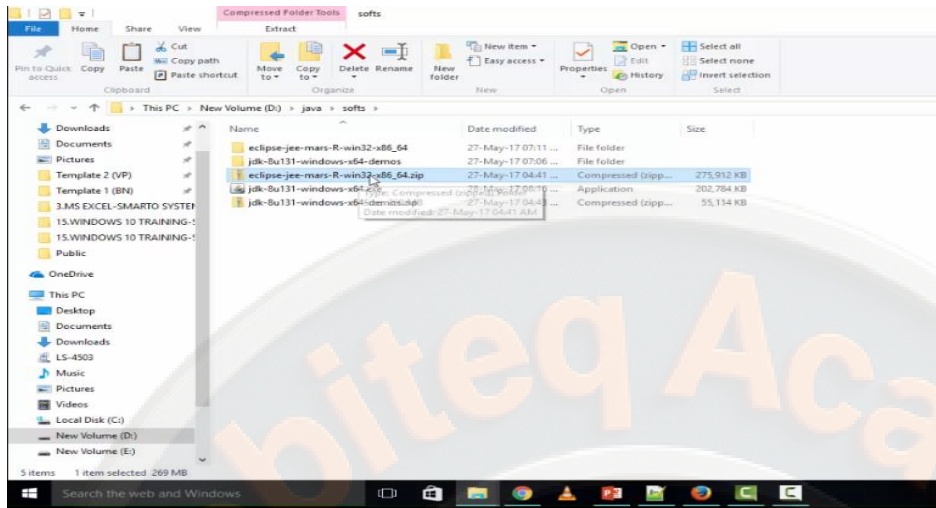
हमारे पास एक वेबसाइट open हो गई है इसमें हमारे पास काफी link available है हमने यहा से download पर क्लिक किया आप बिट के अनुसार ही डाउनलोड करे



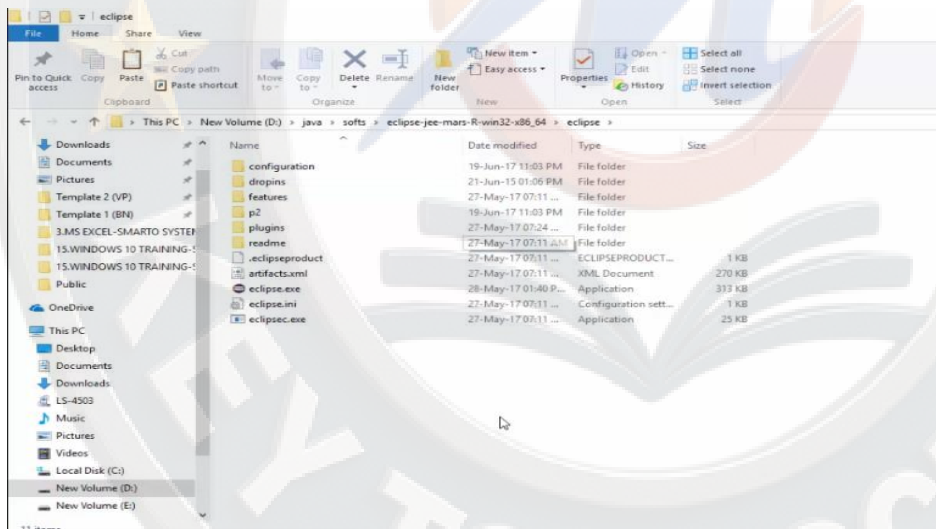
हमें eclipse ने direct 64 बिट डाउनलोड का option दे दिया है हमें इस पर क्लिक करना है तो हमारे पास downloading शुरू हो जाएगी



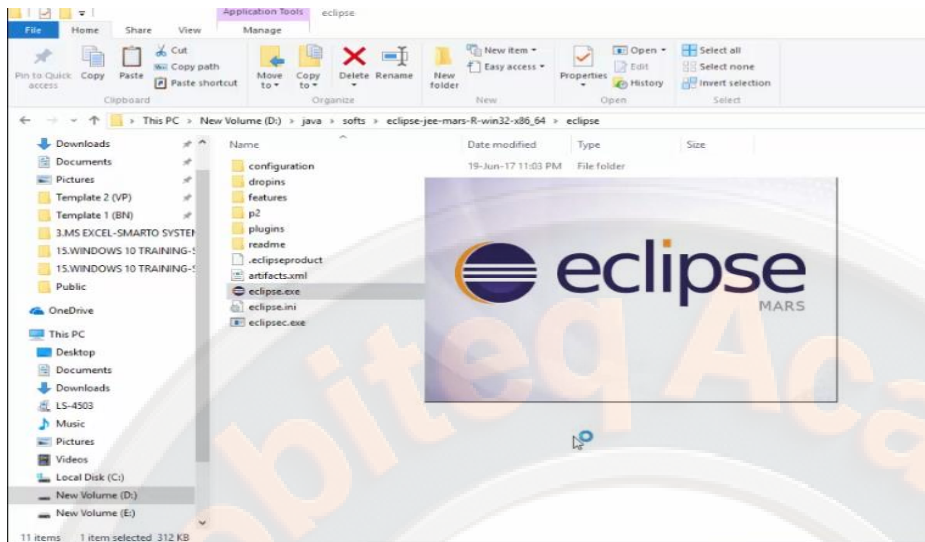
हमारे पास final link available है हम क्लिक कर देते हैं तो हमारे पास डाउनलोड होना शुरू हो गया है



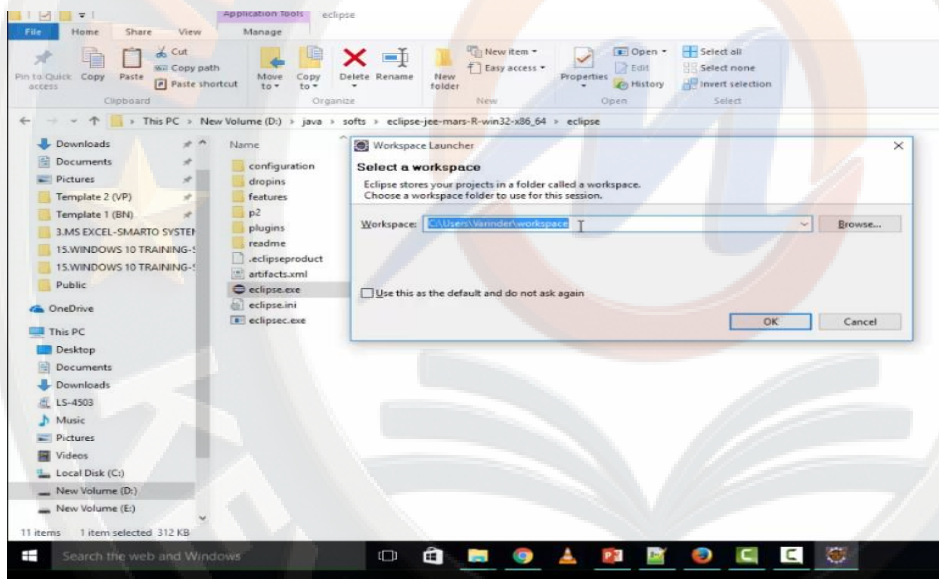
हमारे पास यहा पर zip में available हो गया है इसे unzip करने के लिए right क्लिक करके extract all option पर क्लिक कर सकते है तो हमारे पास unzip फोल्डर available हो जाएगा हमने उसे double क्लिक करके open किया फिर से क्लिक करके open किया



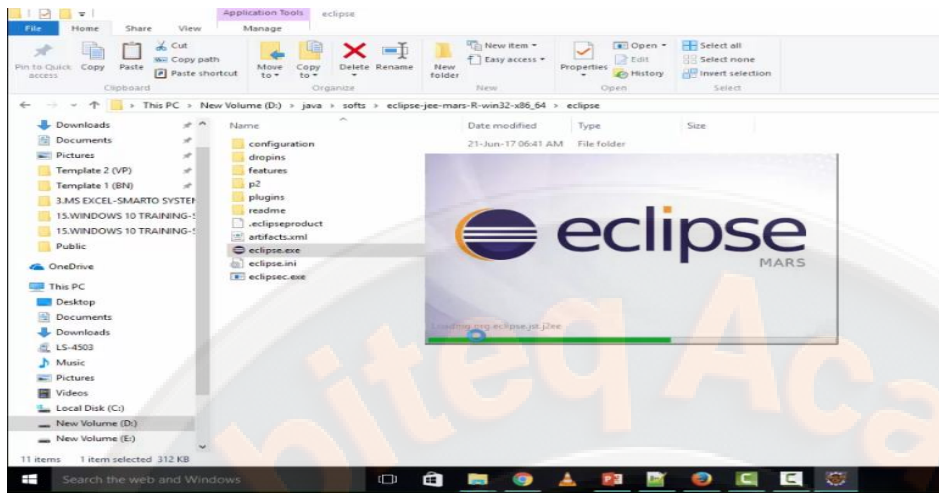
अब हमारे पास eclipse runnable version में available है हमे eclipse को अलग से install करने की जरूरत नहीं है eclipse पहले से ही portable version में available है



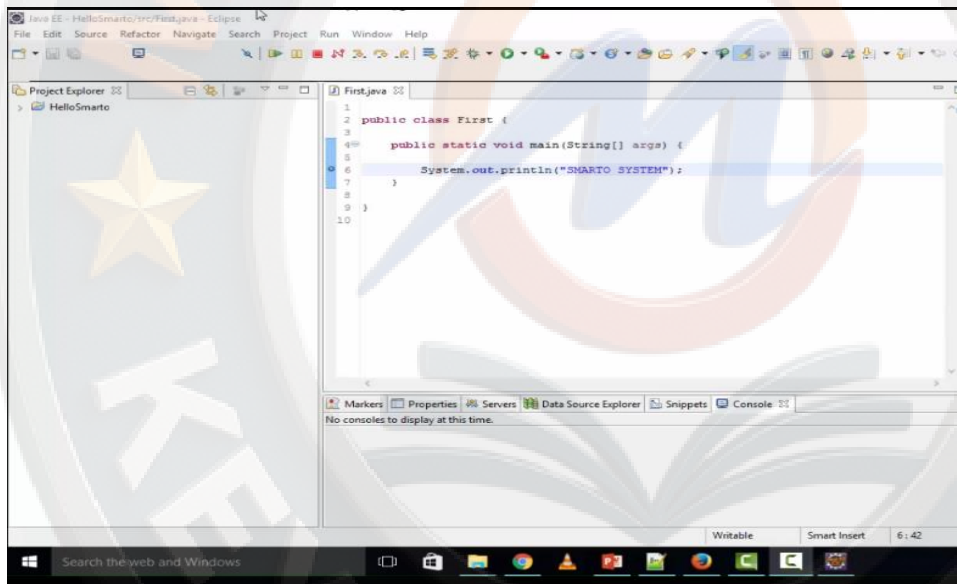
हमें eclipse.exe पर double क्लिक करना है तो eclipse open हो जाएगा



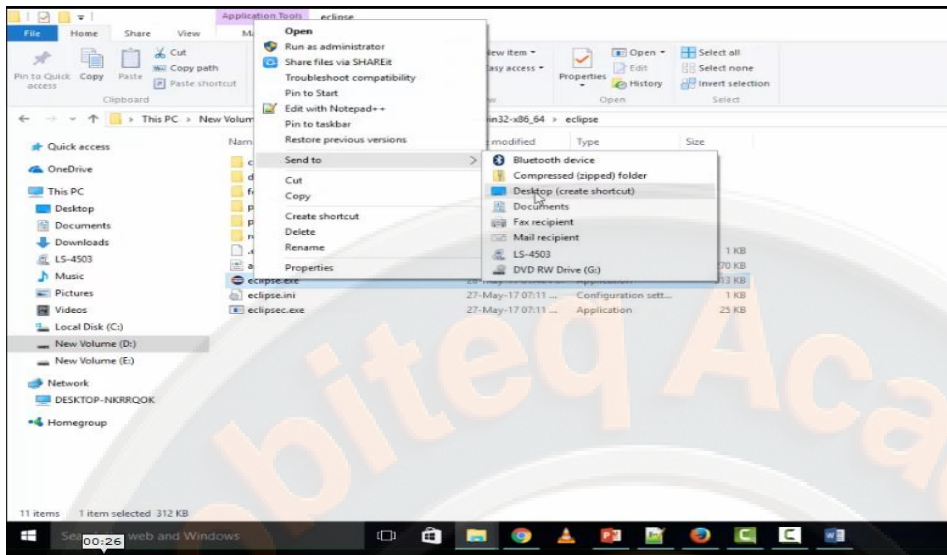
जब हम eclipse को 1st time open करते है तो workspace का setup करवाया जाता है workspace वह जगह है जहा पर हम projects को store करते है अगर आप पहले से java के लिए किसी फोल्डर का use कर रहे है तो browse करके उस फोल्डर पर जा सकते है उसको as a workspace use कर सकते है हमने इसी workspace को ok कर दिया है



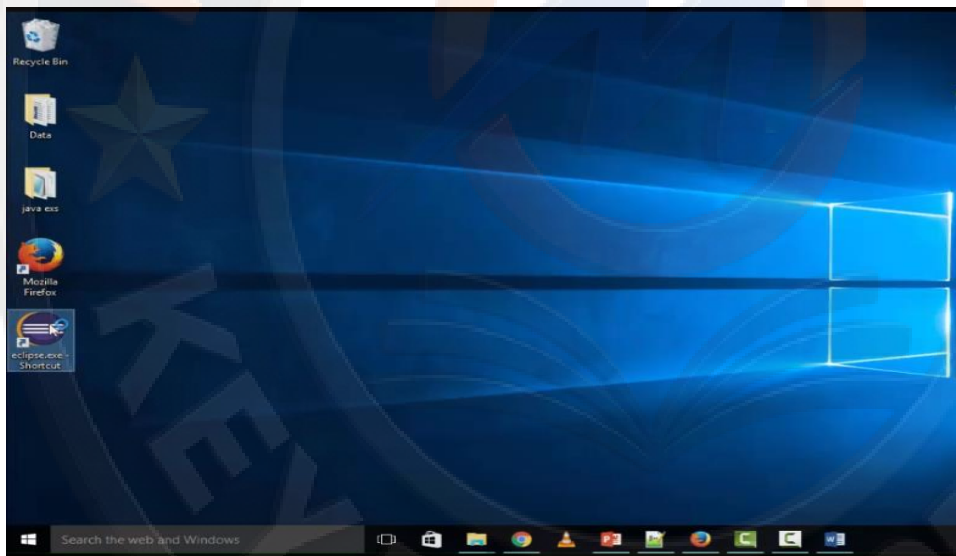
eclipse आपने interface को open करने से पहले कुछ setting करेगा कुछ setting eclipse के द्वारा की जा रही है



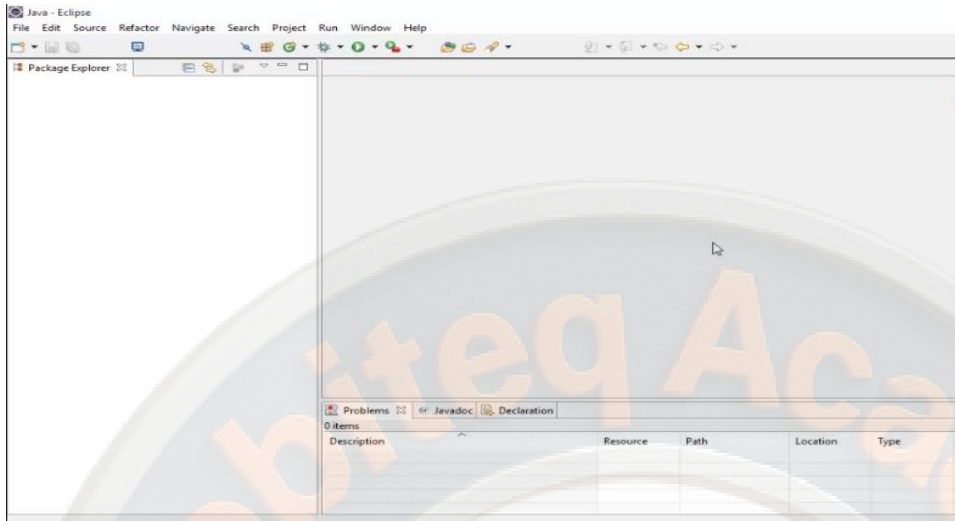
यहा पर हमारे पास eclipse available हो गया है इस तरह से हम eclipse को डाउनलोड करके use कर सकते है



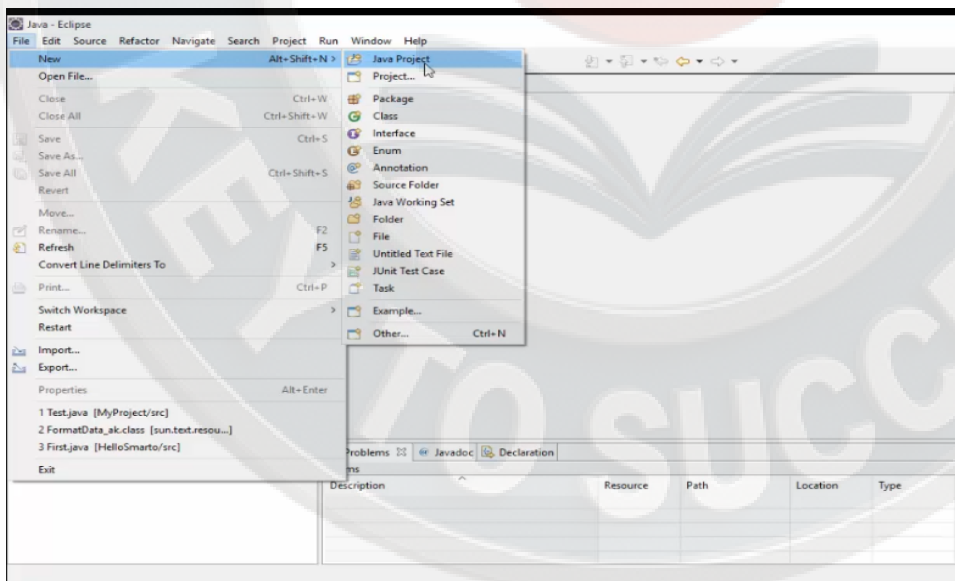
हमने eclipse.exe पर right क्लिक करके एक shortcut बनाया है हमने close किया



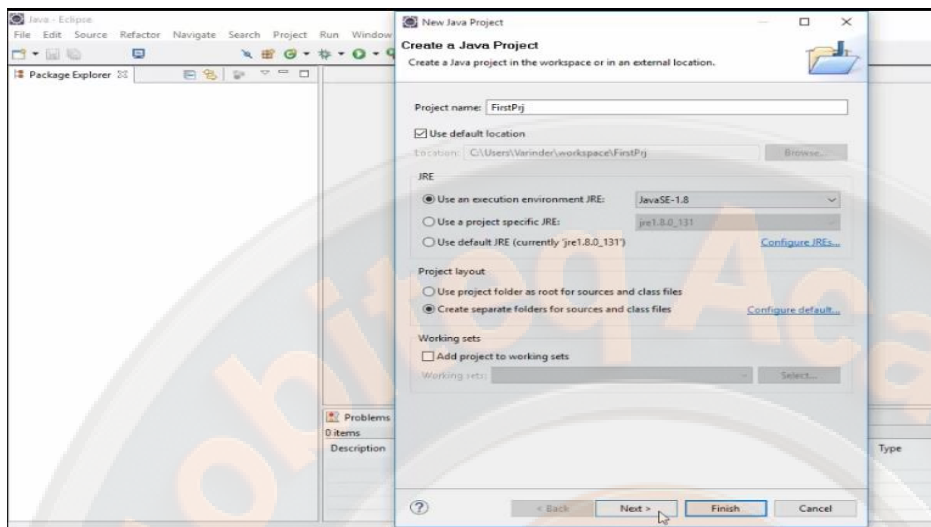
हमारे desktop पर shortcut available है हमने double क्लिक करके open कर लेते हैं



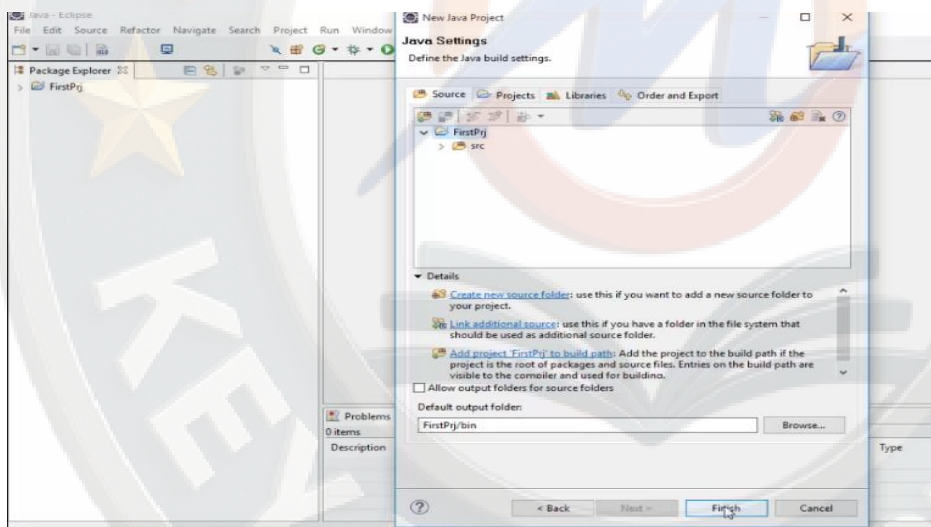
हमारे पास eclipse available है इसमें काम करने के लिए एक project डिफाइन करना होता है ताकि हम काम को आसानी से कर सकें



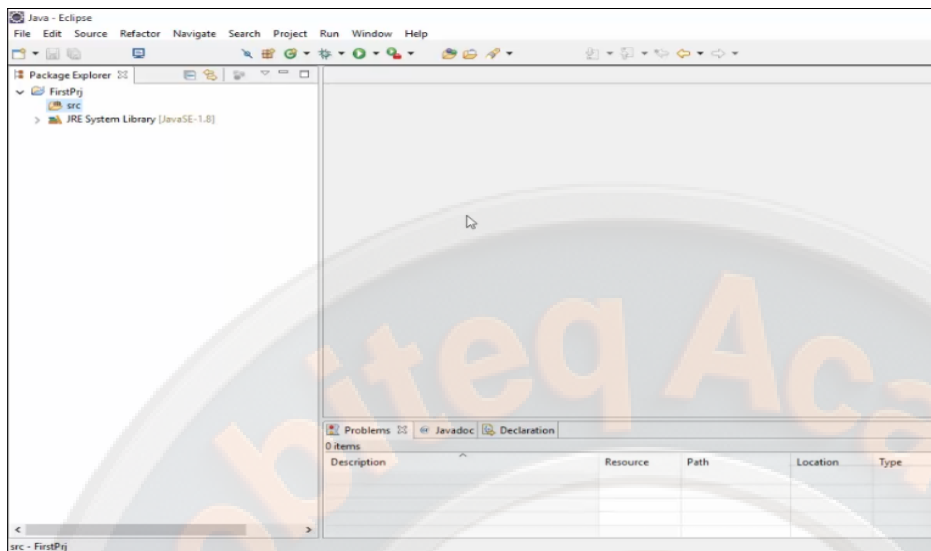
इसके लिए file menu पर क्लिक करके new में से java project पर क्लिक करते हैं



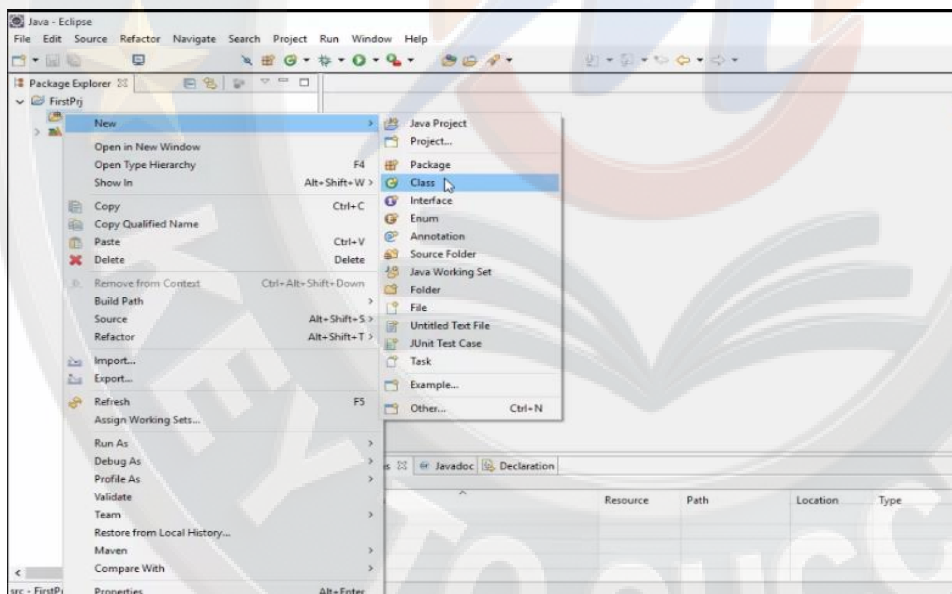
एक box available होगा यहा से हमने project name type किया और कुछ setting करके next पर क्लिक किया



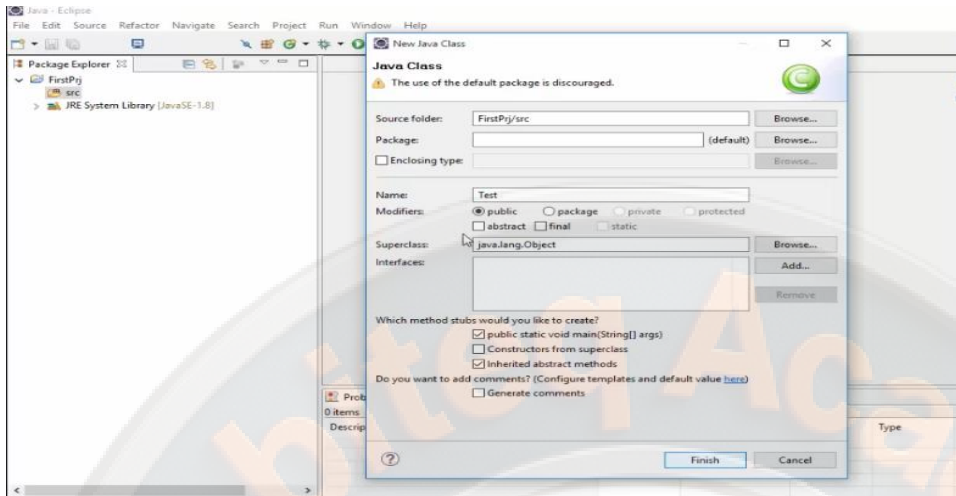
यहा पर एक फोल्डर create हो गया है और उसके अंतर्गत एक source फोल्डर भी create किया गया है यह कार्य eclipse ने खुद किया है जितनी भी files हम type करते है जोकि java files होती है वह सभी files source फोल्डर के अंतर्गत मिलेगी और जब उन्हें compile कर लेते है तो इसी फोल्डर में bin में वह files मिलेगी जोकि हमारी class files होगी हमने finish पर क्लिक किया



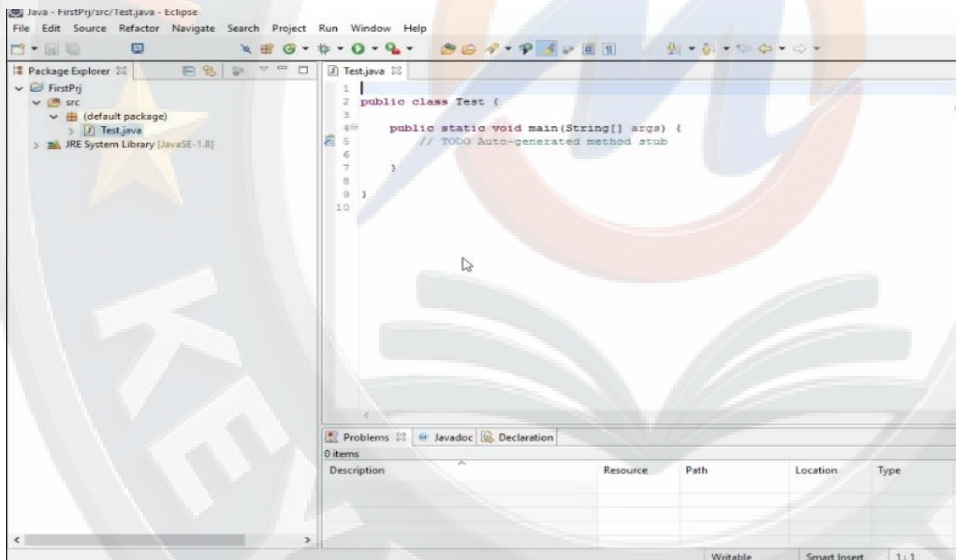
हमारे पास left side project available हो गया है जो हमने नाम रखा था



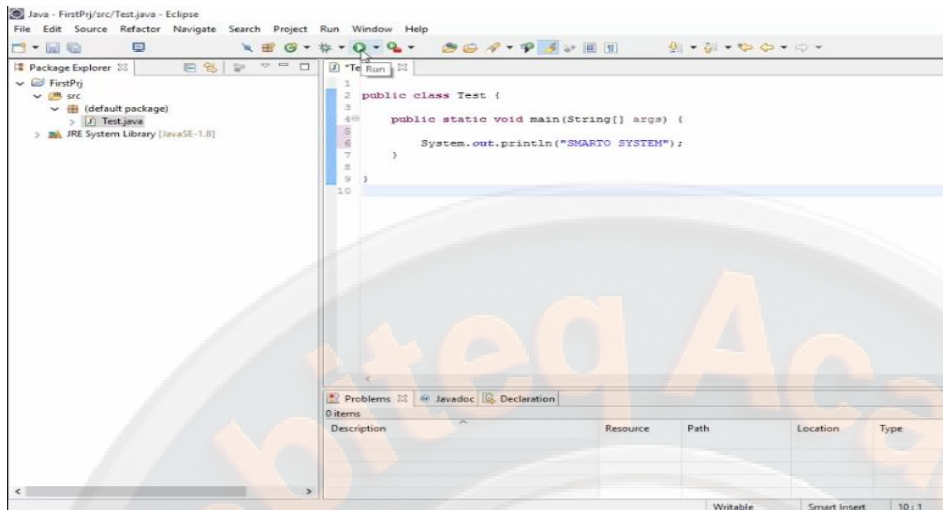
हमारे पास source फोल्डर available है इसमें हम java file create करनी इसके लिए class को डिफाइन करना है इसके लिए हमे eclipse में coding की जरूरत नहीं है हम src पर right क्लिक करके new पर जाकर class पर क्लिक करते है



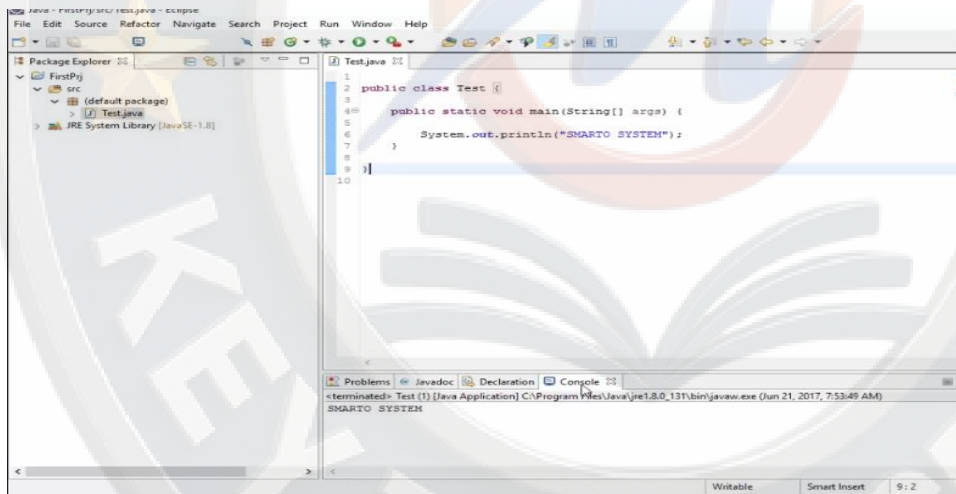
एक box available होगा यहां पर हमने class name Test type किया हमने कुछ setting करके finish कर दिया है



यहां पर हमारे पास left side पर Test.java एक file create हो गई है और हमारे पास coding available हो गई है यहां पर जो हमारी class का नाम है वह हमारी file name भी है

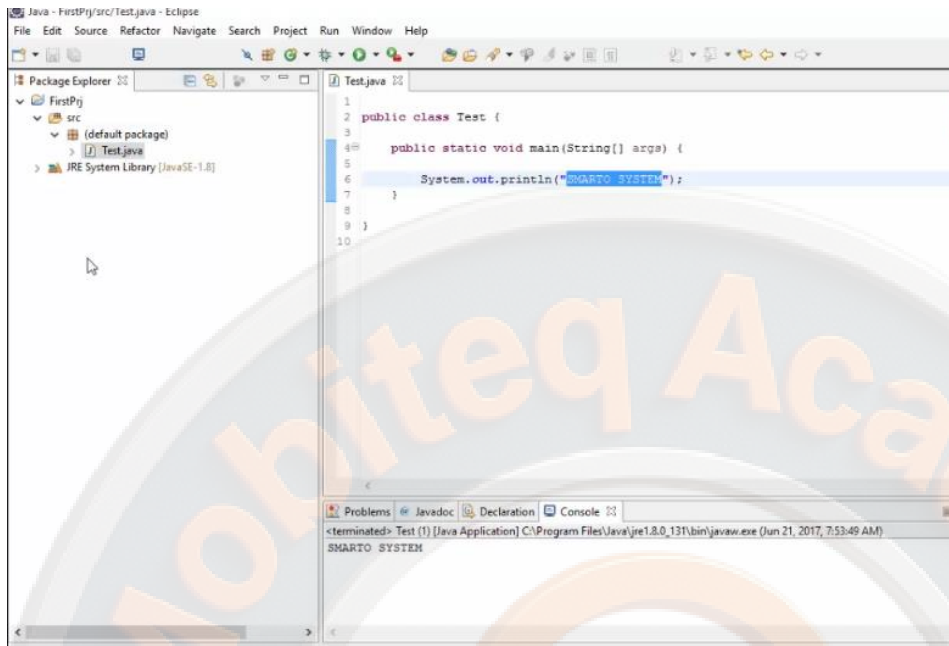


हमने यहा पर एक program डिफाइन किया है इस program को run करने के लिए ऊपर run बटन available है उस पर क्लिक करते ही एक box available होता है उसे ok करते है तो हमारा program run हो गया है



इसकी जानकारी हमे console से मिलती है हमारे पास नीचे console panel available है जहा पर SMARTO SYSTEM लिखा हुआ आ गया है

इस तरह से आप eclipse में एक छोटा सा program run करके देख सकते है कि eclipse ठीक से कार्य कर रहा है इस तरह से eclipse को verify करके देख सकते है



हम eclipse के interface के बारे में जान लेते हैं जब हम program बनाते हैं तो एक package और source डिफाइन करना पड़ता है यह सभी चीजे left side package explorer में show की जाती है जिस भी program को right क्लिक करके open करते हैं वह tab के रूप में बीच में ऊपर available होता है अगर एक से ज्यादा program को डिफाइन करते हैं तो यह tab के रूप में available हो जाती है इसी तरह right side पर task list available रहती है और नीचे की तरफ program का आउटलाइन view रहता है जिसमें main main function की जानकारी दी जाती है हम यहा से क्लिक करके access कर सकते हैं और नीचे की तरफ अगर program में कोई problem है तो उसे बताया जाएगा इसी तरह other option भी available है यहा सबसे important console रहता है यदि program में कोई error रहता है या compile नहीं होता है तो console में उसके बारे में जानकारी दी जाती है और अगर output बिलकुल ठीक है तो वह भी show किया जाएगा इसी तरह ऊपर कुछ menu available हैं जिस पर क्लिक करे तो options available हो जाते हैं जोकि basic text editor की तरह हैं

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project named 'FirstPj'. The main editor window displays the code for a Java class named 'Test'. The code is as follows:

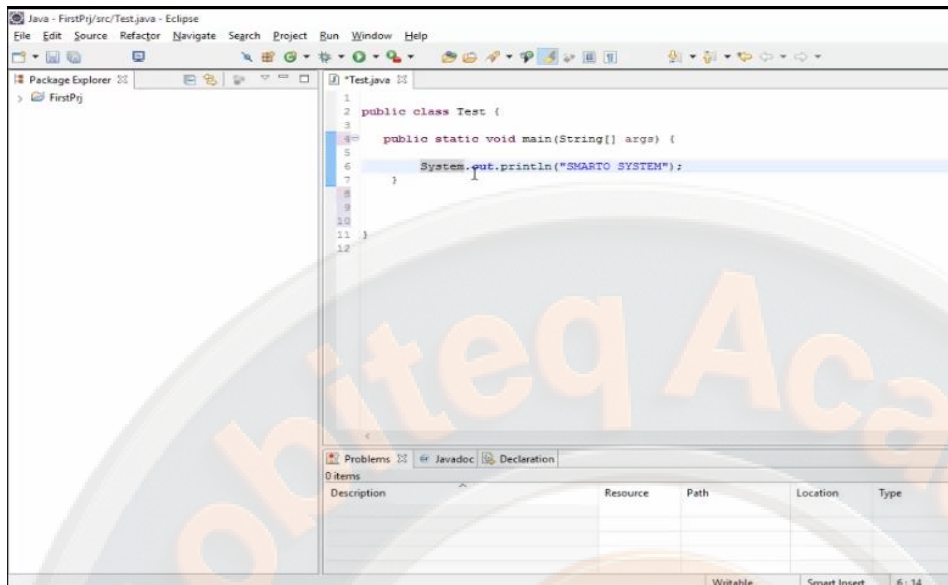
```

1 public class Test {
2
3
4
5
6
7
8 }
9

```

Below the code editor, there is a 'Problems' tab which is currently empty. The status bar at the bottom right indicates 'Writable', 'Smart Insert', and '3 : 1'.

अगर हमे java में कोई program बनाना है तो सबसे पहले class को डिफाइन करना है उसका एक नाम होना चाहिए उसके बाद open curly ब्रैकेट और last में ending curly ब्रैकेट का use किया जाता है तो इस तरह से इस section के अंतर्गत हमे program डिफाइन करना है जो भी class का name हम रखते है उसी नाम से file का name भी रहता है जैसे कि class का नाम test है तो file का नाम भी test है जब भी हम class का नाम डिफाइन करे तो first letter capital होना चाहिए हमे java के rules को मानना चाहिए open ब्रैकेट के बाद code डिफाइन किया जाता है java के program को run करने के लिए void main method use किया जाता है main function का नाम है main के साथ इसकी बाँडी में जिसको हम execute करना चाहते है उसे शामिल कर सकते है



```
1 public class Test {
2
3
4     public static void main(String[] args) {
5
6         System.out.println("SMARTO SYSTEM");
7     }
8
9
10
11 }
12
```

Description	Resource	Path	Location	Type

Writable Smart Insert 6: 14

हमने यहा पर एक line print करवाना चाहते है इसके लिए हम system name की predefind class का use करते है इसके साथ out एक object डिफाइन है उसके बाद println function का use कर सकते है उसके बाद value को double quotes में रखते है उसके बाद main बाँडी को close किया गया है और फिर class को भी close कर दिया है इसको हम run कर सकते है इस तरह से यह एक basic program है जिसमे हम कुछ print करवा सकते है

CLASS AND OBJECT



अब class और object के बारे में जानेगे हमने अभी जो example देखा था उसमे public class का use किया था java में सबसे important class और object होते है हम इन्हें इस example से समझते है हमारे पास इसमें एक simple car है जोकि class car है इसी तरह नीचे तीन car ओर भी है जोकि अलग अलग कम्पनी की है यहा पर हमारे पास एक basic template है वह class car है एक car में जो होता है वह हम class से डिफाइन करते है जैसे कि एक car में four wheel होने चाहिए एक body उसमे स्टेरिंग आदि होना चाहिए यहा पर हमने केवल template डिफाइन किया है उसमे यह नही बताया कि कलर क्या चाहिए, speed क्या चाहिए हमने केवल बताया है कि car कैसी होती है लेकिन जैसे ही इसको implement करने की बात आती है तो object के रूप में हम उसे implement करते है नीचे हमने eon object के रूप में implement किया है जिसमे कलर red है इसकी कम्पनी hundui है और इसमें काफी सारे features है इसी तरह nano object और maruti object भी है जोकि class car की implement है इस तरह हमने एक class car को तीन तरह से implement किया है जोकि object1, obj2, obj3 है इस तरह class एक blue print या map की तरह है जोकि एक basic structure है लेकिन जब इसे implement किया जाता है इसमें new feature होते है तो वह उसका object बन जाता है

CLASS AND OBJECT

- Java is an object oriented language, which is based on classes and object. It is a a strong **Object oriented programming language(oops)**.
- Oops is basically a programming approach which provides real world based solutions to complex problems.
- Because java is based on oops it follows all Features of OOPS
 - Inheritance
 - Polymorphism
 - Encapsulation
 - Abstraction
- Java is purely based on classes and objects.
- **Class, objects, fields and methods** are the basic blocks of object-based Java programming.

java object oriented language है object oriented language वह language होती है जो language class और object पर based रहती है इसे oops कहते है जोकि short form है यह class और object process को follow करती है oops एक तरह से programming approach है जोकि real world के example से मिलता है java object oriented के सभी features को follow करती है इसमें मुख्यतः four features होते है java class, objects, fields और method को follow किया जाता है हम इनके बारे में जानेगे

जैसे कि हमने class के बारे में जाना कि class एक template है जिसमें मुख्यतः दो चीजें शामिल होती हैं उस class की properties और behaviors शामिल हैं properties यहाँ पर variables जहाँ पर हम data को store कर सकते हैं और behavior से मतलब implement से है जोकि method है इस तरह

CLASS AND OBJECT

- **Class** : A class can be defined as a template/blueprint which has its properties (variables) and behaviors (methods), used to create its kind of objects. Everything is encapsulated in Java under classes.
- **Object**: An object is an **instance** of a class, physical existence of a class. Class behaviors and workout implementation comes under Object.

एक class होती है कोई भी information java की classes में encapsulated अर्थात् पूरी की पूरी छिपी रहती है जब तक class परमिशन नहीं देती है हम इस information को बाहर access नहीं कर सकते हैं यह classes का सबसे बड़ा feature है

Object :- object एक तरह से class का instance या एक copy होती है लेकिन यह एक physical existence होता है अर्थात् इसे practically implement किया जाता है class के पास जितने behaviors रहते हैं उन्हें practically implement object के द्वारा किया जाता है

इस तरह class हमारे पास एक guide map या blueprint रहता है और object इसका practical implement रहता है यह class की copy रहता है

fields (variables) and behaviors (methods)

Basic class in Java has :

fields - Variables : used to store data , every object can have its own data stored in these fields or variables.

Class Car can have modelNo, color, speed

Methods - Functions : used to define set of instruction to do something specific. Object can call them to execute that specific work.

Class Car can have setSpeed() method to increase or decrease speed in car

Blocks – brackets : are used to define body of particular class or method.

अब हम fields और method के बारे में जानेगे हम java में class की बात करे तो class में दो चीजे fields और behavior रहते है java में function word का use ना करके method word use किया जाता है


fields-variables -: variable एक तरह से वह जगह है जहा पर java के अंतर्गत data को store करते है जैसे कि class car से related data store करना चाहते है जैसे कि car का model No, color और speed इस तरह की कोई भी जानकारी save करना चाहते है तो variables की जरूरत रहती है

Method -: method एक तरह से instruction का डिफाइन set होता है जो एक special कार्य करने वाला है यदि हम java की class में कोई खास काम करना चाहते है तो उसको method में डिफाइन करना है जैसे कि हम car की speed को set करना चाहते है तो यह एक method है जिसमे हम set speed method का use करेगे इससे हम speed को कम या ज्यादा कर सकते है

Block -: class में हमारे पास block भी शामिल रहते है blocks किसी class की body को डिफाइन करते है हम curly ब्रैकेट का use करते है

Class : A class can be defined as a template/blueprint which has its fields (variables) and **behaviours (methods)**, used to create its kind of objects. Everything is encapsulated in Java under classes.

Constructor is a special method having same name of class.



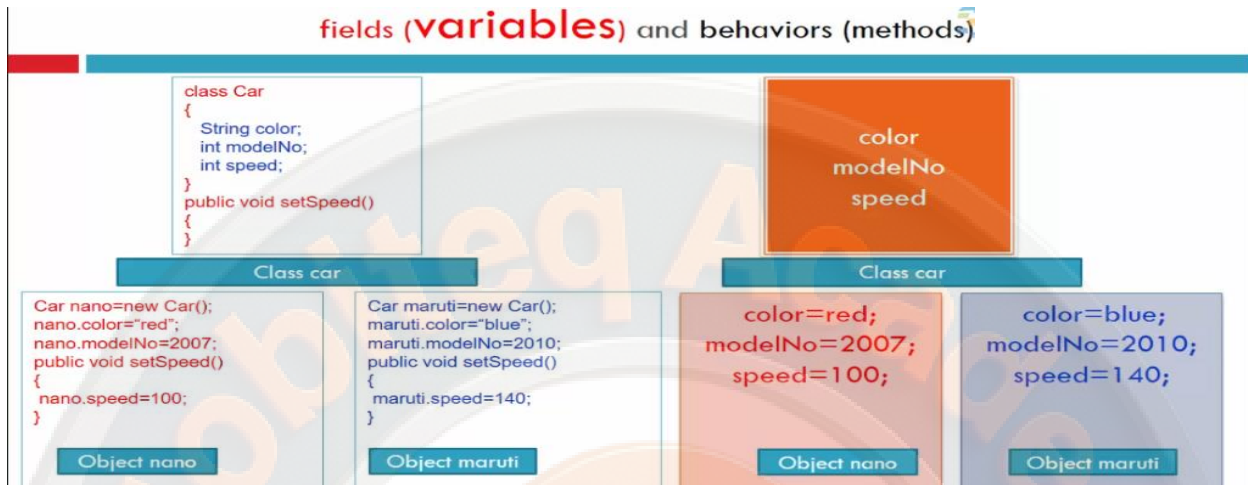
```
class Car
{
    String color;
    int modelNo;
    int speed;
}
public void setSpeed(int givenSpeed)
{ speed=givenSpeed;
}
```

```
Car nano=new Car();
nano.color="red";
nano.modelNo=2007;
nano.setSpeed(100);
```

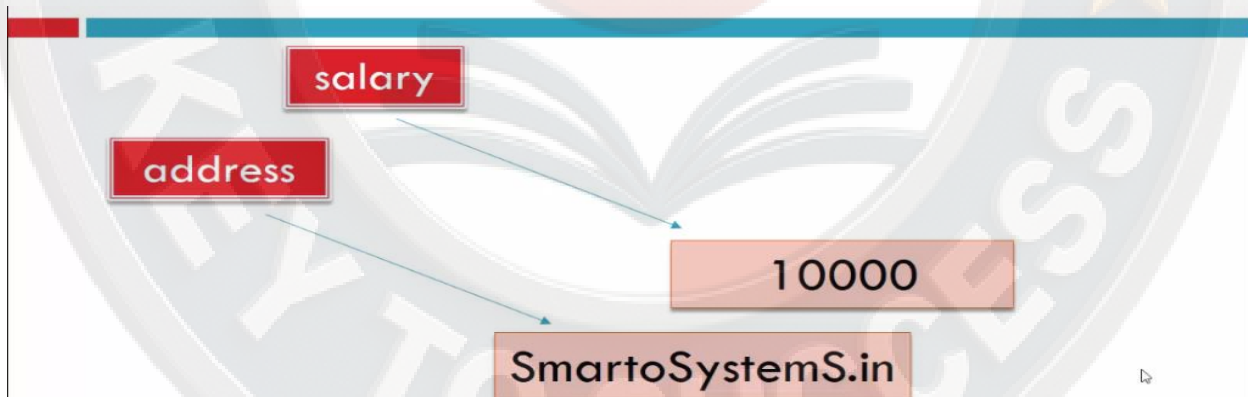
```
Car maruti=new Car();
maruti.color="blue";
maruti.modelNo=2010;
maruti.setSpeed(140);
```

हमने यहा पर class car बनाई है जिसमे तीन variable color, model no और speed है method बनाने के लिए method का नाम रखना होता है जोकि यहा पर set speed है इसके साथ हमे दो curly ब्रैकेट का use किया है इसमें हमने given speed use किया है जो भी speed इसमें देंगे वह class के speed variable में store होगी इस तरह जो object इसे call करेगा उसकी speed set speed के द्वारा डिफाइन की जा सकती है नीचे हमारे पास object nano है अब हमने car class के साथ nano object बनाया है nano में color और model no का information store किया है और यदि किसी method को call करना है तो nano.set speed यह पर dot method को call करने के लिए लगाया है हमने rounded ब्रैकेट के बीच information डाल दी है जो nano speed 100 set करना चाहते है यह speed given speed में जाएगी और given speed से यह speed variable में जाएगी इसी तरह हमने maruti object basic info को set करने के बाद maruti.set speed 140 set की गई है इस तरह हमने अलग अलग speed set की है

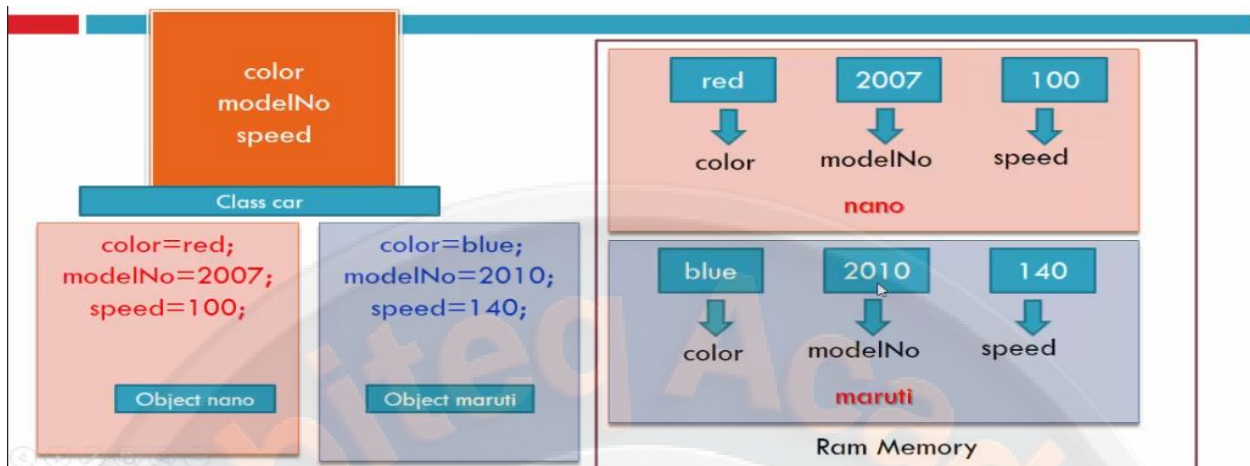
इस तरह set speed method car class का है परन्तु object इसे आपने लिए speed set के लिए use कर पा रहे है इस तरह से हम set of instruction को किसी भी method के अंतर्गत जिनको object के साथ call करके execute किया जा सकता है



अब हम variables के बारे में जानेगे अभी तक हमने basic variables को जाना है हमने यहा पर left side पर coding और right side पर एक guide map को show किया है हमने अभी coding को देखा था लेकिन guide map में हमने color, model no, speed को store किया गया है इस तरह variable data store करने में मदद करते है जोकि java में store होते है



जैसे कि हमे salary का data store करना है इसको हम particular memory location पर store करके name रख सकते है यहा पर 10000 value है जो एक memory location है और इस memory location का नाम salary है इसी तरह से address है इस तरह से हम data store करके location का नाम रख देते है तो इसे variable कहते है



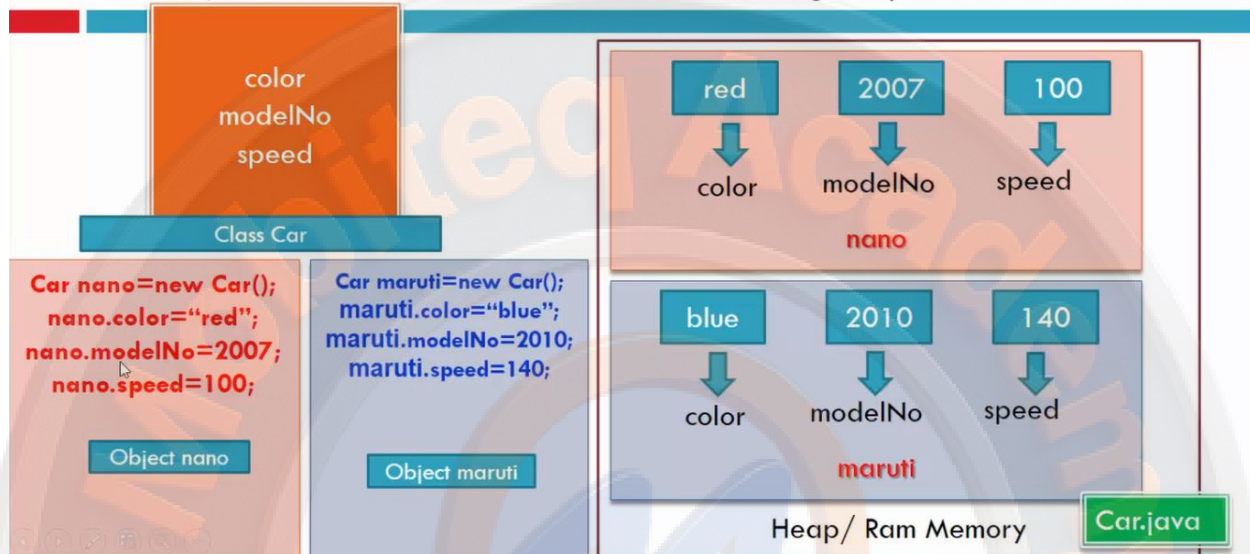
यहा पर जो हमारे object है वह अपना data अलग अलग जगह store करते है जैसे की nano object में color variable में red की जानकारी store कर रहा है model no में 2007 की और speed में 100 की इसी तरह से maruti object में भी data अलग अलग store है इस तरह यहा पर color variable का नाम और red data है और यह data nano object से जुड़ा है और नीचे का data maruti object से जुड़ा है

- In Java Programming language there are three kinds of variables.
 1. Instance variables/object based
 2. Static Variables/class based
 3. Local Variables/ within any method
 1. Parameters/within any method

यहा पर variables के types के बारे में जानकारी दी गई है जोकि three type के है

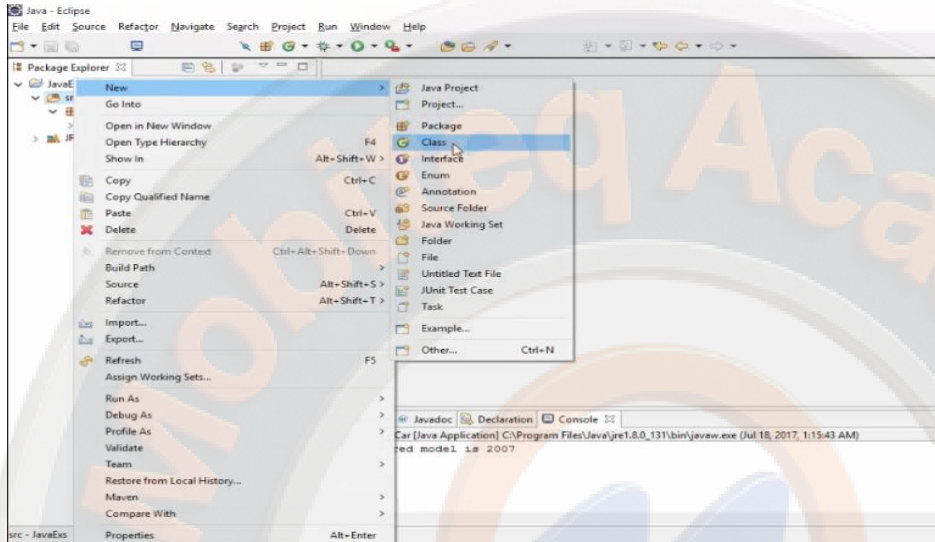


1. Instance variables/object based : declare in class but outside any method/ constructor. These variables are used by objects to store their data. Here every object can store their data in its variables and can get any time when it needs.

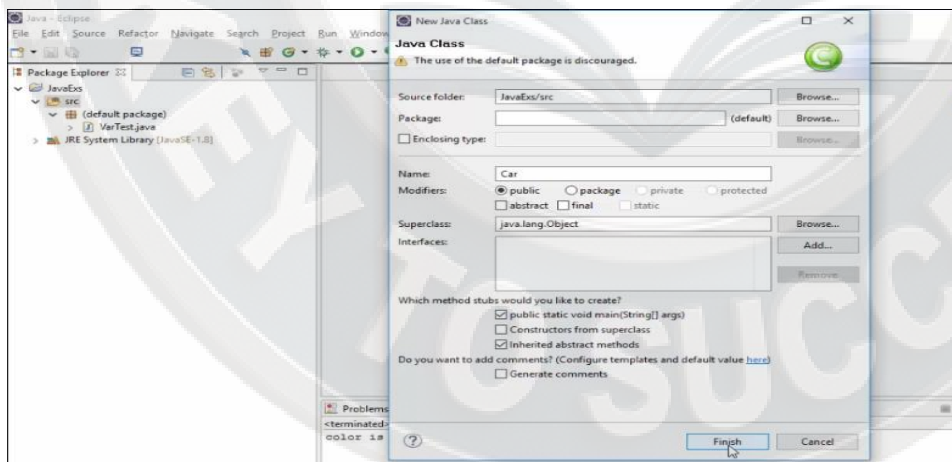


अब हम variable के type के बारे में जानेगे

1. Instance variables/object based :- यह basic variables है जिसे class में declare किया जाता है इसमें ध्यान रखा जाता है कि यह किसी method या constructor के अंदर ना आए इसका मतलब वह variables जो class में तो है किसी method या constructor में नहीं है इस तरह के variables object के द्वारा data store करने के लिए use किये जाते है और जब data चाहिए हो उसे निकाल सकते है हम इसका example image में देख सकते है कि जो हमारे object है वह अपना data अलग अलग जगह store करते है हम इसे practical करके देखते है



हमने eclipse को open किया है और यहा से हमने source पर right क्लिक करके new में class पर क्लिक किया है हमे new class बनाना है



इसमें हमने car name रखा और setting करके finish पर क्लिक किया

```

1 public class Car {
2
3
4 public static void main(String[] args) {
5     // TODO Auto-generated method stub
6
7
8
9 }
10

```

Problems | Javadoc | Declaration | Console

<terminated> Car [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 18, 2017, 1:15:43 AM)
color is red model is 2007

हमारे पास class car available हो गई है जिसमें void main एक method है इसके बाद ब्रैकेट का use किया है और इसके बीच में जितनी instruction दी है उन सभी को follow करता है हमारे पास अभी comment है हम इसे हटा देते हैं इसमें अगर हम किसी एक ब्रैकेट पर क्लिक करते हैं तो उसके साथ वाली ब्रैकेट भी select हो जाती है जिससे हम आसानी से ब्रैकेट match कर सकते हैं

```

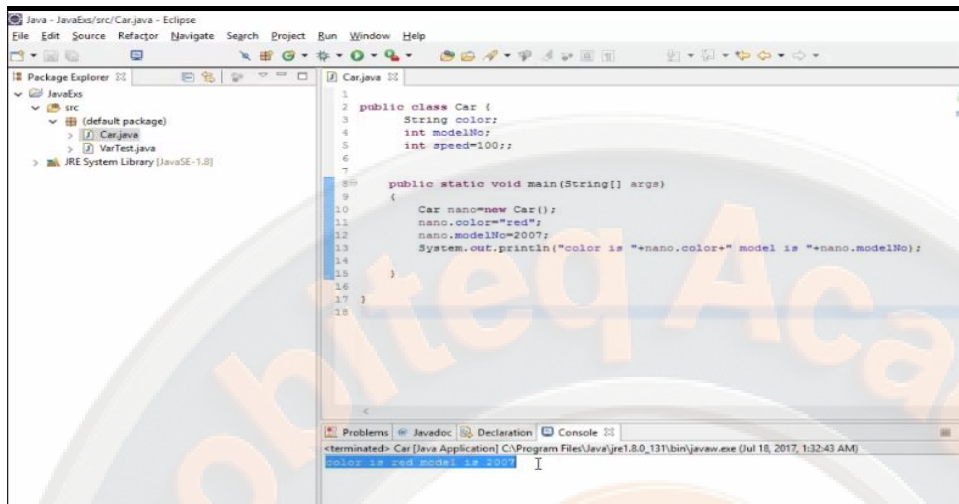
1 public class Car {
2     String color;
3     int modelNo;
4     int speed=100;
5
6
7
8
9 public static void main(String[] args)
10 {
11     Car nano=new Car();
12     nano.color="red";
13     nano.modelNo=2007;
14     System.out.println("color is "+nano.color+" model is "+nano.modelNo);
15
16 }
17
18

```

Problems | Javadoc | Declaration | Console

<terminated> Car [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 18, 2017, 1:15:43 AM)
color is red model is 2007

हमने आपनी coding को type कर लिया है हमने यहा पर class car बनाई है जिसमें तीन variable color, model no और speed हैं उसके बाद public static void main है इसके बाद opening curly ब्रैकेट है और फिर हमारे पास object nano है nano में color red और model no 2007 डिफाइन किया है इसके बाद हमने दोनों को system.out println के साथ print करवाया है इसमें हमने कुछ symbol का use किया है इस तरह हमने एक program बना लिया है हमने इसे run किया और ok किया है



```

1 public class Car {
2     String color;
3     int modelNo;
4     int speed=100;;
5
6
7
8     public static void main(String[] args)
9     {
10        Car nano=new Car();
11        nano.color="red";
12        nano.modelNo=2007;
13        System.out.println("color is "+nano.color+" model is "+nano.modelNo);
14    }
15 }
16
17
18

```

Problems | Javadoc | Declaration | Console

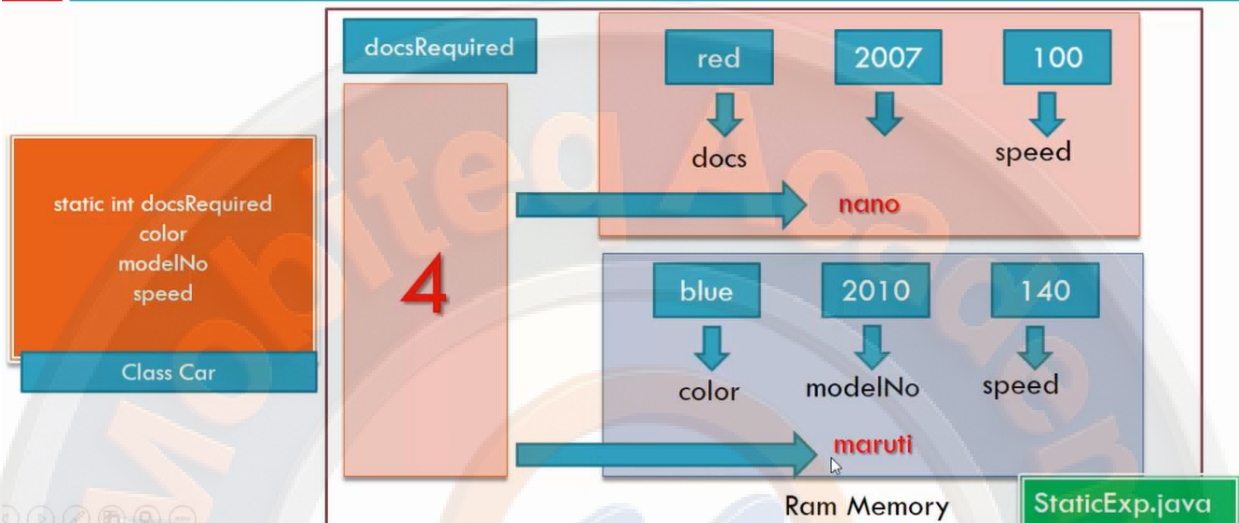
```

<terminated> Car [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 18, 2017, 1:32:43 AM)
color is red model is 2007

```

हमें नीचे output available हो गया है यह पूरा example instance variables से related है इसमें तीनो variables किसी class के अंदर तो है लेकिन इन्हें किसी method में declare नहीं किया गया है जैसे कि public static method है इसमें हमने variables को declare नहीं किया है इन variables को हम पूरी class में access कर सकते हैं यदि हमें variables को किसी static method में use करना है तो हमें उसके लिए उसके object का use करना होगा जैसे की हमने nano का use करके simple variables name use किए हैं अगर हम simple variable name का use सीधा करते हैं तो error available हो जाएगा हम किसी non-static method में इन simple variables का सीधा use कर सकते हैं

1. Static Variables/class based : static variable is class variable and can be shared by all objects of its class. That means all objects of that class can share that variable and static variable mostly used for shared data.



2. Static variables/class based :- यह class base variables होते हैं जब हमारे काफी सारे objects को shared data अर्थात जो सबके लिए एक जैसा हो तब हम static variables का use करते हैं ज्यादातर static variables का use final keyword के साथ किया जाता है और mostly इसे एक fixed value के साथ use किया जाता है इस तरह static class based होते हैं और सभी object को उसका same value ही available हो पाता है और यह data को share करने के लिए use किए जाते हैं इसमें नीचे एक example है इसमें class car है इसमें हमने static int docsRequired एक variable बनाया है इसमें हमने 4 value store की है अर्थात इसमें हमने 4 numerically store कर दिया है अब अगर nano इस particular variable का use करता है तो इसकी value 4 आती है और इसी तरह maruti इसे use करता है तो उसके पास भी 4 value होगी इस तरह docsRequired दोनों के लिए shared variable है अर्थात एक ही तरह का data show किया जा रहा है लेकिन अगर हम पहले वाले variables की बात करें तो वह seprate हैं जैसे nano के पास अपना color model no और speed है और इसी तरह maruti के पास भी है हम इसे practical देखते हैं

```

1 public class StaticExp {
2     static int docs=4;
3
4     public static void main(String[] args) {
5
6
7
8         System.out.println("Documents are required = " + docs);
9
10    }
11 }
12
13 }
14

```

Problems Javadoc Declaration Console

<terminated> StaticExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 18, 2017, 1:53:56 AM)
docs are 4
Documents are required = 4

हमने यहा पर आपनी class को डिफाइन कर लिया है हमने static class डिफाइन की है फिर हमने static int docs एक variable declare करवाया है docs variables का नाम है फिर value 4 की है फिर हमने void main method में इसका directly use कर सकते है बिना किसी object को बनाए हुए हमने यहा पर किसी object को call नही किया है इस तरह यह class base variables है

```

1 public class StaticExp {
2     static int docs=4;
3
4     public static void main(String[] args) {
5
6
7
8         System.out.println("Documents are required = " + docs);
9
10    }
11 }
12
13 }
14

```

Problems Javadoc Declaration Console

<terminated> StaticExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 18, 2017, 2:02:43 AM)
Documents are required = 4

हमने आपने program को run किया और हमारे पास 4 output available है इसको static method या non static method में दोनों में बिना किसी object के directly call कर सकते है

Variable is a name of memory location of storage.
Which is used to store data for program.

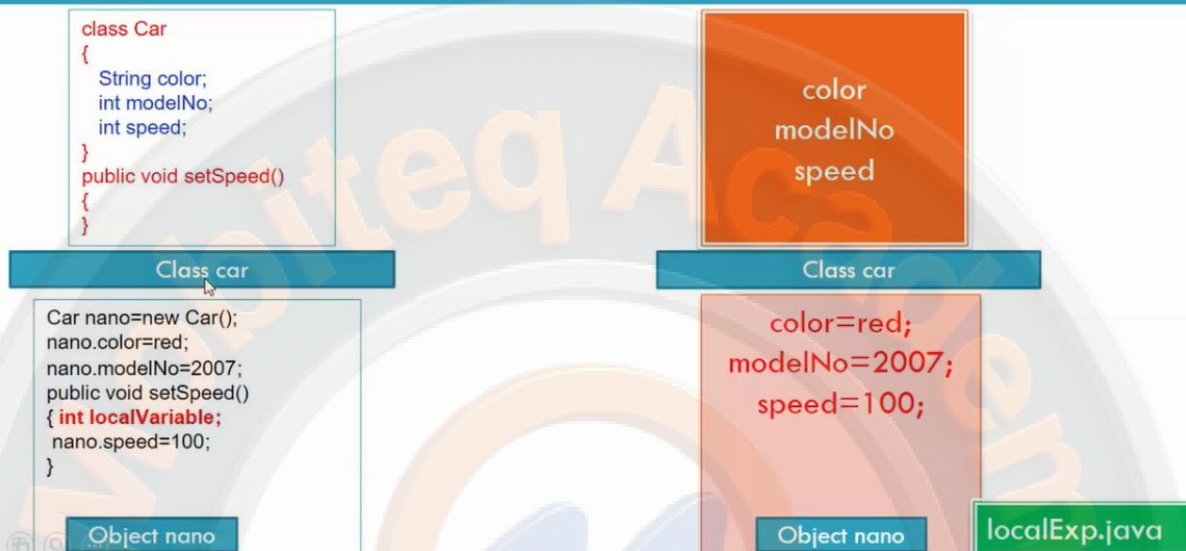
1. **Instance variables/object based** : declare in class but outside any method/ constructor. These variables are used by objects to store their data. Here every object can store their data in its variables and can get any time when it needs.
2. **Static Variables/class based** : static variable is class variable and can be shared by all objects of its class. That means all objects of that class can share that variable and static variable mostly used for shared data.
3. **Local Variables/ within any method: variables which are declared under a block or method and destroyed once that block or method ends. It includes basic method and special method constructor also or any block which we create using { } for variables.**
 1. **Parameters/within any method**

localExp.java

3. **Local variables/within any method** -: इसका मतलब जब किसी variable को किसी particular method में declare करवाए जाए यदि हम ऐसा करते हैं तो वह केवल तभी तक available रहता है जब तक वह method रहता है जैसे ही method खत्म हो जाता है तो वह local variable भी आपनी existence से हट जाता है अर्थात वह भी memory में available नहीं रहता है यह variable मुख्यतः हमारे पास method में local workout के लिए declare किए जाते हैं इसको हम example से समझते

७५

Variable is a name of memory location of storage.
Which is used to store data for program.



यहाँ हमारे पास class car है नीचे object nano में setSpeed method दिया गया है इसके अंतर्गत एक variable int local variable declare किया गया है यह set speed के अंतर्गत declare किया गया है अर्थात् यह ब्रैकेट्स के अंदर तक ही available रहता है जैसे ही हम block से बाहर जाएंगे इसकी availability खत्म हो जाएगी और इसका use हम नहीं कर पाएंगे अब हम इसे practically देखते हैं

```

1 public class LocalExp {
2
3     int insVar=10;
4
5     public void first()
6     {
7         int localVar1=20;
8         System.out.println("Local variable in first method = "+localVar1);
9         System.out.println("Instance variable in first method = "+insVar);
10    }
11
12
13
14    public void second()
15    {
16        int localVar2=30;
17        System.out.println("Local variable in second method = "+localVar2);
18        System.out.println("Instance variable in first method = "+insVar);
19    }
20
21    public static void main(String[] args) {
22
23        LocalExp v=new LocalExp();
24        System.out.println("Instance variable in first method = "+v.insVar);
25    }
  
```

यहाँ पर हमारे class का नाम LocalExp है और इसके अंतर्गत एक variable जो कि instance variables है उसे insVar के साथ declare किया है यह एक instance base या object base variable है इसके बाद हमारे पास दो method first और second available हैं first variable में एक local variable1

declare किया गया है जिसमें 20 assign किया गया है और second method में एक local variable2 declare किया गया है इस तरह हमारे पास तीन variables है जैसे कि insvar है जोकि object base variable है जोकि पूरी class में available है क्योंकि इसे हमने सबसे पहले declare किया है उसके बाद हमने दो variables first और second method में declare किए हैं local var1 first method में चलेगा और local var2 second method में चलेगा इसके बाद हम void main में आते हैं जिसमें हमने system.out.println के साथ एक object class v के नाम से बना लिया है और इसके द्वारा हम insvar को call करते हैं अगर हम इसे run करें तो हमारे पास 10 output available हो जाता है क्योंकि यह instance base variable था तो इसे directly call किया है इसे v object के साथ instance करके call किया लेकिन अगर हम यहाँ पर localvar1 को call करते तो हमारे पास error available हो जाता है अर्थात् यह exist नहीं कर रहा है क्योंकि यह केवल उन ब्रैकेट्स के बीच ही available रहेगा जहाँ पर इसे declare किया गया है जैसे ही हम ब्रैकेट से बाहर आकर इसे static void main में use करते हैं तो यह वहाँ पर available नहीं हो पाएगा इसी कारण से इन्हें local variables कहते हैं क्योंकि इनकी execution का scope अपने method या block तक ही रहता है यहाँ पर जो insvar रहते हैं उन्हें पूरी class में कहीं भी use कर सकते हैं इस तरह से हम इन variables का use कर सकते हैं

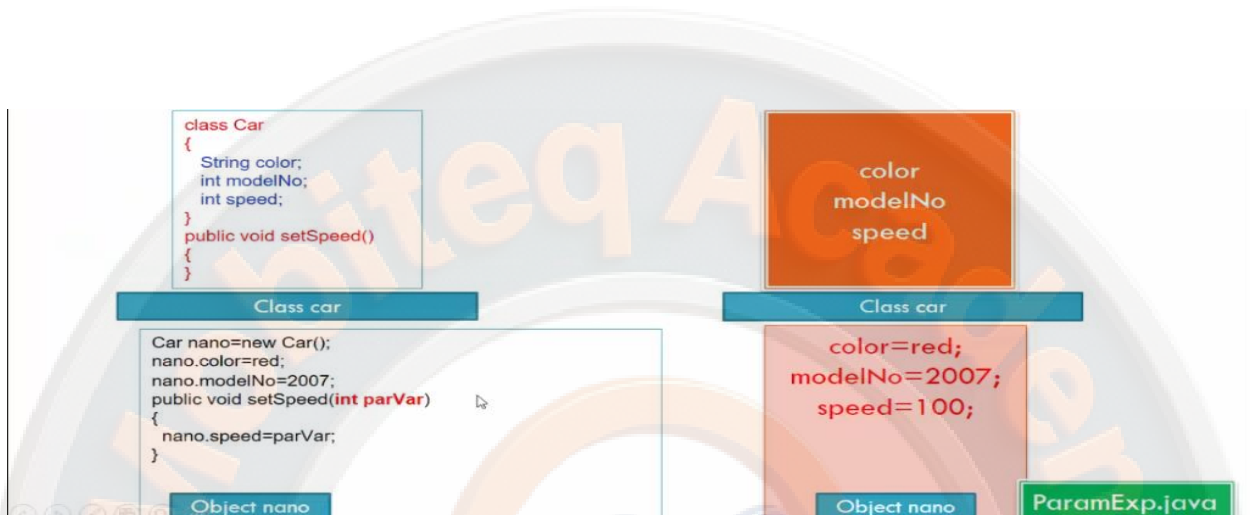
Variable is a name of memory location of storage.
Which is used to store data for program.

1. **Instance variables/object based** : declare in class but outside any method/ constructor. These variables are used by objects to store their data. Here every object can store their data in its variables and can get any time when it needs.
2. **Static Variables/class based** : static variable is class variable and can be shared by all objects of its class. That means all objects of that class can share that variable and static variable mostly used for shared data.
3. **Local Variables/ within any method**: variables which are declared under a block or method and destroyed once that block or method ends. It includes basic method and special method constructor also or any block which we create using { } for variables.
 1. **Parameters/within any method** : used in any method as an argument and behave exactly like local variables

ParamExp.java

3. 1. Parameters/within any method -: यह कोई अलग से variable नहीं है यह local variable ही है जो किसी particular method में use किए जाते हैं परन्तु इन्हें argument की तरह use किया

जाता है इसका मतलब जब हम किसी method को अलग से कोई extra information provide करे तो वह information argument कहलाती है



यहा हम example में देखते है हमारे पास public void setspeed के नाम से एक function है इसमें int parVar दिया गया है जोकि integer data type का है और यह parameter variable है इसमें जो value आएगी वह value nano. speed में save कर दी जाएगी यह बिल्कुल local variable की तरह ही है अर्थात जब भी इसे किसे method में declare करवाया जाता है उसी के अंदर ही यह available रहता है इसके बाहर इसे access नहीं कर सकते है

```

1 public class ParamExp {
2     int total;
3     public void sum(int x,int y)
4     {
5         total=x+y;
6         System.out.println("total is= "+total);
7     }
8     public static void main(String[] args) {
9         ParamExp p=new ParamExp();
10    }
11 }
12
13
14
15
16
17
18
19
20
21 }
22

```

यहा हमारे पास paramExp class है फिर एक variable int total लिया है उसके बाद हमारे पास एक method void sum है इसमें हमने दो parameter int x और int y पास किए है उसके बाद दोनों का use करते हुए total कर लेते है और जो total variable बनाया था उसमे store कर देते है और फिर उसे print किया है इसमें जो argument है वह parameter variable है यह केवल void sum method

में ही चलेगे इसके बाहर इसे access नहीं कर सकते हैं नीचे हमारे पास static void main method है और p object भी create किया है अगर हम

```

1 public class ParamExp {
2     int total;
3
4     public void sum(int x,int y)
5     {
6         total=x+y;
7         System.out.println("total is= "+total);
8     }
9
10
11
12
13
14 public static void main(String[] args) {
15     ParamExp p=new ParamExp();
16
17     p.sum(10,20);
18     System.out.println();
19 }
20
21
22
23

```

terminated- ParamExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 18, 2017, 2:42:15 AM)
total is= 30

इस method को p.sum करके call करते हैं और ब्रैकेट में value पास करते हैं और run करते हैं तो हमारे पास total 30 available हो जाता है अगर हम p.x =30 करते हैं तो error available हो जाता है क्योंकि sum method के अंतर्गत ही इसे call किया जा सकता है और यह local variable है इस तरह हम parameter variables का use कर सकते हैं और argument पास कर सकते हैं और method में variables का use कर सकते हैं

Naming rules for variables.

- ❑ Variable names are **Case-sensitive** in Java.
- ❑ Variable name **can have Letter, digits** and two special characters **underscore** and **'\$'** sign.
- ❑ Name should with **alphabet at the start** , cannot start with number. Can use underscore and **'\$'** sign.
- ❑ **Following characters** can be letters, digits, \$ or _ character.
- ❑ **White space** , special characters like **, * ;** are **not allowed**.
- ❑ Variable name must **not be a keyword** (reserved word).

अब हम जानेगे कि जब हम variables का नाम रखते है तो किन rules को follow करना चाहिए वह यहा पर show किए जा रहे है अगर हम इन्हें follow नही करेगे तो हमारे पास error available हो जाएगे इस तरह हमे इन्हें ध्यान में रखना चाहिए

- **datatype** **variablename;**



Lets now look at this part



We have learnt

variables को declare करने के लिए हमे दो चीजो की जरूरत होती है 1st datatype कि उस data का टाइप क्या है और 2nd variablename कि उस variable का name क्या है variables name वाला part हमने सीख लिया है अब हम data type की जानकारी लेगे |

- Data types represent the value which is to be stored in the variable memory space. And we need to tell the system in advance that which type of data type we are going to store in variable memory In java.

Java data types:

Primitive data types

Non-primitive data types

जब हम कोई data कंप्यूटर में store करते हैं तो यह बताने के लिए की data किस प्रकार का है यह हम datatype से बताते हैं इसका मतलब हम java में पहले से बता देते हैं की data किस प्रकार का है जैसे की data word है कि number है इनकी जानकारी देने के लिए java में मुख्यतः दो types available हैं जोकि primitive और non-primitive datatype हैं

Data types			
Primitive			Non-Primitive
boolean	Numeric		String To store text strings eg. "India"
boolean to store true/false	character To store characters eg. 'a'	char	Array To store same datatype values eg. int arr[5];
	integer To store numbers eg. 10	byte, short, int, long	Class To store userdefined datatype eg. Student s1=new Student();
	float To store decimals eg. 10.58	float, double	

1. Primitive datatype में दो section boolean और numeric हैं हम boolean में true और false data को save कर सकते हैं
2. यदि हम किसी प्रकार के number को store करना चाहते हैं तो numeric datatype का use कर सकते हैं numeric datatype में हम अगर word store करना चाहते हैं तो उसके लिए char datatype का use करेंगे

3. इसी तरह अगर किसी प्रकार के number को store करना चाहते हैं और number छोटा है तो उसके लिए byte और short का use कर सकते हैं और number बड़ा है तो int और long का use करेंगे
4. यदि हम decimal value store करना चाहते हैं तो float का use कर सकते हैं और value decimal में बड़ी है तो double का use करेंगे

2. Non-Primitive data type में string का use कर सकते हैं यदि हमें काफी सारे character इकट्ठे store करने हैं तो string का use कर सकते हैं

इसी तरह से array का use कर सकते हैं यदि हमें काफी सारे variables एक ही datatype के store करना चाहते हैं तो array का use कर सकते हैं इसी तरह class एक user defined datatype है

Variables and Data Types					
Primitive Data Types					
1. Integer			2. Floating Point		
Type	Size	Values Range	Type	Size	Values Range
byte	1 byte	-128 to 127	float	4 bytes	3.4e-038 to 3.4e+038
short	2 bytes	-32768 to 32767	double	8 bytes	1.7e-308 to 1.7e+308
int	4 bytes	-2,147,483,648 to 2,147,483,647			
long	8 bytes	9,223,372,036,854,775,808 to 9,223,372,036,854,750,000			
3. Character			4. Boolean		
Type	Size	Values Range	Type	Size	Range of values that can be stored
char	2 bytes	can store one character eg. "a"	boolean	1 bit	can store true/false

primitive datatype में चार categories का use किया है यहाँ पर data type के sizes की जानकारी दी है जिसमें की byte के बारे में बताया है और साथ ही value range की जानकारी दी गई है



Lets learn how to declare variable

```
int a, char b;
a=10;
b="G";
```

```
int a;
char b;
int a=10;
char b="G";
```

```
int a=10;
char b="G";
```

```
int a,b,c;
```

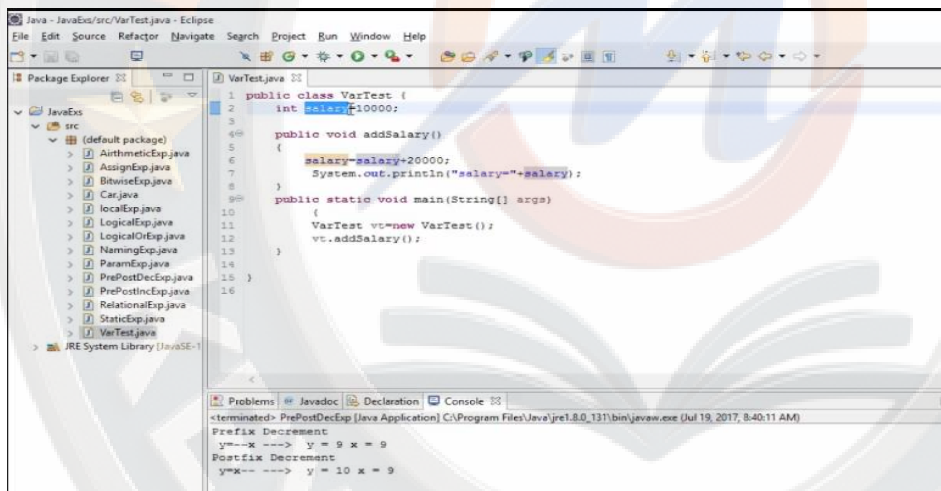
```
long l=150L;
float f=180.34f;
double d= 250.43d;
```

VarTest.java

By Using Literals values directly cast to respective datatype.
Otherwise it casts first as small (byte) and than large(long) datatype.

Default value of all primitive numeric value is always = 0 .

यहा पर variables declare करवाने के तरीके बताए गए है कि किस प्रकार से variables को declare कर सकते है इसमें से किसी भी एक तरीका का use कर सकते है



```
1 public class VarTest {
2     int salary=10000;
3
4     public void addSalary()
5     {
6         salary=salary+20000;
7         System.out.println("salary="+salary);
8     }
9
10    public static void main(String[] args)
11    {
12        VarTest vt=new VarTest();
13        vt.addSalary();
14    }
15 }
16
```

Console Output:

```
<terminated> PrePostDecExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 8:40:11 AM)
Prefix Decrement
y=-x ----> y = 9 x = 9
Postfix Decrement
y=x----> y = 10 x = 9
```

इसमें हमने int एक variable declare किया है जिसका नाम salary है और इसमें value 10000 store की है क्योंकि इसमें हम numeric data store कर सकते हैं हमने एक बार variable declare करने की जरूरत होती है हम इसे बार बार declare करने की जरूरत नहीं है इसका use हम कहीं भी कर सकते हैं इसमें हम numerically data store कर सकते हैं हम इसे run करे तो हमारे पास output available हो जाता है इस तरह से आसानी से variables को create कर सकते हैं

```

1 public class VarTest {
2     int salary=10000;
3     char a='a';
4
5     public void addSalary()
6     {
7         salary=salary+20000;
8         System.out.println("salary="+salary);
9         System.out.println("a="+a);
10    }
11
12    public static void main(String[] args)
13    {
14        VarTest vt=new VarTest();
15        vt.addSalary();
16    }
17 }
18

```

Console Output:

```

<terminated> VarTest [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 9:12:48 AM)
salary=30000
a=a

```

इसी तरह से हम char variables create करके print करवाया है और run किया तो हमारे पास output available है इस तरह हम variables को create कर सकते हैं हमें पहले से बताना पड़ता है कि हमें किस प्रकार का data store करना है इस तरह से हम इनका use कर सकते हैं

Operators to workout on Variables

- Arithmetic Operators
- Assignment Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Postfix and Prefix Operators
- Conditional Operator

अब हम operators के बारे में जानकारी लेंगे इससे हम आसानी से calculation कर सकते हैं information को relate कर सकते हैं इसके अतिरिक्त बहुत से कार्य कर सकते हैं

1. Arithmetic Operators

Arithmetic Operators			
Operator	Description	Example	Output X=10; b=5;
+ (Addition)	It Adds values .	X + Y	15
- (Subtraction)	It Subtracts values.	X - Y	5
* (Multiplication)	It Multiplies values.	X * Y	50
/ (Division)	It Divides values.	X / Y	2
% (Modulus)	It provides remainder in result.	X % Y	0

ArithmeticExp.java

यह हमारे arithmetic operators हैं जिनसे हम calculation कर सकते हैं इसे practically देखते हैं

```

1 public class AirthmeticExp {
2     int x=10,y=5,z;
3
4     public void calculate()
5     {
6         z=x+y;
7         System.out.println("addition="+z);
8
9         z=x-y;
10        System.out.println("subatration="+z);
11
12        z=x*y;
13        System.out.println("multiplication="+z);
14
15        z=x/y;
16        System.out.println("division="+z);
17
18        z=x%y;
19        System.out.println("modulus =" +z);
20    }
21
22    public static void main(String[] args)
23    {
24        AirthmeticExp obj=new AirthmeticExp();
25        obj.calculate();
26    }
27 }

```

Problems | Javadoc | Declaration | Console

<terminated> VarTest [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 9:12:58 AM)

salary=30000

हमने यहा पर तीन variables x, y, z declare किया है इसके बाद public void method बनाया है इसमें हमने सभी operators को शामिल किया है जो हमारे पास arithmetic examples में available है उसके बाद void main में जाकर arithmetic class का obj बनाया है और इससे method को call किया है

```

1
2 public class AirthmeticExp {
3
4     int x=10,y=3,z;
5
6     public void calculate()
7     {
8         z=x+y;
9         System.out.println("addition="+z);
10
11        z=x-y;
12        System.out.println("subtraction="+z);
13
14        z=x*y;
15        System.out.println("multiplication="+z);
16
17        z=x/y;
18        System.out.println("division="+z);
19
20        z=x%y;
21        System.out.println("modulus =" +z);
22
23    }
24 }

```

<terminated> AirthmeticExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 11:24:44 AM)
 addition=13
 subtraction=7
 multiplication=30
 division=3
 modulus =1

हमने इसे run किया और हमारे पास output available हो गए हैं इस तरह यह बिल्कुल simple operators हैं जिनका use आसानी से कर सकते हैं

2. Assignment Operators

operator	Description	Example X=10; Y=5	Operation	Output
=	Assignment operator is used to assigns values from right to left.	X=Y;	X=Y	X=5;
+=	Add & assignment operator is used to add first and then assign the result to left.	X+=Y;	X=X+Y	X=15;
-=	Subtract & assignment operator is used to subtract first and then assign the result to left.	X-=Y;	X=X-Y	X=5;
=	Multiply & assignment operator is used to multiply first and then assign the result to left.	X=Y;	X=X*Y	X=50;
/=	Divide & assignment operator is used to divide first and then assign the result to left.	X/=Y;	X=X/Y	X=2;
%=	Modulus & assignment operator is used to modulus first and then assign the result to left.	X%=Y;	X=X%Y	X=0;

यहा पर हमारे पास assignment operators हैं इससे हम आसानी से value को assign कर सकते हैं अब हम इसे practical करके देखेंगे

```

1 public class AssignExp {
2
3     public static void main(String args[]){
4         int x = 20;
5         int y = 30;
6         int z = 50;
7
8         z = x + y;
9         System.out.println("assignment = " + z);
10
11        [x=20;y=30;z=50;
12
13        z += x;
14        System.out.println("add & assignment = " + z);
15
16        x=20;y=30;z=50;
17        z -= x;
18        System.out.println("subtract & assignment = " + z);
19
20        x=20;y=30;z=50;
21        z *= x;
22        System.out.println("multiply & assignment = " + z);
23
24        x=20;y=30;z=50;
25
26        z /= x;
27        System.out.println("divide & assignment = " + z);
28
29        x=20;y=30;z=50;
30        z %= x;
31        System.out.println(" modulus & assignment = " + z);
32
33    }
34 }

```

Console Output:

```

<terminated> AssignExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 11:34:05 AM)
assignment = 30
add & assignment = 30
subtract & assignment = 30
multiply & assignment = 1000
divide & assignment = 2
modulus & assignment = 10

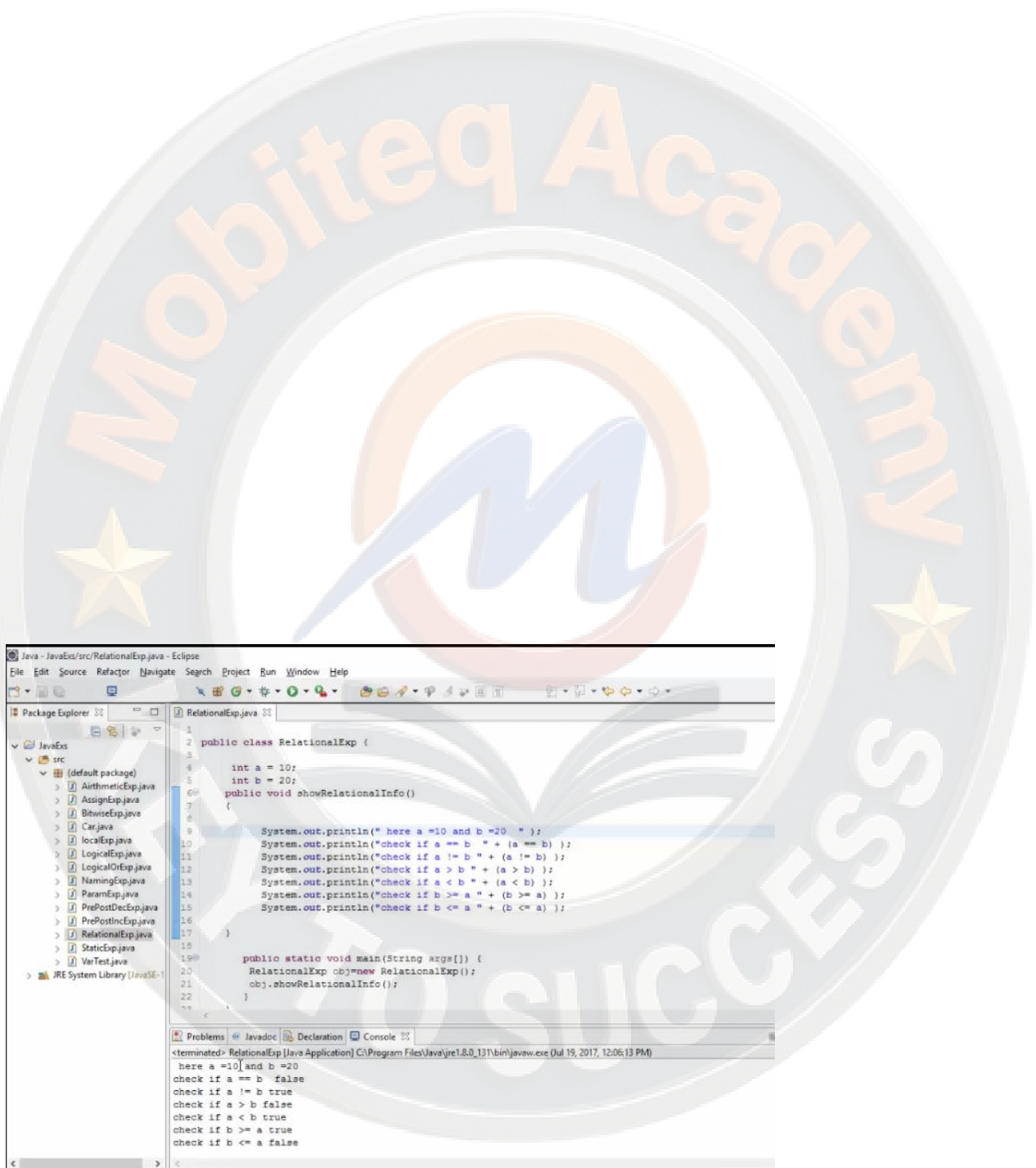
```

यहा पर हमने तीन variables x, y, z declare किया है और assignment examples को शामिल किया है और run किया तो हमारे पास output available है इस तरह आसानी से assignment operators का use कर सकते है

3. Relational Operators

□ Relational Operators		
operator	Description	Example
== (equal to)	It Checks the values are equal or not and provides true in case of equal and false in case of not equal values.	(X == Y)
!= (not equal to)	It Checks the values are equal or not and provides true in case of not equal and false in case of equal values.	(X != Y)
> (greater than)	It Checks the values are greater than or not and provides true in case of greater and false in case of not greater values.	(X > Y)
< (less than)	It Checks the values are less than or not and provides true in case of less than values and false in case of greater than values.	(X < Y)
>= (greater than or equal to)	It Checks the values are greater and equal than or not and provides true in case of greater and equal and false in case of not greater and equal values.	(X >= Y)
<= (less than or equal to)	It Checks the values are less than and equal than or not and provides true in case of less than and equal and false in case of not less than and equal values.	(X <= Y)

यहा पर हमारे पास relational operators available है अगर variables के बीच में condition को check करना चाहते है जैसे कि क्या एक variable दुसरे variable से बड़ा है या बराबर है या छोटा है इस प्रकार की condition के लिए relational operators से check कर सकते है इसमें यदि condition true है तो output true available होगा और अगर condition false है तो output false available होगा इसे हम practical करके देखते है



यहा पर हमने दो variables को a और b को declare किया है इसके बाद हमने सभी relational operators का use किया है इसे run करे तो output भी available हो गया है हमारे पास output true और false में available हुआ है इस तरह से हम इनका use कर सकते है

4. Logical Operators

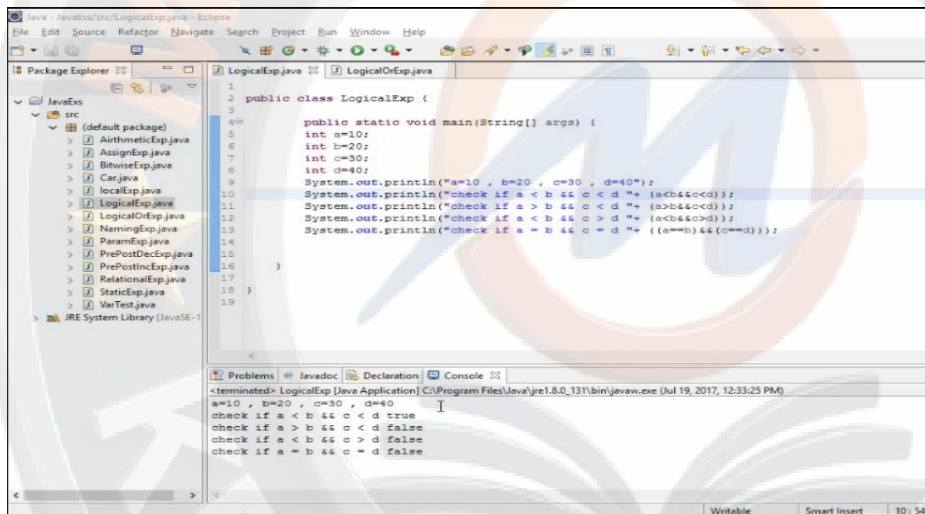
□ Logical Operators

operator	Description	Example
&& (logical and)	It gives true result if all conditions are true in output.	(X&& Y)
(logical or)	It gives true result if any one condition is true in output.	(X Y)
! (logical not)	Provides complement output.	!(X)

इसमें हम एक से ज्यादा condition को check कर सकते हैं यहा पर logical operators available हैं इसमें 1 and logical है इसमें अगर सभी condition true है तो output भी true होगा

2 or logical में यदि कोई एक condition true रहती है तो output true रहता है

3 not logical हमें complement output देता है इसका मतलब reverse output देता है इसमें हम not का sign use करते हैं



```

1 public class LogicalExp {
2
3
4
5     public static void main(String[] args) {
6         int a=10;
7         int b=20;
8         int c=30;
9         int d=40;
10        System.out.println("a=10 , b=20 , c=30 , d=40");
11        System.out.println("check if a < b && c < d "+ (a<b&&c<d));
12        System.out.println("check if a > b && c < d "+ (a>b&&c<d));
13        System.out.println("check if a < b && c > d "+ (a<b&&c>d));
14        System.out.println("check if a = b && c = d "+ ((a==b)&&(c==d)));
15    }
16
17
18 }
19

```

<terminated> LogicalExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 12:33:25 PM)
 a=10 , b=20 , c=30 , d=40
 check if a < b && c < d true
 check if a > b && c < d false
 check if a < b && c > d false
 check if a = b && c = d false

यहां पर हमने and logical operator को practical किया है और run किया तो हमारे पास output available है

```

1
2 public class LogicalOrExp {
3
4     public static void main(String[] args) {
5         int a=10;
6         int b=20;
7         int c=30;
8         int d=40;
9         System.out.println("a=10 , b=20 , c=30 , d=40");
10        System.out.println("check if a < b || c < d "+ (a<b||c<d));
11        System.out.println("check if a > b || c < d "+ (a>b||c<d));
12        System.out.println("check if a < b || c > d "+ (a<b||c>d));
13        System.out.println("check if a = b || c = d "+ ((a==b)|| (c==d)));
14    }
15 }
16
17
18
19

```

```

<terminated> LogicalOrExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 12:37:52 PM)
a=10 , b=20 , c=30 , d=40
check if a < b || c < d true
check if a > b || c < d true
check if a < b || c > d true
check if a = b || c = d false

```

इसी तरह हमने or logical operator को practical किया है और run करने के बाद output available है इस तरह से हम इनका use कर सकते हैं

5. Bitwise Operators

□ Bitwise Operators

operator	Description	Example
& (bitwise and)	It copies a bit to the result if all operands have 1 in value.	(X & Y)
(bitwise or)	It copies a bit to the result if any operand have 1 in value.	(X Y)
^ (bitwise XOR)	It copies a bit to the result if any one operand have 1 in value, but not in both.	(X ^ Y)
~ (bitwise compliment)	It flips the 1 with 0 and 0 with 1 in result.	(X ~ Y)
<< (left shift)	It moves left operands value by left side with the number of bits specified at right operand.	(X << Y)
>> (right shift)	It moves left operands value by right side with the number of bits specified at right operand.	(X >> Y)

bitwise operators मुख्यतः bitwise level पर काम करते हैं अर्थात binary digits पर काम करती हैं

x = 50 , y = 25

x = 00110010	x = 00110010	x = 00110010	x = 00110010
y = 00011001	y = 00011001	y = 00011001	
= 00010000	= 00111011	= 00101011	= 11001101
x & y = 16 (bitwise and)	x y = 59 (bitwise or)	x ^ y = 43 (bitwise XOR)	~ x = -51 (bitwise compliment)

x = 00110010	x = 00110010
= 11001000	= 00001100
X << 2 = 200 (left shift)	X >> 2 = 12 (right shift)

BitwiseExp.java

Bitwise Operators

यहा पर example में bitwise operators को देख सकते हैं जोकि binary digits में है यहा पर हमारे पास x 50 का binary code और y 25 का binary code available है आप and bitwise में x और y को देखे इसमें अगर दोनों में 1 है तो output 1 है और इसी तरह और bitwise में अगर किसी में भी एक है तो output 1 है अगर xor bitwise में देखे तो अगर किसी एक में भी 1 है तो output 1 है और अगर दोनों में 1 है तो output 0 है अगर दोनों में 0 है तो output 0 है इसी तरह complement bitwise में बिलकुल reverse value आती है इसमें जहा 0 है वहा पर 1 आ गया है और जहा पर 1 है वहा पर 0 आता है इसी तरह left shift bitwise में x की value को left side कितने digits से shift करेगा यह हम left shift operator के बाद देगे हमने 2 किया तो इसका मतलब left side 2 digits shift करेगे इसी तरह right shift भी available है

```

1 public class BitwiseExp {
2
3     public static void main(String[] args)
4     {
5         int x=50;
6         int y=25;
7
8
9         System.out.println("x & y = " + (x & y) );
10        // & (bitwise and) copies a bit to the result if all operands have 1 in value.
11
12        System.out.println("x | y = " + (x | y) );
13        // | (bitwise or) copies a bit to the result if any operand have 1 in value.
14
15        System.out.println("x ^ y = " + (x ^ y) );
16        // ^ (bitwise XOR) copies a bit to the result if any one operand have 1 in value, but not in both.
17
18        System.out.println("~x = " + ~x );
19        // ~ (bitwise compliment) flips the 1 with 0 and 0 with 1 in result.
20
21
22
23        System.out.println("x << 2 = " + (x << 2) );
24        // << (left shift) moves left operands value by left side with the number of bits specified at right
25
26
27        System.out.println("x >> 2 = " + (x >> 2) );
28        // >> (right shift) moves left operands value by right side with the number of bits specified at right
29
30    }
31
32 }

```

```

<terminated> BitwiseExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 3:04:42 PM)
x & y = 16
x | y = 59
x ^ y = 43
~x = -51
x << 2 = 200
x >> 2 = 12

```

हमने आपने program को type करके run किया तो हमारे पास output available है इस तरह से हम bitwise operator का use कर सकते है

6. Prefix and Postfix operators

	prefix	x=10, y=0	postfix	x=10,y=0
Increment Adds 1 in x	++x	y=++x y=11, x=11	x++	y=x++ Y=10, x=11
Decrement Subtract 1 from x	--x	y>--x y=9, x=9	x--	y=x-- y=10, x=9

अब हम prefix और postfix operators के बारे में जानेगे यहा पर हमारे पास दोनों के example available है इसमें हमारे पास left side पर increment operator दिया गया है prefix में जो variables होते हैं उनके आगे दो plus के निशान use किए जाते हैं और postfix operator में variables के बाद plus के निशान लगाए जाते हैं दोनों cases में इसका अर्थ यह होता है कि x की value में 1 जोड़ा जाए जैसे कि अगर x= 10 है तो दोनों case में 1 जोड़कर 11 हो जाएगा लेकिन इसमें फर्क है दोनों में y =x++ है दोनों cases में x=11 है इसमें y की value अलग है जब हम prefix use करते हैं तो x आपनी value में 1 add करता है यानि 10 है तो 11 हो जाएगा इसमें x जो 11 आया है और उसे y को assign किया जाएगा जबकि postfix में पहले x आपनी value को assign करता है यानि की y हमारे पास 10 आएगी और उसके बाद x आपनी value में 1 add करता है इस तरह difference assignment में आया है इसी तरह हमारे पास prefix और postfix दोनों में decrement operator भी है इसमें जो value होती है उसमे में से 1 less कर देता है इसमें minus का use किया गया है बाकी same है इसमें भी assignment में फर्क है इस तरह से हम इन दोनों का use कर सकते हैं

The screenshot shows the Eclipse IDE with a Java project named 'JavaEx'. The Package Explorer on the left shows a package named 'src' containing various Java files. The main editor displays the code for 'PrePostIncExp.java'. The code defines a public class 'PrePostIncExp' with a 'main' method. It declares two integer variables, 'x' and 'y', both initialized to 10. The 'main' method contains two blocks of code. The first block demonstrates the prefix increment operator: it prints 'Prefix Increment', increments 'y' to 11, and prints 'y = 11 x = 11'. The second block demonstrates the postfix increment operator: it prints 'Postfix Increment', increments 'x' to 11, and prints 'y = 10 x = 11'. The Console window at the bottom shows the output of the program, which matches the printed statements in the code.

```

1 public class PrePostIncExp {
2
3
4     public static void main(String[] args) {
5
6         int x,y;
7         x=10;
8         y=0;
9
10        System.out.println("Prefix Increment ");
11        y++;
12        System.out.println(" y==x ----> y = "+y+" x = " + x);
13
14
15
16        x=10;
17        y=0;
18
19        System.out.println("Postfix Increment ");
20        y=x++;
21        System.out.println(" y=x+ ----> y = "+y+" x = " + x);
22
23    }
24 }

```

```

<terminated> PrePostIncExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 3:24:45 PM)
Prefix Increment
y==x ----> y = 11 x = 11
Postfix Increment
y=x+ ----> y = 10 x = 11

```

यहा पर हमने increment operator का program prefix और postfix दोनों में type करके run किया है और हमारे पास output available है

The screenshot shows the Eclipse IDE with the same Java project. The Package Explorer on the left shows the package 'src' containing various Java files. The main editor displays the code for 'PrePostDecExp.java'. The code defines a public class 'PrePostDecExp' with a 'main' method. It declares two integer variables, 'x' and 'y', both initialized to 10. The 'main' method contains two blocks of code. The first block demonstrates the prefix decrement operator: it prints 'Prefix Decrement', decrements 'y' to 9, and prints 'y = 9 x = 9'. The second block demonstrates the postfix decrement operator: it prints 'Postfix Decrement', decrements 'x' to 9, and prints 'y = 10 x = 9'. The Console window at the bottom shows the output of the program, which matches the printed statements in the code.

```

1 public class PrePostDecExp {
2
3
4     public static void main(String[] args) {
5
6         int x,y;
7         x=10;
8         y=0;
9
10        System.out.println("Prefix Decrement ");
11        y--;
12        System.out.println(" y==x ----> y = "+y+" x = " + x);
13
14
15
16        x=10;
17        y=0;
18
19        System.out.println("Postfix Decrement ");
20        y=x--;
21        System.out.println(" y=x-- ----> y = "+y+" x = " + x);
22
23    }
24 }

```

```

<terminated> PrePostDecExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 3:26:44 PM)
Prefix Decrement
y==x ----> y = 9 x = 9
Postfix Decrement
y=x-- ----> y = 10 x = 9

```

यहा पर decrement operator का example लिया और उसे run किया तो हमारे पास output available है इस तरह से postfix और prefix operators का use आसानी से कर सकते है हमे use करते समय इसके बीच जो फर्क है उसे ध्यान में रखना ताकि हमे output सही available हो

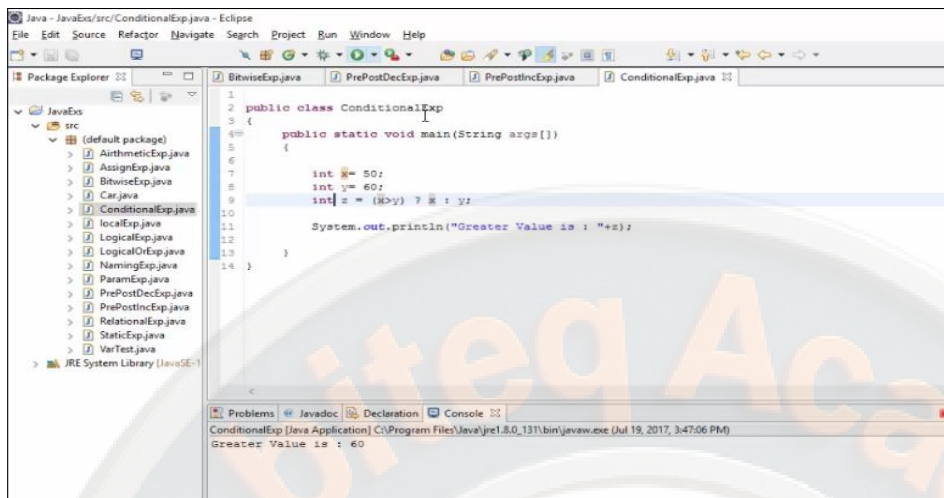
7. Conditional Operators

- When we need to define conditional expressions we can use conditional operators to define them

(Condition) ? value if true : value if false ;

$(x < y) ? x : y;$

अब हम conditional operators के बारे में जानेगे जब भी हमे conditional expressions बनाने होते है तब conditional operators का use कर सकते है इसमें सबसे पहले condition आती है अगर condition true है question mark के साथ पहली value select की जाएगी अगर condition false हो तो double column के पश्चात् आने वाली value को select किया जाएगा जैसे की example के लिए एक code available है यदि x less then हो y से तो x की value select की जाए परन्तु यदि y छोटा है x से तो y की value select की जाए इसे हम practical करके देखते है

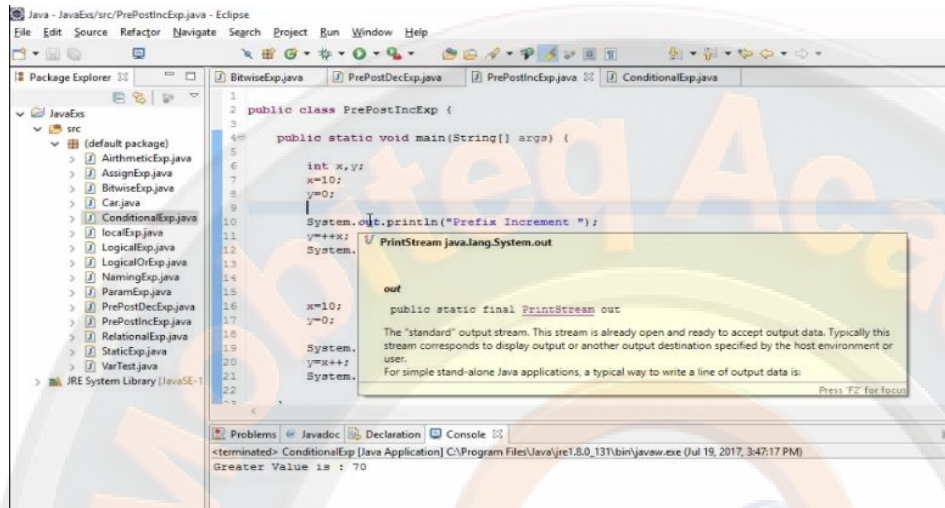


```
1 public class ConditionalExp
2 {
3     public static void main(String args[])
4     {
5         int x = 50;
6         int y = 60;
7         int z = (x > y) ? x : y;
8
9         System.out.println("Greater Value is : "+z);
10    }
11 }
12
13
14 }
```

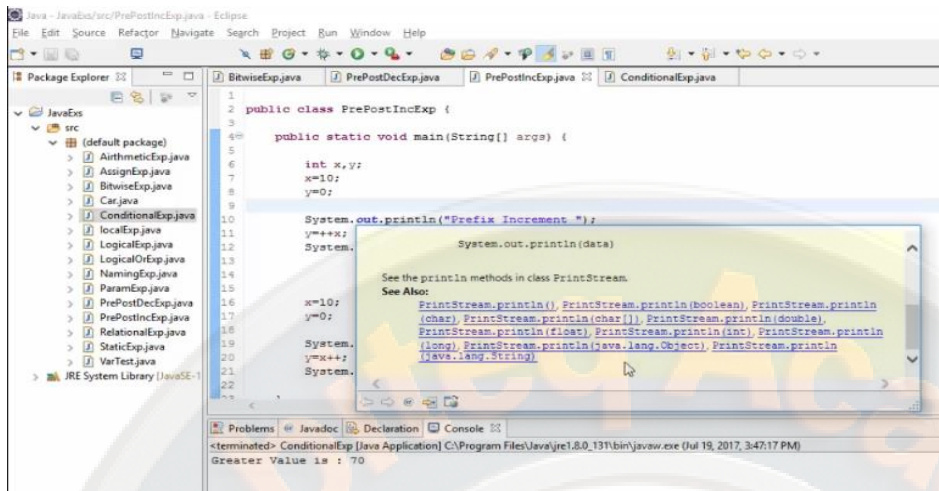
ConditionalExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\java.exe (Jul 19, 2017, 3:47:06 PM)
Greater Value is : 60

यहा पर हमने एक class create की है उसमे तीन variables declare करवाए है और z में condition operator का use किया है और run किया तो हमारे पास 60 output available है इस तरह जो बड़ा number है वह z में store होगा इस तरह से हम conditional operators का use कर सकते है और बहुत ही आसानी से use कर सकते है

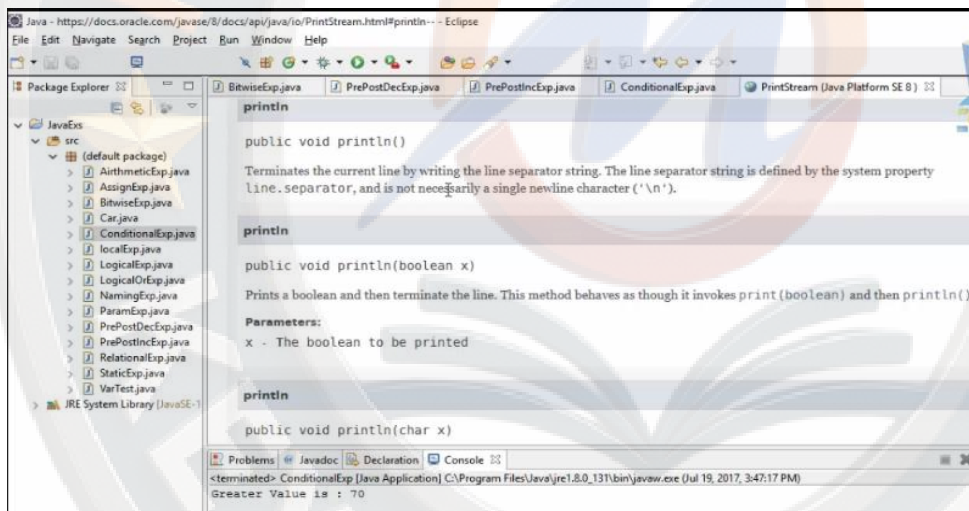
Help In Java



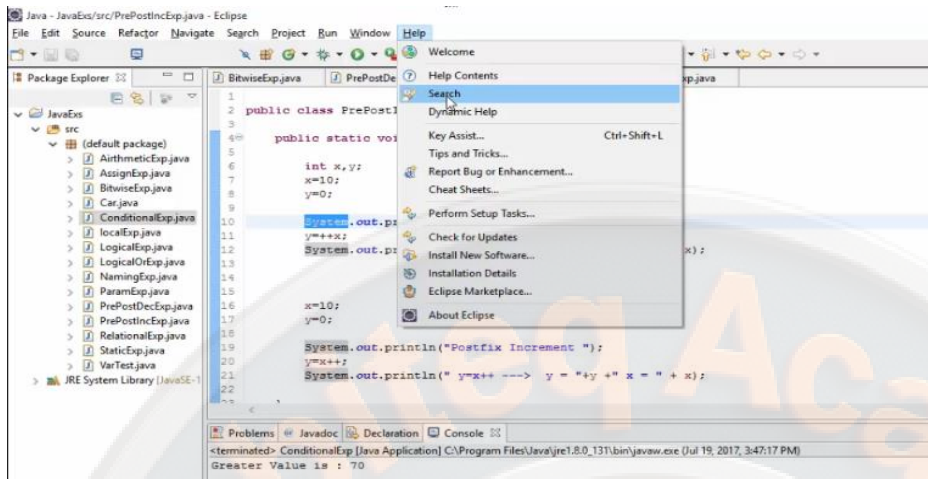
अब हम जानेगे कि eclipse की सहायता से या google से online help का use करके आपने program को आसानी से बना सकते है इसके लिए हमे java help के बारे में पता होना चाहिए हमने यहा पर एक program type किया हुआ है मान लीजिए हमे किसी topic में help चाहिए तो हमे बस कर्सर को वहा पर लेकर जाना है तो हमारे पास एक popup box available हो जाता है तो उससे related सारी information available हो जाती है हम इस box पर आए तो scrollbars भी available हो जाएगे इसे scroll करके सारी information को देख सकते है



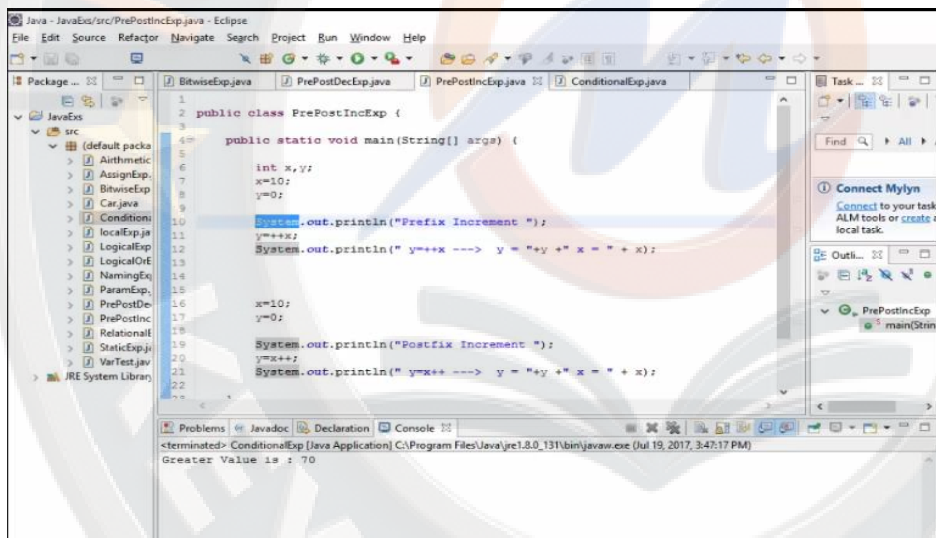
अगर हम नीचे आए तो हमारे पास links available है अगर हमें ओर जानकारी चाहिए तो इन पर क्लिक कर सकते है



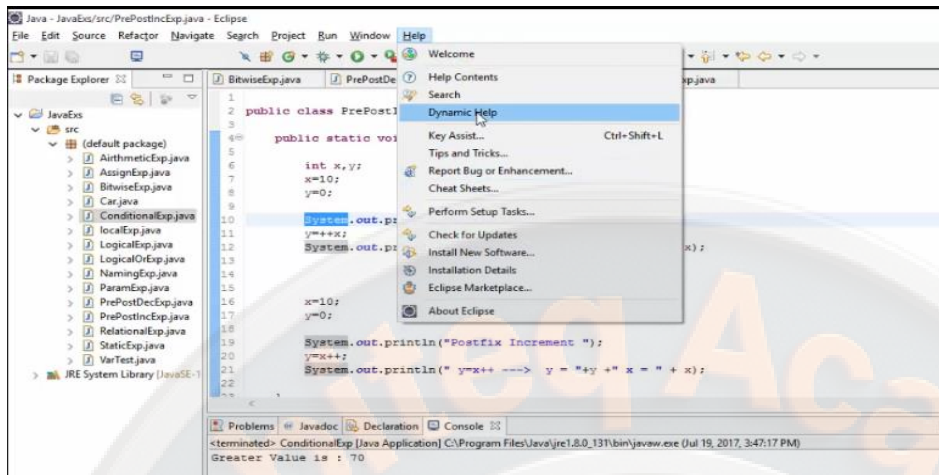
तो हमारे पास एक वेब browser open हो जाएगा जिसमे हमारे पास help available हो जाएगी इसमें basic से लेकर advance तक की सभी information ले सकते है



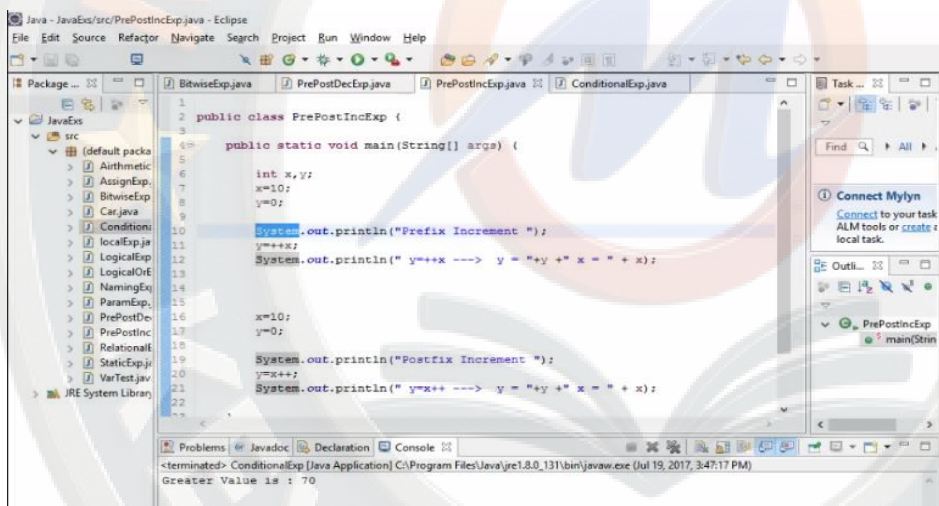
हम help के लिए उस word को select करके help menu में जाकर search पर क्लिक करके उसकी जानकारी ले सकते हैं



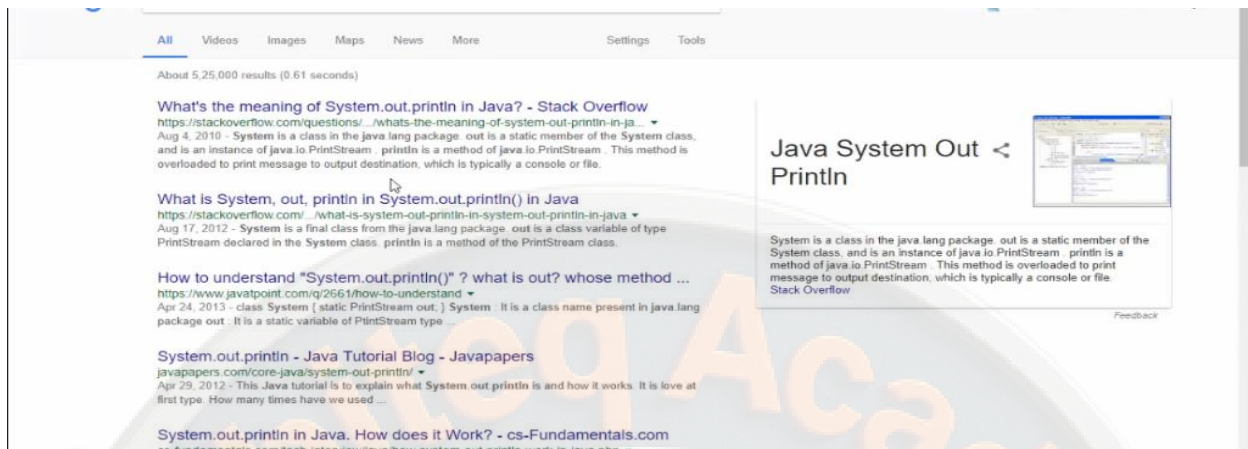
हमारे पास right side पर box available हो जाएगा हम उसमे type करके भी enter press कर सकते हैं और हमारे पास information available हो जाएगी



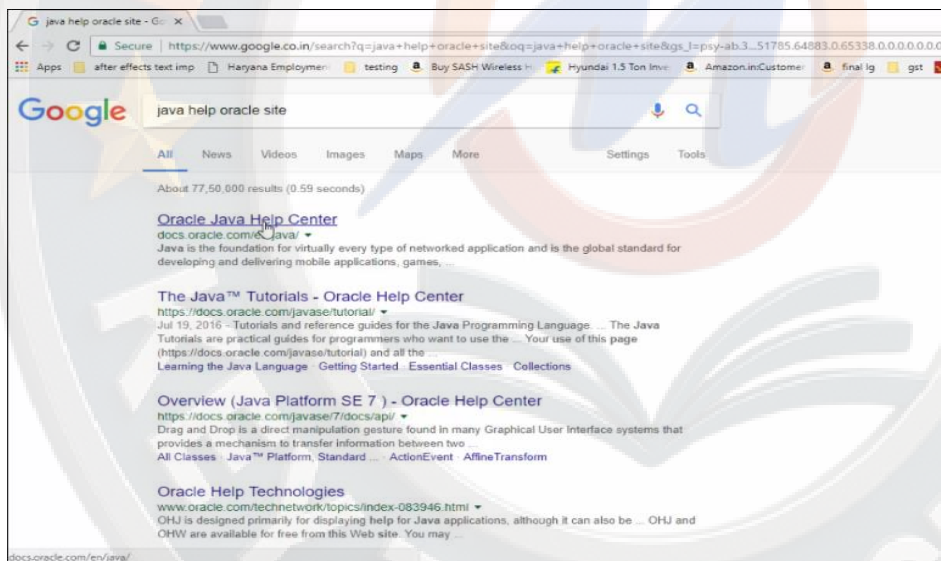
इसी तरह से किसी word को select करके help में से dynamic help पर क्लिक करे तो हमारे पास info available हो जाएगी



यहा पर हमारे पास information available हो गई है यहा पर हमारे पास काफी सारे links available हो जायेंगे जोकि eclipse में inbuilt है इन links पर क्लिक करके हम information ले सकते है इस तरह हम help का use कर सकते है



इसी तरह से हम google में किसी भी topic की जानकारी ले सकते हैं आप किसी भी topic को type करके search करे तो उससे related information available हो जाती है आप जिस वेबसाइट से जानकारी लेना चाहते हैं उस पर क्लिक करके ले सकते हैं



इसी तरह से हम java की वेबसाइट पर जाकर भी किसी topic की जानकारी ले सकते हैं यहा से link पर क्लिक करे

The screenshot shows the Oracle Help Center search results for the query 'system'. The browser address bar displays 'docs.oracle.com/apps/search/search.jsp?category=java&product=&q=system'. The search results are filtered by the 'Java' category, which is selected in the left-hand 'Categories' list. The main content area shows a list of search results, with the top result being 'JDK 8 Installation on the Oracle Solaris Operating System'. This result includes a link to the installation guide and a section titled 'Accessing Resources' which explains that a system resource is either built-in to the system or kept by the host. The page also shows pagination controls indicating 1 to 20 of 145542 results.

यहाँ से particular topic पर सर्च कर सकते हैं उससे related सभी information available हो जाएगी हमारे पास आगे से आगे links available होते जाएंगे जिससे हम info प्राप्त कर सकते हैं इस तरह से java की online help का use कर सकते हैं

Basic Error In Java

```

1 public class ConditionalExp
2 {
3     public static void main(String args[])
4     {
5
6         int x = 70;
7         int y = 60;
8         int z = (x>y) ? x : y;
9
10        System.out.println("Greater Value is : "+z);
11    }
12 }
13
14

```

Problems | Javadoc | Declaration | Console

<terminated> ConditionalExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 19, 2017, 10:51:10 PM)

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Syntax error, insert ")" to complete ClassBody

at ConditionalExp.main(ConditionalExp.java:14)

अब हम errors के बारे में जानेंगे जोकि कोई भी program बनाते समय user कर सकता है हमें इन्हें ध्यान में रखना है

- 1 अगर हम किसी ब्रैकेट का use ना करे तो हमारे पास left side पर red color में एक cross available हो जाता है या फिर अगर हम इसमें error होने पर भी run करे तो box available होगा जिसमें error की जानकारी दी जाएगी अगर हम proceed करे तो हमारे पास नीचे error की जानकारी दी जाएगी या फिर हम cross पर कर्सर ले जाए तो भी हमें जानकारी दी जाएगी इस तरह हमें इसका ध्यान रखना चाहिए
- 2 इसी तरह अगर हम कहीं पर semicolon का use ना करे तो error available हो जाता है हमें java में error की जानकारी साथ में ही pointout की जाती है किसी basic text editor में ऐसा नहीं होता है
- 3 इसी तरह अगर हम कहीं पर small letter का use कर लेते हैं और उसी letter को कहीं पर capital letter में use करते हैं तो भी हमारे पास error available हो जाता है इस तरह से कुछ basic errors होते हैं जिनका हमें program बनाते समय ध्यान में रखना चाहिए

Control Statements

We can control the flow of program using control flow statements like

For taking decisions :

- **If Statement**
- **If-else Statement**
- **Switch Statement**

For Repetition workout

Loops

- **For loop**
- **while loop**
- **Do while loop**

अभी तक हमने basic programs की जानकारी ली थी अब हम control statements के बारे में जानकारी लेंगे control statements से हम program के flow को control कर सकते हैं जैसे कि अगर किसी student के 35 से ऊपर marks हैं तो वह पास है और अगर 35 से कम है तो fail कहा जाए यहा पर हमारा काम marks देने का है बाकि कार्य control statements का है इस तरह के decisions लेने के लिए if statement, if-else statement, nested if statements का use कर सकते हैं और अगर हमारे पास काफी सारी options हो और उनमे से किसी एक को choose करना है तो उसके लिए switch statement का use कर सकते हैं

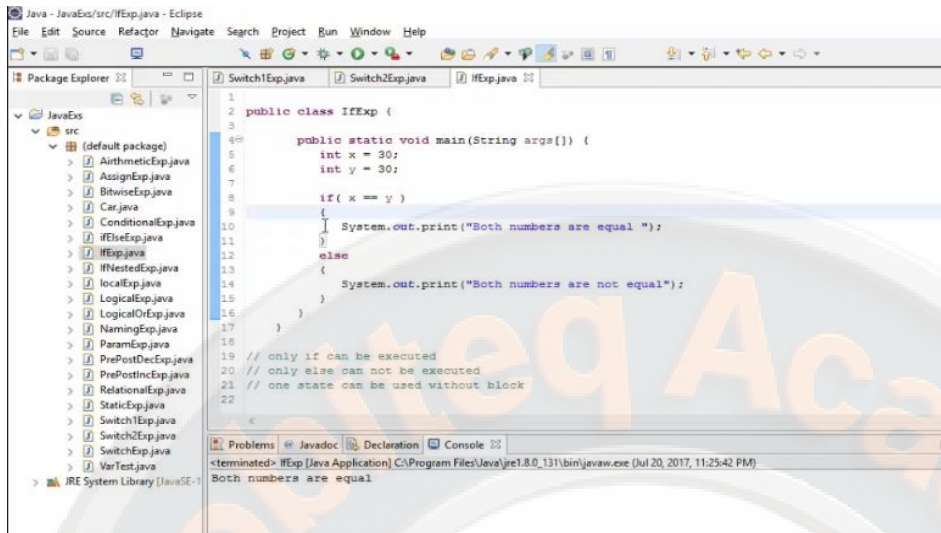
इसी तरह repeted workout करने के लिए हमारे पास loops available हैं जो काम हमे बार बार करना पड़ता है उसके लिए हम loops का use कर सकते हैं loops में हमारे पास तीन प्रकार की loops available हैं इनका use हम आसानी से कर सकते हैं loop से हमे बस एक बार कार्य बताना है उसके बाद control statements से वह automatic repeat होता जाता है इस तरह से हम इनका use करेगे

Control statements : "If"

- "If" is a control statement to execute a single statement of code/ a block of code, "if" needs to provide condition following by "if" block for statements with "else" block for otherwise options.
- based on condition it takes decisions like
 - ▣ if condition is true than "if" block is executed
 - ▣ if condition is false than it skips "if" block and "else" block is executed .

```
if(condition)
{
    // execute if condition is true
}
else
{
    // execute if condition is false
}
```

हम सबसे पहले if statement के बारे में जानेगे यदि हम आपनी programming में कोई decision लेना चाहते है तो हम if का use कर सकते है यह एक control statement है इसके द्वारा चाहे तो single statement को या block of code को execute कर सकते है जैसे की हम right side में देख सकते है इसमें if दिया हुआ है और rounded ब्रैकेट में एक condition दी हुई है इसके आधार पर if decision लेता है जो condition दी है अगर वह true है तो if block execute होगा अगर condition false है तो if block execute नहीं होगा उस case में else block execute होगा दोनों cases में कोई एक block ही execute होगा यदि हम condition के बाद एक ही line use कर रहे है तो उसमे block की जरूरत नहीं होती है



```
1 public class IfExp {
2
3
4 public static void main(String args[] ) {
5     int x = 30;
6     int y = 30;
7
8     if( x == y )
9     {
10        System.out.print("Both numbers are equal ");
11    }
12    else
13    {
14        System.out.print("Both numbers are not equal");
15    }
16 }
17 }
18
19 // only if can be executed
20 // only else can not be executed
21 // one state can be used without block
22
```

Both numbers are equal

हमने आपने program को create किया यहा पर हम x और y को देखना चाहते है कि यह बराबर है जोकि बराबर है हमने condition लगाई है और इनमे से कोई एक block execute होगा हमने run किया और हमारे पास output available है इस तरह से हमारे पास if block execute हुआ है

```

1
2 public class IfExp {
3
4     public static void main(String args[] ) {
5         int x = 30;
6         int y = 40;
7
8         if( x == y )
9         {
10            System.out.print("Both numbers are equal ");
11        }
12        else
13        {
14            System.out.print("Both numbers are not equal");
15        }
16    }
17 }
18
19 // only if can be executed
20 // only else can not be executed
21 // one state can be used without block
22
<terminated> IfExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 20, 2017, 11:26:17 PM)
Both numbers are not equal

```

इसी तरह अगर हम value change कर देते हैं और run करते हैं तो हमारे पास condition false होने के कारण else block execute हुआ है इस तरह हम इसका use कर सकते हैं अगर हमारी value same है और हमने else block नहीं बनाया है तो भी हमारा program execute होगा और अगर value same नहीं है और else block नहीं बनाया है तो हमारे पास कोई output available नहीं होगा हम सिर्फ else block बनाकर execute नहीं कर सकते हैं इसी तरह एक statement के लिए हमें block बनाने की जरूरत नहीं है अगर एक से ज्यादा statement है तो block का use करेंगे इस तरह से हम if का use कर सकते हैं

Control statements : "Nested if statement"

- We use many "if" in one block of code and try to workout on based conditions.
- One "if" can be used in another "if" or in another "else" block. And that is called nested if .

IfNestedeExp.java

```

if(marks>=80)
{
    System.out.println("you have got merit");
}
else
{
    if((marks>=60) && (marks<80))
    {
        System.out.println("you have got first divsion");
    }
    else
    {
        if((marks>=40) && (marks<60))
        {
            System.out.println("you have got second divsion");
        }
        else
        {
            System.out.println("appear again In same class");
        }
    }
}
}

```

अब हम nested if के बारे में जानेगे यदि हम एक if को दुसरे if में और else में use करे तो वह nested if कहलाता है इसका example right side पर available है यदि कि student के marks 80 या 80 से ज्यादा है अगर है तो you have got merit होगा इसका मतलब if वाला section execute हो गया है और उसके बाद हमारा program खत्म हो जाएगा परन्तु अगर marks 80 से कम है तो इसके लिए else section execute होगा |

इसके बाद पूरा else section है इसका मतलब अगर marks 80 से कम है तो else section execute होगा परंतु जैसे ही हम else section में आते हैं तो यहा पर एक ओर if available हो जाता है जो ये check करता है कि क्या marks 60 से ज्यादा और 80 से कम है यहा पर हमने दो condition लगाई है इसलिए हमने यहा पर logical and operator का use किया है जिसका अर्थ है कि दोनों condition true होनी चाहिए अगर condition true है तो उसके नीचे वाला message print होगा यदि यह condition सही नहीं है तो यह condition print नहीं होगी और यह नीचे वाले else में चला जाएगा जो कि इससे ऊपर वाले if से connected है इस process को हम nested if कहते है |

इसको practical करके देखते है

```

1 public class IfNestedExp {
2
3
4     public static void main(String[] args) {
5
6         int marks = 10;
7
8         if (marks >= 80)
9         {
10            System.out.println("you have got merit");
11        }
12        else
13        {
14            if ((marks >= 60) && (marks < 80))
15            {
16                System.out.println("you have got first division");
17            }
18            else
19            {
20                if (marks >= 40) && (marks < 60)
21                {
22                    System.out.println("you have got second division");
23                }
24                else
25                {
26                    System.out.println("appear again in same class");
27                }
28            }
29        }
30    }
31 }
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Console Output:
<terminated> IfNestedExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 20, 2017, 11:54:49 PM)
you have got second division

हमने आपने program को create किया है यहा पर हमने int 55 लिया है और run किया तो हमारे पास output available है जिसमे you have got second division का

message print हुआ है इसी तरह अगर हम int marks में 40 से कम marks रखते है और run करते है तो हमारे पास appear again इन same class का message print होगा इस तरह हमारे पास last else section execute हुआ है इस तरह से हम nested if का use कर सकते है

Control statements : "if-else-ladder

- We use many if in one block code and try to workout on based conditions
- You can go for many if else blocks with ladder.

IfElseExp.java

```
if(marks<40)
{
    System.out.println("Student is fail");
}
else if(marks>=40 && marks<50)
{
    System.out.println("E grade");
}
else if(marks>=50 && marks<60)
{
    System.out.println("D grade");
}
else
{ // none of the above
}
```

अब हम if else ladder के बारे में जानेगे इसमें हम else if का use बार बार कर सकते है इसमें हम एक else if के साथ दूसरा else if का use कर सकते है जैसे की हमने example में if का use किया है जिसमे 40 की condition है उसके बाद if block है और इसके बाद else if का block है जिसमे हमने कुछ condition दी है और उसके बाद एक ओर else if का use किया है और सबसे नीचे else block create किया है इस तरह से conditions को verify किया गया है हम जितने चाहे उतने else if का use कर सकते है

KEY TO SUCCESS

```

1 public class ifElseExp {
2     public static void main(String[] args)
3     {
4         int marks=75;
5         if(marks<40)
6         {
7             System.out.println("Student is Fail");
8         }
9         else if(marks>=40 && marks<50)
10        {
11            System.out.println("E grade");
12        }
13        else if(marks>=50 && marks<60)
14        {
15            System.out.println("D grade");
16        }
17        else if(marks>=60 && marks<70)
18        {
19            System.out.println("C grade");
20        }
21        else if(marks>=70 && marks<80)
22        {
23            System.out.println("B grade");
24        }
25        else if(marks>=80 && marks<100)
26        {
27            System.out.println("A grade");
28        }
29        else
30        {
31            System.out.println("Please check your marks!");
32        }
33    }
34 }

```

Console Output:
<terminated> #EhseExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 21, 2017, 12:04:59 AM)
B grade

हमने यहा पर आपने program को create किया है और run किया और हमारे पास output available है यहा पर हमने int marks में 75 marks है जिसके कारण output में B grade available हुआ है अगर

हम 100 से ज्यादा marks store करते है तो हमारे पास else block execute होगा जिसमे हमारे पास please check your marks का message available होगा इस तरह से else if का use कर सकते है

Control statements : “switch statement”

- If you have multiple options and you have to select one or more from them you can use switch and get your selection.

SwitchExp.java

```
switch ( variable )
{
case value:
    execute statements if above value match
    break;
case value:
    execute statements if above value match
    break;
default:
    values does not match with any of above;
}
```

अब हम switch statement के बारे में जानेगे जब भी हमारे पास बहुत सारे option available होते हैं और हमे किसी एक option को choose करना होता है तो हम switch statement का use कर सकते हैं अब हम box में example देखते हैं switch variable को accept करता है यहा पर हम condition पास नहीं करते हैं हमे बस यहा पर एक variable को पास करना होता है जो variable इसे पास किया जाता है वह variable switch के अंदर वाले block में case को आपने आप match करता है तो switch में हमारे पास बहुत से case होते हैं जैसे की हमारे पास यहा पर दो case available हैं variable पहले 1st case से compare करता है अगर comparision मिल जाता है तो case 1 की सभी statement execute हो जाएगी और switch खत्म हो जाएगा क्योंकि आगे इसमें break available है अगर variable 1st case से match नहीं होता है तो वह second case से compare करेगा अगर मिल जाता है तो यह execute होगा और 1st case execute नहीं होगा अगर variable दोनों में से किसी case से match नहीं होता है तो ऐसे case में default available रहता है इस तरह से switch कार्य करता है

```

1 public class SwitchExp {
2
3     public static void main(String args[] ) {
4
5         System.out.println("w : withdrawl, d : deposit , m : mini statement , b : balance ");
6
7         char choose = 'w';
8
9         switch(choose) {
10            case 'w' :
11                System.out.println("You can withdraw amount ");
12                break;
13            case 'd' :
14                System.out.println("You can deposit oash in machine");
15                break;
16            case 'm' :
17                System.out.println("you can get mini statement");
18                break;
19            case 'b' :
20                System.out.println(" you can check balance on screen");
21                break;
22            default :
23                System.out.println("Invalid options choose from given above");
24
25            }
26
27         System.out.println("You are now out of switch");
28
29     } // what if we do not use break
30 }

```

Problems | Javadoc | Declaration | Console

```

<terminated> SwitchExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 21, 2017, 12:25:35 AM)
w, withdrawl, d : deposit , m : mini statement , b : balance
You can withdraw amount
You are now out of switch

```

यहा पर हमने आपने program में switch statement को शामिल किया और हमने user के लिए एक message भी print करवाया है और run किया है हमारे पास output available है क्योंकि हमने char choose में w press किया है तो उसके according output available है इसी तरह अगर हम available option में से कोई choose करते है तो उसके according output available होगा हमे यहा पर break का ध्यान रखना है हमे एक से ज्यादा statement को शामिल करने के लिए block create करने हगे

```

1 public class SwitchExp {
2
3     public static void main(String args[] ) {
4
5         System.out.println("w : withdrawl, d : deposit , m : mini statement , b : balance ");
6
7         char choose = 'w';
8
9         switch(choose) {
10            case 'w' :
11                {
12                    System.out.println("You can withdraw amount ");
13                    System.out.println("Your Balance amount is : 12000/- ");
14                    System.out.println("Thanks for withdrawing your money");
15                }
16                break;
17            case 'd' :
18                System.out.println("You can deposit cash in machine");
19                break;
20            case 'm' :
21                System.out.println("you can get mini statement");
22            }
23        }
24    }
25 }

```

<terminated>- SwitchExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 21, 2017, 12:30:00 AM)
 w : withdrawl, d : deposit , m : mini statement , b : balance
 You can withdraw amount
 Your Balance amount is : 12000/-
 Thanks for withdrawing your money
 You are now out of switch

हमने यहा पर एक से ज्यादा statement को शामिल करने के लिए block create किया है और हमने तीन statement को शामिल किया और run किया तो हमारे पास output available है इस तरह से block को create कर सकते है इस तरह से switch का use कर सकते है

switch statement using numbers

```

public static void main(String args[]) {
    System.out.println("1 : withdrawl, 2 : deposit , 3 : mini statement , 4 : balance ");
    int choose = 1;
    switch(choose) {
        case 1 :
            System.out.println("You can withdraw amount ");
            break;
        case 2 :
            System.out.println("You can deposit cash in machine");
            break;
        case 3 :
            System.out.println("you can get mini statement");
            break;
        case 4 :
            System.out.println(" you can check balance on screen");
            break;
        default :
            System.out.println("Invalid options choose from given above");
    }
    // what if we do not use break
}

```

Console Output:

```

1 : withdrawl, 2 : deposit , 3 : mini statement , 4 : balance
you can get mini statement

```

यहा पर हमने switch का ही use किया है इसमें हमने केवल charecter की जगह पर number का use किया है और run किया तो हमारे पास output available है इस तरह हम आसानी से numbers का use भी कर सकते है अगर कोई case

match नहीं होता है तो default case available होगा हम आपनी requierment के हिसाब से charecter या numbers का use कर सकते है

switch statement using strings

```

1
2 public class Switch2Exp {
3
4     public static void main(String args[]) {
5
6         System.out.println("wd : withdrawl, dep : deposit , mini : mini statement , bal : balance ");
7
8         String choose = "bal";
9
10
11         switch (choose) {
12             case "wd" :
13                 System.out.println("You can withdraw amount ");
14                 break;
15             case "dep" :
16                 System.out.println("You can deposit oash in machine");
17                 break;
18             case "mini" :
19                 System.out.println("you can get mini statement");
20             case "bal" :
21                 System.out.println(" you can check balance on screen");
22                 break;
23             default :
24                 System.out.println("Invalid options choose from given above");
25         }
26         // what if we do not use break
27     }
28 }

```

Console Output:

```

<terminated>- Switch2Exp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 21, 2017, 12:40:39 AM)
wd : withdrawl, dep : deposit , mini : mini statement , bal : balance
you can check balance on screen

```

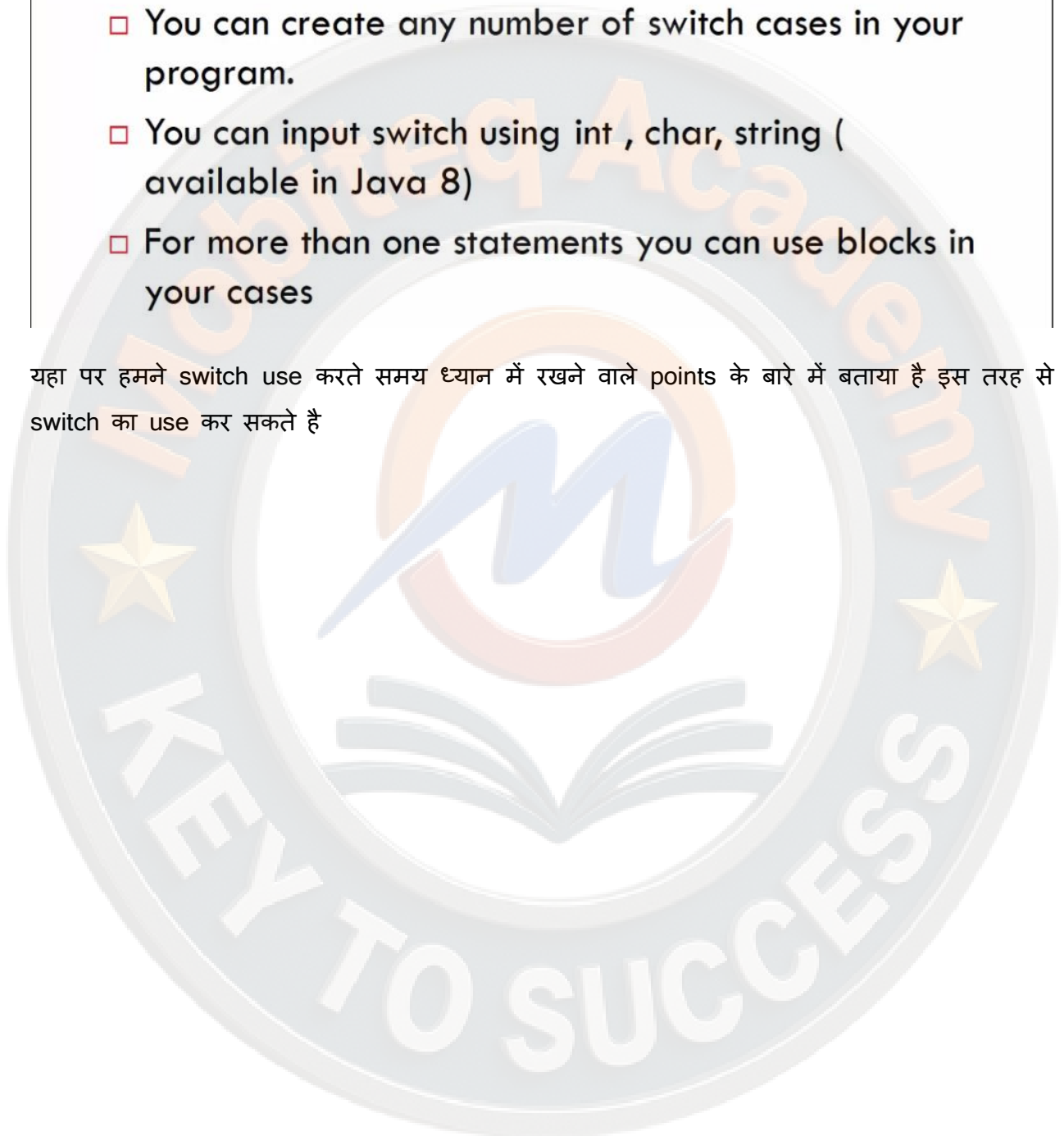
अभी हमने switch के दो example देखे है यह भी उसी की तरह है अबकी बार हम string का use करेगे string में हम उन variable को save करना चाहते है जो user choose करता है जैसे की

user ने balance choose किया तो यह उस variable में save हो जाता है हमने program को run किया और हमारे पास output available है हमने string choose में bal choose किया था इसलिए हमारे पास उसके according output available है इस तरह से string का use करते हुए java में switch का use कर सकते है

Important in switch

- ❑ You can create any number of switch cases in your program.
- ❑ You can input switch using int , char, string (available in Java 8)
- ❑ For more than one statements you can use blocks in your cases

यहा पर हमने switch use करते समय ध्यान में रखने वाले points के बारे में बताया है इस तरह से switch का use कर सकते है



Loops In Java Language

- for loop
- while
- do while

- Every loop has three parts
 - Initialization
 - Condition
 - Iteration : increment or decrement

A loop statement in Java programming language is used to repeatedly execute statements till a provided condition is true.

अब हम loops के बारे में जानेगे हमारे पास loops के तीन types available हैं यह तीनों loops एक जैसा ही कार्य करती है loop हमारी एक statement या एक से ज्यादा statement को यदि हम बार बार execute करना चाहते हैं तो उसके लिए loops का use किया जाता है जिससे हम किसी particular task को execute कर सकते हैं जब तक साथ में जुड़ी statement true रहती है अब इसमें तीन चीजे शामिल हैं इसमें हमें initialization की जरूरत होती है

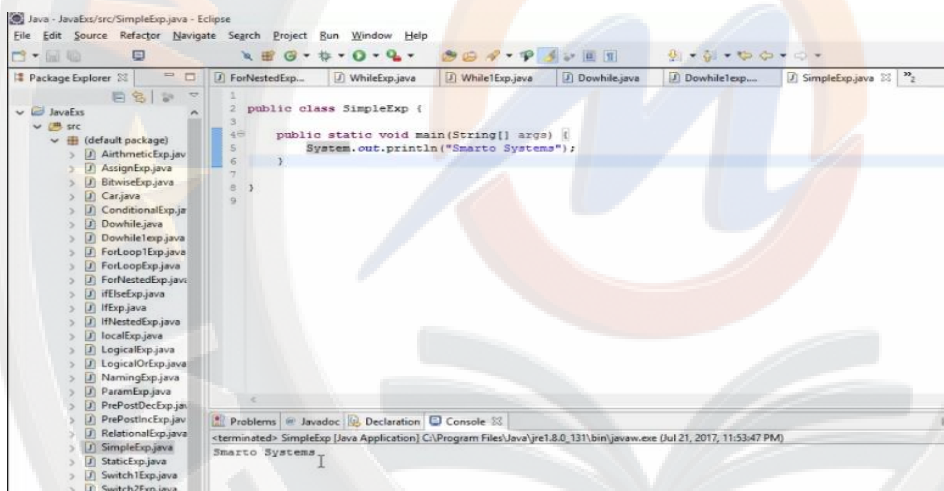
1st Initialization है किसी भी loop को चलाने के लिए उसका starting point बताना जरूरी होता है मान लीजिए हमने 50 lines print करवानी है इसका मतलब शुरुवात 1 से होगी और जब तक 50 line print नहीं हो जाती तब तक यह कार्य चलेगा इसका मतलब हमारी condition यह है कि जब तक हमारा counter 50 से छोटा या 50 के बराबर नहीं हो जाता है तब तक कार्य चलता रहेगा 3rd point iteration है जिसमें increment और decrement दोनों प्रकार से किया जा सकता है इसको हम example से देखते हैं मान लीजिए हम 1 से शुरू होकर 50 तक जाना चाहते हैं तो इसमें 1 में हम क्या जोड़े कि हम 50 तक पहुंच जाए इसमें हम 1 में 1 जोड़े तो 2, इस तरह हम 50 तक पहुंच जाएगे इसका मतलब counter में 1 जोड़कर आगे बढ़ना है और counter - हम decrement के लिए कर सकते हैं तो इस प्रकार यह तीनों part loop के लिए जरूरी होते हैं

Lets take an example

- Write any program to display “Smarto Systems “

```
public class SimpleExp
{
    public static void main(String[] args)
    {
        System.out.println("Smarto Systems");
    }
}
```

इसे समझने के लिए हमने यहा पर एक example लिया है और इसमें हम smarto systems print करवाना चाहते है हमने इसे practical किया है



हमने आपने program को create करके run किया है और हमारे पास output में 1 बार print हो गया है अगर हमे इसी line को 5 बार print कराना चाहते है तो हम इस line को copy करके paste कर सकते है और run कर सकते है तो हमारे पास output available हो जाता है लेकिन अगर हमे इसी line 500 या 5000 बार print करना हो तो हम loop का use करेगे

Loops In Java Language

- for loop
- while
- do while

- Every loop has three parts
 - Initialization : `int i=1;`
 - Condition : `i<=500;`
 - Iteration : increment or decrement
 - `i++` or `i--` or `i=i+5` , or `i=i-5;`

A loop statement in Java programming language is used to repeatedly execute statements till a provided condition is true.

हमारी तीनों loop में यह तीन पार्ट होने जरूरी है इससे हम बता सकते हैं कि हमें काम कितनी बार करना है initialization में हम जहाँ से शुरू करना चाहते हैं इसे बताने के लिए java में variable को create करना होता है हमने `i = 1` store किया है और उसके बाद condition जब तक `i <= 500` तक ना हो जाए तब तक यह चलता रहे इसके लिए हम iteration का use करते हैं यदि हम 1 जोड़ना हैं तो `i++` और घटाने के लिए `i--` का use करेंगे और यदि एक से ज्यादा जोड़ना यह घटना है तो `i = i + 5` और `i = i - 5` का use करेंगे इस तरह जब तक condition true रहती है तब तक loop चलेगी

for loop in java

- Every loop has three parts
 - ▣ Initialization : `int i=1;`
 - ▣ Condition : `i<=500;`
 - ▣ Iteration : increment or decrement
 - `i++` or `i--` or `i=i+5` , or `i=i-5;`

```
for(int i=1; i<=50; i++)
{
    System.out.println("Smarto Systems");
}
```

ForLoop1Exp.java

ForLoopExp.java

अब हम for loop के बारे में जानेगे जैसे की हमें एक line को 50 या 500 बार print करवाना है तो हम for loop का use कर सकते है इसके लिए for block को create करना है इसके लिए हम for keyword का use किया है और दो rounded ब्रैकेट में तीन पार्ट को specify किया है इसमें हमने i variable create किया है और उसमें 1 assign किया है उसके बाद semicolon use किया है i जब तक 50 तक नहीं पहुँच जाता है यह हमारी condition है उसके बाद हमने iteration बताया है की 1 जोड़कर चलेगा इसके बाद semicolon का use नहीं किया है क्योंकि इसके बाद कोई पार्ट available नहीं है और जो कार्य करवाना है उसे हमने दो curly ब्रैकेट में दिया है हमें अगर इस line 500 बार print करना है तो हमें बस 50 के आगे एक 0 लगनी है और run करना है इस तरह हम loop का use कर सकते है

```

Java - JavaEcs/src/ForLoopExp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
src
  (default package)
  ArithmeticExp.jav
  AssignExp.java
  BitwiseExp.java
  Car.java
  ConditionalExp.ja
  Dowhile.java
  Dowhile1exp.java
  ForLoop1Exp.java
  ForLoopExp.java
  ForNestedExp.jav
  IfElseExp.java
  IfExp.java
  IfNestedExp.java
  localExp.java
  LogicalExp.java
  LogicalOrExp.jav
  NamingExp.java
  ParamExp.java
  PrePostDecExp.ja
  RelationalExp.ja
  SimpleExp.java
  StaticExp.java
  Switch1Exp.java
  Switch2Exp.java
  Switch3Exp.java
  VarTest.java
  While1Exp.java
  ForLoopExp.java
  ForLoop1Exp.java
  ForNestedExp.java
  SimpleExp.java
  public class ForLoopExp {
  2
  3
  4 public static void main(String[] args) {
  5
  6     for(int i=1; i<=500; i++)
  7     {
  8         System.out.println("Smarto Systems "+i);
  9     }
  10
  11
  12
  13 }
  14
  Problems Javadoc Declaration Console
  <terminated> ForLoopExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 22, 2017, 12:05:15 AM)
  Smarto Systems 495
  Smarto Systems 496
  Smarto Systems 497
  Smarto Systems 498
  Smarto Systems 499
  Smarto Systems 500
  Writable Smart Insert 6:28
  
```

यहा पर हम for loop के example को practical किया और run करने के बाद हमारे पास output में एक line 500 बार print हो गई है इस तरह हम किसी एक काम को बार कर सकते है अगर 5000 बार करना है तो एक 0 लगाई और run किया तो यह 5000 बार print हो जाएगा इस तरह हम for loop का use आसानी से कर सकते है

```

Java - JavaEcs/src/ForLoopExp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
src
  (default package)
  ArithmeticExp.jav
  AssignExp.java
  BitwiseExp.java
  Car.java
  ConditionalExp.ja
  Dowhile.java
  Dowhile1exp.java
  ForLoop1Exp.java
  ForLoopExp.java
  ForNestedExp.jav
  IfElseExp.java
  IfExp.java
  IfNestedExp.java
  localExp.java
  LogicalExp.java
  LogicalOrExp.java
  NamingExp.java
  ParamExp.java
  PrePostDecExp.ja
  RelationalExp.ja
  SimpleExp.java
  StaticExp.java
  Switch1Exp.java
  Switch2Exp.java
  Switch3Exp.java
  VarTest.java
  While1Exp.java
  ForLoopExp.java
  ForLoop1Exp.java
  ForNestedExp.java
  SimpleExp.java
  public class ForLoopExp {
  2
  3 public static void main(String args[]){
  4     for(int i=500; i>=1; i--)
  5     {
  6         System.out.println("Smarto Systems "+i);
  7     }
  8 }
  9
  Problems Javadoc Declaration Console
  <terminated> ForLoopExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 22, 2017, 12:06:46 AM)
  Smarto Systems 7
  Smarto Systems 6
  Smarto Systems 5
  Smarto Systems 4
  Smarto Systems 3
  Smarto Systems 2
  Writable Smart Insert 4:26
  
```

उसी तरह हमने यहा पर decrement का use किया है और हमारे पास output available है इस तरह हम इसका use भी कर सकते है अगर हम program में condition को गलत डिफाइन करते है तो यह execute नही होगी

Nested For Loop In Java

```
for(int r=1;r<=4;r++)
{
    for(int c=1;c<=5;c++)
    {
    }
}
```

If we use one loop inside another , that is called Nested loop , here is for loop example for nested loop.

ForNestedExp.java

इसी तरह से nested for loop है जिसमें हम एक loop में दूसरी loop का use कर सकते हैं

```
public class ForNestedExp {
    public static void main(String[] args) {
        for(int r=1;r<=4;r++)
        {
            System.out.println();
            System.out.print(" r "+r);
            for(int c=1;c<=5;c++)
            {
                System.out.print(" c "+c);
            }
        }
    }
}
```

Console Output:

```
r 1 c 1 o 2 c 3 o 4 c 5
r 2 c 1 c 2 c 3 o 4 c 5
r 3 c 1 c 2 c 3 c 4 c 5
r 4 c 1 c 2 c 3 c 4 c 5
```

यहा पर हमने nested for loop का use किया है इसमें हमने दो loop का use किया है इसे हमने run किया तो हमारे पास output available है इसमें हमारे पास r 1 है जैसे ही हमने आउटर loop में enter किया तो r को print करवाया है उसके बाद हम अगली loop में आ जाएगे जोकि inner loop है जिसमें सबसे पहले c में 1 print होगा फिर 2,3 और 4 print होगा अब हम inner loop से exit होंगे जैसे ही हम बाहर गए तो उसे पता लग जाएगा की अभी r 1 था अभी वह 4 के बराबर नहीं हुआ है तो फिर से r को plus किया जाएगा और उसके बाद 2nd time इसे print करवाया जाएगा उसके बाद फिर से inner loop होगी जिसमें c को 5 तक print करवाएंगे इस तरह nested loop से row और column का combination बनाते जाएंगे इस तरह से इसका use कर सकते हैं

While loop

Loops In Java Language

- for loop
- **while**
- do while

- Every loop has three parts
 - **Initialization**
 - **Condition**
 - **Iteration : increment or decrement**

A loop statement in Java programming language is used to repeatedly execute statements till a provided condition is true.

अब हम while loop के बारे में जानेगे इसमें भी तीनो पार्ट important रहते है यह भी उसी loop की तरह कार्य करती है

while loop in java

- Every loop has three parts
 - **Initialization : int i=1;**
 - **Condition : i<=500;**
 - **Iteration : increment or decrement**
 - **i++ or i- - or i=i+5 , or i=i-5;**

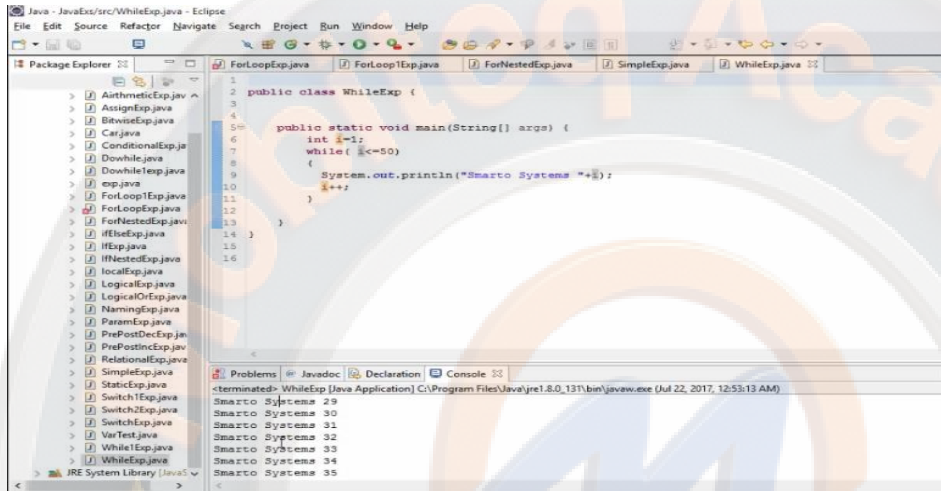
```
int i=1;

while(i<=50)
{
    System.out.println("Smarto Systems");
    i++;
}
```

While1Exp.java

WhileExp.java

while loop structure में अलग है लेकिन काम वैसा ही करता है इसमें हमने तीनों पार्ट्स का use किया है हमने यहा पर while block को use किया है इसमें rounded ब्रैकेट में condition को डिफाइन किया गया है इसमें तीनों पार्ट अलग अलग mention किए गए हैं initialization को हम condition से बाहर रखते हैं जैसे int i= 1 है उसके बाद condition है और while loop के block के बीच में हम iteration specify करते हैं हमने statement को block में शामिल किया है



```

1 public class WhileExp {
2
3
4     public static void main(String[] args) {
5         int i=1;
6         while( i<=50)
7         {
8             System.out.println("Smarto Systems "+i);
9             i++;
10        }
11    }
12 }
13
14
15
16

```

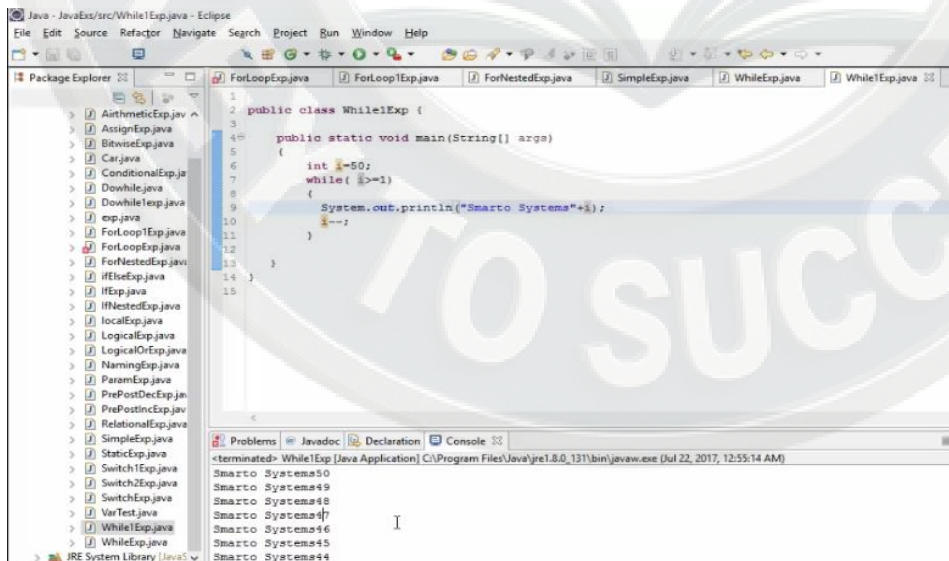
Problems | Javadoc | Declaration | Console

```

<terminated> WhileExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 22, 2017, 12:53:13 AM)
Smarto Systems 29
Smarto Systems 30
Smarto Systems 31
Smarto Systems 32
Smarto Systems 33
Smarto Systems 34
Smarto Systems 35

```

यहा पर हम while loop के example को practical किया है और हमारे पास output available है a इसमें अगर हमारी condition गलत है तो loop execute नहीं होगी इस तरह से while loop का use कर सकते हैं



```

1 public class While1Exp {
2
3
4     public static void main(String[] args)
5     {
6         int i=50;
7         while( i>=1)
8         {
9             System.out.println("Smarto Systems"+i);
10            i--;
11        }
12    }
13
14
15

```

Problems | Javadoc | Declaration | Console

```

<terminated> While1Exp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 22, 2017, 12:55:14 AM)
Smarto Systems50
Smarto Systems49
Smarto Systems48
Smarto Systems47
Smarto Systems46
Smarto Systems45
Smarto Systems44

```

हमने यहा पर decrement का example को practical किया है जिसमे हमारे पास output available है जिसमे की 50 से 1 की तरफ बड़ा जा रहा है क्योंकि i -- किया जा रहा है

इस तरह से हम while loop का use कर सकते हैं

do while loop in java

- Every loop has three parts
 - Initialization : `int i=1;`
 - Condition : `i<=500;`
 - Iteration : increment or decrement
 - `i++` or `i--` or `i=i+5` , or `i=i-5;`

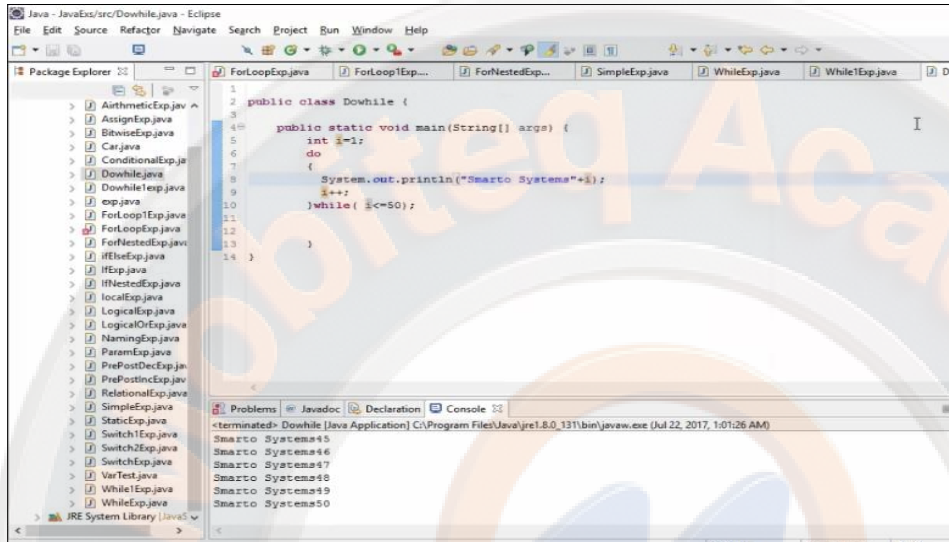
```
int i=1;

do
{
    System.out.println("Smarto Systems");
    i++;
} while(i<=50);
```

Dowhile1Exp.java

DowhileExp.java

अब हम do while loop के बारे में जानेगे इसमें भी तीनों पार्ट important रहते हैं इसमें structure different है लेकिन काम वही होगा इसमें हमने initialization point पहले डिफाइन किया है जोकि `i =1` है उसके बाद do से block को open किया है इसमें हमने do के साथ किसी condition को डिफाइन नहीं किया है do से loop को शुरू किया है उसके ब्रैकेट को open करके statement को डिफाइन किया है उसके बाद iteration डिफाइन की है जोकि block के अंतर्गत है फिर block को close किया गया है और condition को specify किया है उसके बाद semicolon का use किया है इसका ध्यान जरूर रखे नहीं तो error available होगा



```

1 public class Dowhile {
2
3
4     public static void main(String[] args) {
5         int i=1;
6         do
7         {
8             System.out.println("Smarto Systems"+i);
9             i++;
10        }while( i<=50);
11
12    }
13
14 }

```

Problems | Javadoc | Declaration | Console

```

<terminated> Dowhile [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 22, 2017, 1:01:26 AM)
Smarto Systems45
Smarto Systems46
Smarto Systems47
Smarto Systems48
Smarto Systems49
Smarto Systems50

```

हमने यहा पर दो while के example को practical किया है और हमारे पास output same available होता है इसी तरह हम decrement का use भी कर सकते है

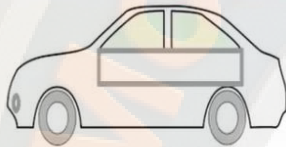
इस loop में हम condition को नीचे डिफाइन करते है जिसमें condition गलत भी हो तो statement एक बार execute जरूर होगी क्योकि बाद में वह loop में आता है तो वह check करता है की condition तो गलत है तो कोई ऐसी सिचुएशन जिसमे आप एक बार execute करना चाहते है चाहे condition गलत या सही हो तो हम do while का use कर सकते है

Method In Java

Class has properties and behaviours

Class : A class can be defined as a template/blueprint which has its fields (variables) and **behaviours (methods)**, used to create its kind of objects. Everything is encapsulated in Java under classes.

Constructor is a special method having same name of class.



Class car



Object nano



Object maruti

```
class Car
{
    String color;
    int modelNo;
    int speed;
}
public void setSpeed(int givenSpeed)
{
    speed=givenSpeed;
}
```

Class car

```
Car nano=new Car();
nano.color="red";
nano.modelNo=2007;
nano.setSpeed(100);
```

Object nano

```
Car maruti=new Car();
maruti.color="blue";
maruti.modelNo=2010;
maruti.setSpeed(140);
```

Object maruti

जब हमने class के बारे में जाना था तो उसमें हमने variables और method के बारे में बात की थी हमने variables की जानकारी ले ली है लेकिन अब हम methods के बारे में जानेंगे जब हम किसी specific code को एक नाम देकर execute कर सकते हैं उसे हम method कहते हैं जैसे की हम car की speed बढ़ानी है और इसकी coding लिखनी है तो उसके हम set speed नाम दे देते हैं जैसे ही set speed का use करते हैं तो यह हमारा एक method है जिससे हम car की speed को कम या ज्यादा कर सकते हैं इसमें हम constructor जोकि एक special method होता है इसका नाम class के नाम जैसा होता है इससे हम data members को initialize करा सकते हैं कुछ task कर सकते हैं इसके बारे में हम जानकारी लेंगे

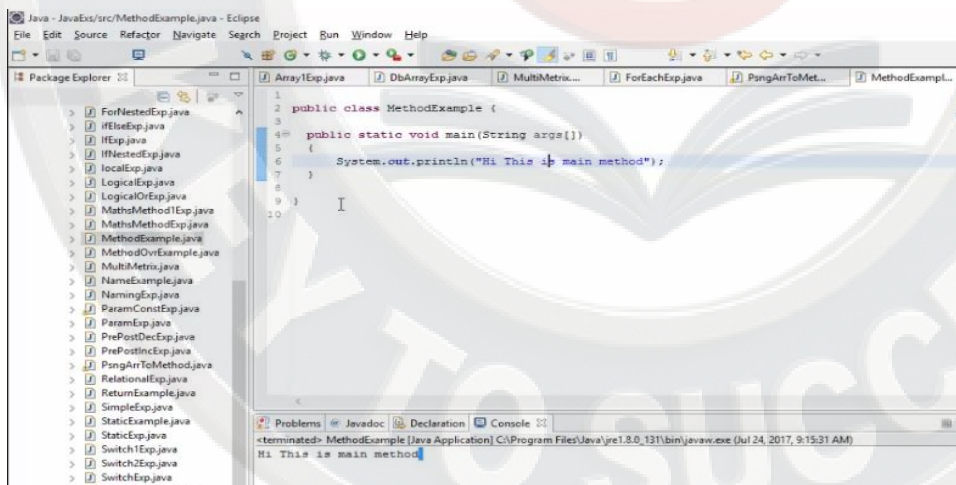
Methods in Java

- Methods are collection of statements which are grouped together in a block to perform desired task.

```
public static void main(String[] args)
{
    // statements are defined here.
}
```

MethodExample.java

method एक collection of statements है जिसमें काफी सारी statement को collect करके रख सकते हैं जैसे ही collect करते हैं तो यह एक block format में आ जाता है और इसका हम एक नाम रख सकते हैं जैसे कि नीचे एक example है जिसका use हम अपने program में शुरू से करते आ रहे हैं यहाँ पर main हमारा method है



यहाँ पर हमने एक line print करवाई है इस तरह से public static void main एक default method है जोकि class के साथ use किया जाता है उसकी coding को execute करने के लिए use किया जाता है

Two types are methods

□ Predefined

- Already defined in java.
 - Mathematical methods
 - `Math.random();`
 - `Math.max();`
 - String methods

MathsMethodExp.java

MathsMethod1Exp.java

□ User defined

- We can create own required methods with own desired code in java

<https://docs.oracle.com/javase/7/docs/>
<https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html>

हमारे पास method के दो types available हैं predefined जोकि पहले से ही defined है और user defined जो हम defined करेगे

Predefined method में पहले से defined method available है जोकि classes के साथ जुड़े है जैसे की हमारे पास math class है और उसके काफी सारे method पहले से available है जिनसे defined task कर सकते है इसी तरह से string class है और उसके method available है

User defined method जो हम खुद create करेगे और सारी की सारी coding हमारे द्वारा की जाएगी हमने image में नीचे कुछ URL दिए है जिसमे हम कुछ example को देख सकते है जिसमे class के particular method की जानकारी ले सकते है हमने यहा पर नीचे वाले URL को क्लिक करके open कर लेते है फिर हमने chorme पर क्लिक करके ok किया

Summary Nested | Field | Const | Method Detail: Field | Const | Method

java.lang
Class Math
 java.lang.Object
 java.lang.Math

public final class Math
 extends Object

The class Math contains methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions. Unlike some of the numeric methods of class StrictMath, all implementations of the equivalent functions of class Math are not defined to return the bit-for-bit same results. This relaxation permits better-performing implementations where strict reproducibility is not required.

By default many of the Math methods simply call the equivalent method in StrictMath for their implementation. Code generators are encouraged to use platform-specific native libraries or microprocessor instructions, where available, to provide higher-performance implementations of Math methods. Such higher-performance implementations still must conform to the specification for Math.

The quality of implementation specifications concern two properties, accuracy of the returned result and monotonicity of the method. Accuracy of the floating-point Math methods is measured in terms of *ulp*, units in the last place. For a given floating-point format, an *ulp* of a specific real number value is the distance between the two floating-point values bracketing that numerical value. When discussing the accuracy of a method as a whole rather than at a specific argument, the number of *ulps* cited is for the worst-case error at any argument. If a method always has an error less than 0.5 *ulps*, the method always returns the floating-point number nearest the exact result; such a method is correctly rounded. A correctly rounded method is generally the best a floating-point approximation can be. However, it is impractical for many floating-point methods to be correctly rounded. Instead for the Math case, a larger error bound of 1 or 2 *ulps* is allowed for certain methods. Informally, with a 1 *ulp* error bound, when the exact result is a representable number, the exact result should be returned as the computed result; otherwise, either of the two floating-point values which bracket the exact result may be returned. For exact results large in magnitude, one of the endpoints of the bracket may be infinite. Besides accuracy at individual arguments, maintaining proper relations between the method at different arguments is also important. Therefore, most methods with more than 0.5 *ulp* errors are required to be semi-monotonic: whenever the mathematical function is non-decreasing, so is the floating-point approximation. Not all approximations that have 1 *ulp* accuracy will automatically meet the monotonicity requirements.

Since:
 JDK1.0

Field Summary

Fields

यहा पर हमारे पास available हो गई है इससे हम यह बताना चाहते है कि java की पहले से online help available है जिसे predefined method की जानकारी ले सकते है

Java - JavaEcs/src/MathsMethodExp.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

```

1 public class MathsMethodExp {
2
3     public static void main(String[] args) {
4
5         double x = Math.random();
6         int y = (int) (x*10);
7
8
9
10        System.out.println("Random double value in x : " + x);
11        System.out.println("Random int value in y : " + y);
12
13    }
14
15 }
16

```

Problems | Javadoc | Declaration | Console

<terminated> MathsMethodExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 9:25:23 AM)

Random double value in x : 0.7824464215664543
 Random int value in y : 7

Writable Smart Insert 16:1

हमने यहा पर math method exp.java का example लिया है इसमें हमने एक class ली है और फिर method का use किया है और फिर predefin method के लिए math.random class का use किया है इसमें हमने math जोकि predefined class है उसका use किया है और random जोकि already class में available है इसको हमने double x में store किया है फिर इसको हमने 10 से multiply किया है ताकि जो number है उसे int में convert कर पाए इसके int का use किया है हम यहा पर double से int में आने के लिए int से typecast किया है क्योंकि x double में था फिर y में store होगा उसके बाद हमने double x को और y को print करवाया है हमने run किया है

```

1 public class MathsMethodExp {
2
3
4     public static void main(String[] args) {
5
6         double x = Math.random();
7         int y = (int) (x*10);
8
9
10        System.out.println("Random double value in x : " + x);
11        System.out.println("Random int value in y : " + y);
12
13    }
14
15 }
16

```

Problems | Javadoc | Declaration | Console

<terminated> MathsMethodExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 9:24:45 AM)

Random double value in x : 0.8137752438009906
Random int value in y : 8

तो हमारे पास output में random double value बहुत बड़ी available हुई है और जो नीचे value है वह 8 available हुई है क्योंकि 10 से multiply किया था इस कारण हमारे पास पहला numeric digits है जोकि decimal के बाद है उसे 10 से multiply करने के बाद integer में convert करने के बाद show किया गया है अब यहा पर random method एक random number generate करता है जोकि अलग अलग रहता है

```

1 public class MathsMethodExp {
2
3
4     public static void main(String[] args) {
5
6         double x = Math.random();
7         int y = (int) (x*10);
8
9
10        System.out.println("Random double value in x : " + x);
11        System.out.println("Random int value in y : " + y);
12
13    }
14
15 }
16

```

Problems | Javadoc | Declaration | Console

<terminated> MathsMethodExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 9:25:19 AM)

Random double value in x : 0.6768430106268025
Random int value in y : 6

हमने इसे दोबारा run किया तो हमारे पास output में 6 available हुआ है इस तरह हर बार run करने पर अलग number available होता है इस तरह इसे random number द्वारा generate किया जा रहा है जोकि math class का है

```

1 public class MathsMethod1Exp {
2
3
4 public static void main(String[] args) {
5     int a=40,b=20;
6
7     int c=Math.max(a, b);
8     int d=Math.min(a, b);
9     System.out.println("Max no is "+c);
10    System.out.println("Min no is "+d);
11 }
12 }
13 }
14 }

```

Problems | Javadoc | Declaration | Console

```

<terminated> MathsMethod1Exp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 9:27:21 AM)
Max no is 40
Min no is 20

```

इसी तरह हमने math method1 exp. java का example को practical किया है हमने यहा पर int a और b लिया है और math.max और math.min का use किया है जोकि predefined है और a और b को math max और min में पास किया है जो output आएगा उसे c और d में store किया जाएगा और फिर दोनों को print करवा लिया है हमने run किया और हमारे पास output available है हमे इसने दोनों में से बड़ा number बता दिया है और छोटा number भी बता दिया है इस तरह हम आसानी से predefined method का use करके आपने कार्य को आसानी से कर सकते है

User Defined Methods

Two types are methods

```
public void calTotal()  
{  
a=10;  
b=20;  
c=a+b;  
System.out.println("Total is "+c);  
}
```

□ User defined

- ▣ We can create own required methods with own desired code in java

UserMethodExp.java

<https://docs.oracle.com/javase/7/docs/>
<https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html>

अब हम user defined method के बारे में जानेगे इसे हम खुद create करते है इसके लिए हम left side example को देख सकते है हमने यहा पर public void cal total के नाम से method बनाया है यह हमने खुद से बनाया है इसमें हमने दो variables को लिया है और total को c में store करवाया है और उसे print करवाया है

```

1 public class UserMethodExp {
2     int a,b,c;
3     public void callTotal ()
4     {
5         a=10;
6         b=20;
7         c=a+b;
8         System.out.println("Total is "+c);
9     }
10
11
12
13
14
15 public static void main(String[] args) {
16     UserMethodExp obj=new UserMethodExp ();
17     obj.callTotal ();
18 }
19
20
21
22

```

Problems | Javadoc | Declaration | Console

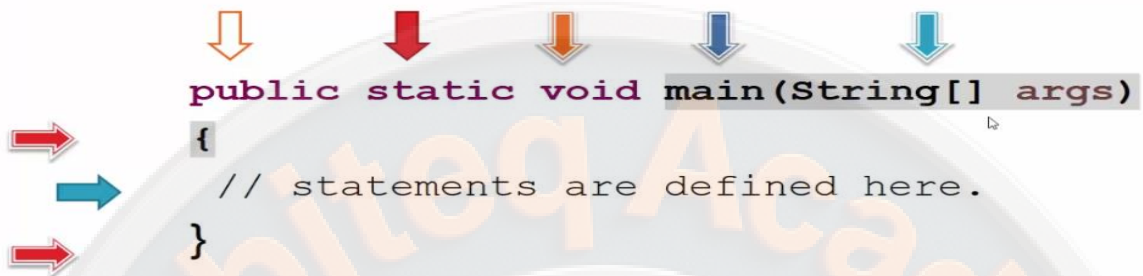
<terminated> UserMethodExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 10:32:25 AM)

Total is 30

हमने यहा पर user defined method को practical किया है हमने variables declare करने के बाद block में user defined method को डिफाइन किया है जिसका नाम caltotal है इसके बाद हमने इसमें statement को शामिल किया है इस तरह हमने group of statement को शामिल किया है अगर हमे इसे execute करना है तो इसे call करना होगा इसके लिए हमने static void main में obj.calltotal से call किया है हमने इसमें user method exp का एक object obj बनाया है फिर obj के साथ . लगाकर इस method को call किया है जब हम इसे run करते है तो यह method execute होकर total show कर देता है इस तरह से इसका use कर सकते है

Methods in Java

```
public static void main(String[] args)
{
    // statements are defined here.
}
```



यहा पर हमारे पास public static method available है जिसे हम use कर रहे थे हम सभी के बारे में जानकारी लेंगे

Access specifiers

Defines the accessibility of any class/ methods/variables .

1. public : Accessibility to public means you can access methods from any class in Java anywhere.
2. private : Accessibility to private means you can access methods within only its declared class.
3. protected: protected means you can access methods within only its declared class or in its subclass.

```
public static void main(String[] args)
{
    // statements are defined here.
}
```

यहा पर हम सबसे पहले access specifiers के बारे में जानेगे हम किसी भी method के आगे public keyword का use कर रहे है इसे हम access specifier कहते है यह किसी class, method और variables की accessibility को डिफाइन करता है अर्थात यदि हम इस समय method के संदर्भ में देखे तो इस method को कहाँ कहाँ use किया जा सकता है यह access specifier decide करता है यह किसी method में सबसे पहले लगाया जाता है और हमारे पास तीन तरह के access specifier available है 1st public इसका अर्थ जो method हम public बनाए उसे java में हम कही भी किसी भी class में use कर सकते है इसके लिए public keyword का use कर सकते है 2nd private जिस class में method को declare करते है केवल उसी class में private access specifier के साथ specify method किए हुए method को use कर सकते है अर्थात private method केवल उसी class में access कर सकते है जिस class में उसे declare या डिफाइन किया गया है हम उस class के बाहर इसका use नही कर सकते है 3rd protected access specifier है इसका अर्थ एक वह class जिसमे उसे declare किया गया है और एक वह class जिसे clear की गई class की subclass है तो वहा पर इसका use कर सकते है अर्थात इसका use दो जगह कर सकते है एक तो declare class जहा उसे बनाया गया है दूसरा उस declare की गई class की subclass में use कर सकते है इस तरह से हम methods की access को specify करते है

```

3 {
4
5     public static void main(String[] args) {
6         second sb=new second();
7         sb.calTotal();
8     }
9
10
11 }
12
13
14 class second{
15
16     int a,b,c;
17     public void calTotal()
18     {
19         a=10;
20         b=20;
21         c=a+b;
22         System.out.println("Total is "+c);
23     }
24 }
25

```

Problems | Javadoc | Declaration | Console

<terminated> AccessSpExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 3:54:10 PM)

Total is 30

हमने यहा पर सबसे पहले public class access spe example हमारे पास main class है जिसमे हमने void main method को भी शामिल किया है दूसरी class हमारे पास second है जिसमे हमने तीन variables को create किया है और public void caltotal एक method डिफाइन किया है यह method दो variables को जोडकर c में store करता है और print करके दिखाता है इसको execute हमने आपनी access spe example class में जिसमे हमने static void main में जाकर second class का एक object बनाया है फिर इस object से हमने particular method को call किया है हमने इसे run किया तो हमारे पास output 30 available हो गया है इस तरह यह method ठीक है यह method public है इसलिए execute हुआ है

The screenshot shows the Eclipse IDE with a Java project. The Package Explorer on the left shows a package named 'src' containing several Java files, including 'AccessSpExample.java'. The main editor displays the following code:

```

3 {
4
5     public static void main(String[] args) {
6         second obj=new second();
7         obj.callTotal();
8     }
9
10 }
11
12 }
13
14 class second{
15
16     int a,b,c;
17     private void calTotal()
18     {
19         a=10;
20         b=20;
21         c=a+b;
22         System.out.println("Total is "+c);
23     }
24 }
25 }

```

The console at the bottom shows the output of the program:

```

<terminated> AccessSpExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 3:54:10 PM)
Total is 30

```

अगर हम इस method को private करते हैं तो हमारे पास error available हो जाता है क्योंकि private होने के कारण हम इसे method में ही use कर सकते हैं इसे हम second class में ही use कर सकते हैं इससे बाहर इसका use नहीं कर सकते हैं इस तरह से public कही भी accessible है और private आपनी ही class में accessible है इस तरह से access specifier का use कर सकते हैं

static methods

You can declare class based methods using static keyword , and can be called by static methods without creating any object of class.

if we don't specify static feature , your method will be instance method and if you want to call them from static methods than we need to create object of class.

```
public static void main(String[] args)
{
    // statements are defined here.
}
```

अब हम static के बारे में जानेगे यह class based method रहते हैं जो class के आपने method होते हैं उन्हें static method कहा जाता है जब हम किसी static keyword के साथ method को declare करते हैं तो वह static method माना जाएगा और अगर static नहीं लगाते हैं तो वह instance method माना जाएगा जिसको class के object के साथ call कर सकते हैं इसको हम non static भी कह सकते हैं इस तरह हम static और non static का use कर सकते हैं जब हम किसी static method से किसी दुसरे static method से call करेगे तो हमे किसी भी object की जरूरत नहीं रहती है यदि हम non static का use करे और उसे static method में call करे तो हमे object की जरूरत रहती है हम इसे example से समझते हैं

```

1 public class StaticExample {
2
3     public static void show() {
4
5         System.out.println("static Method In Java ");
6     }
7
8     public void display() {
9
10        System.out.println("Non static Method In Java ");
11    }
12
13    public static void main(String[] args) {
14
15        show();
16
17        StaticExample obj=new StaticExample();
18        obj.display();
19    }
20
21 }
22
23 }

```

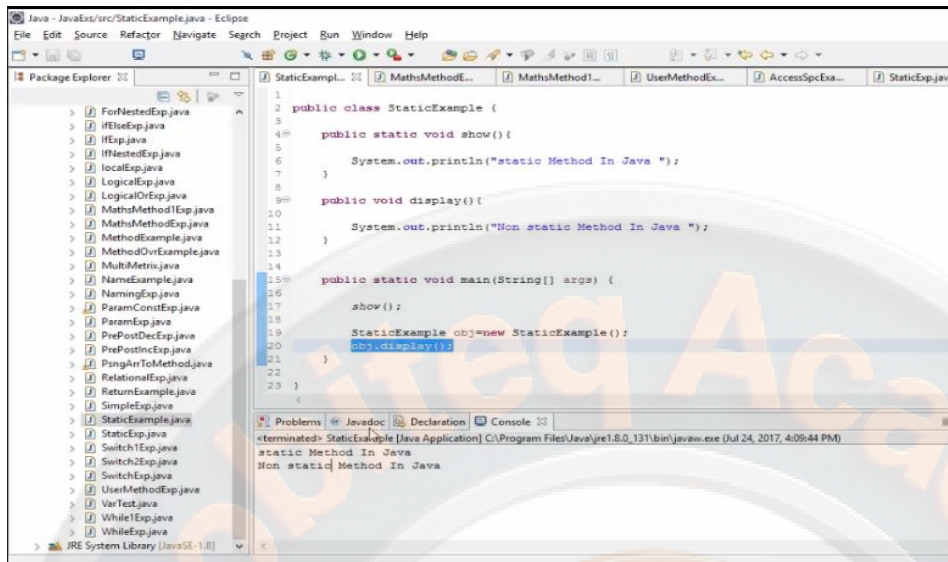
Console Output:

```

<terminated> AccessSpExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 3:56:27 PM)
Total is 30

```

हमने सबसे पहले एक class बनाई है और फिर हमने एक method public static void show का use किया है हमने static keyword का use किया है फिर उसके बाद एक non static method भी use किया है जिसमे हमने static keyword का use नहीं किया है उसके बाद हम public static void main में है जोकि static method है जब किसी static method से static method को call करेंगे तो हमें किसी भी तरह का object create करने की जरूरत नहीं है हमने यहा पर show को direct call किया है अगर हम किसी non static method को call करते हैं जैसे display है ऐसे method instance method कहलाते हैं इसके लिए class के object की जरूरत होती है हमने एक obj create किया है और उसी से display को call किया है जब हम इसे run करते हैं



```
1 public class StaticExample {
2
3
4 public static void show() {
5     System.out.println("static Method In Java ");
6 }
7
8
9 public void display() {
10
11     System.out.println("Non static Method In Java ");
12 }
13
14
15 public static void main(String[] args) {
16
17     show();
18
19     StaticExample obj=new StaticExample();
20     obj.display();
21 }
22
23 }
```

Console Output:

```
<terminated> StaticExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 4:09:44 PM)
static Method In Java
Non static Method In Java
```


तो हमारे पास output available है इस तरह से हम static और non static का use कर सकते हैं

return methods

You can specify return type of any method any data type you want to get in return like int/char.

```
public static int showCalculation()
{
    int total=salary+bonus;
    return total;
}
```

void : is a return data type which means you are not going to return anything from method.



```
public static void main(String[] args)
{
    // statements are defined here.
}
```

अब हम return method के बारे में जानेगे यहा पर हमारे पास void एक return datatype है यदि हम void लगाते है तो जो method है वह कुछ वापस नही करेगा लेकिन यदि हम कोई specific datatype लगाते है जैसे कि int, char, string और double आदि का use करते है हमने यहा पर int लिया है इसका मतलब जो data हमे वापस मिलेगा वह int datatype में होगा इसे वापस करने के लिए हमे return statement को शामिल करना होता है जोकि किसी block में last statement रहती है और return के साथ variables या वह value जो हम वापस करना चाहते है वह यह method हमे वापस करेगा जैसे हमने public static int का use किया है इसमें हमने कुछ कैलकुलेशन की है उसके बाद data को return करने के लिए return keyword का use किया है इसके साथ उसी type का data है जो datatype हमने ऊपर use की है total जोकि एक int datatype का ही data है इस तरह हम data type डिफाइन करके उसी type का data return कर सकते है यदि हम data वापस नही चाहते है तो ऐसे में void का use करते है यह भी एक return datatype है यह किसी प्रकार का data वापस नही करता है यदि हमने ऊपर कोई datatype डिफाइन किया है तो हमे return का use करना जरूरी है

```

1 public class ReturnExample {
2     int total;
3
4     public void showSum(int x,int y)
5     {
6         total=x+y;
7         System.out.println("total is= "+total);
8     }
9
10    public int showTotal(int x,int y)
11    {
12        total=x+y;
13        return total;
14    }
15
16    public static void main(String[] args) {
17        ReturnExample p=new ReturnExample();
18
19        p.showSum(10,20);
20
21        int t=p.showTotal(10, 20);
22        System.out.println("Return total = "+t );
23    }
24
25
26
27
28
29
30
31
32

```

Problems Javadoc Declaration Console

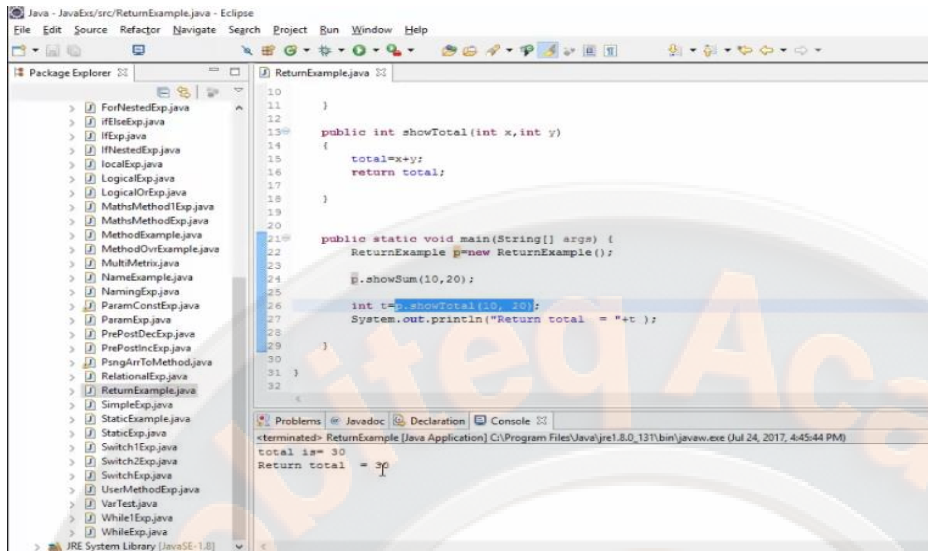
<terminated> StaticExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 4:09:44 PM)

static Method In Java
Non static Method In Java

Writtable Smart Insert 22:21

हमने सबसे पहले एक class डिफाइन की है इसमें हमने total variable लिया है उसके बाद हमने public void show sum method लिया है जोकि simple method है जोकि void datatype का use करता है यह कुछ भी return नहीं करेगा फिर उसके बाद हमने public

int showtotal method लिया है हमने int का use किया है यह हमें कुछ integer वापस करेगा इसके लिए return keyword का use किया है उसके साथ एक variable जो return datatype का है इस प्रकार show sum और show total दो variables हैं उसके बाद हमने स्टाटिव void main में object create किया है जोकि p है p से हमने show sum को call किया है जब call किया तो दो value पास की हैं उसके बाद हमने एक variable int t लिया है क्योंकि show method कुछ वापस करेगा वह integer datatype का या कुछ value हो सकती है जब हमने इसी object p के साथ call किया है उसमें भी value पास की गई है



```
10
11 }
12
13 public int showTotal(int x,int y)
14 {
15     total=x+y;
16     return total;
17 }
18
19
20
21 public static void main(String[] args) {
22     ReturnExample r=new ReturnExample();
23
24     r.showSum(10,20);
25
26     int t=r.showTotal(10,20);
27     System.out.println("Return total = "+t);
28 }
29
30
31 }
32
```

Problems | Javadoc | Declaration | Console

<terminated> ReturnExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 4:45:44 PM)

total is= 30
Return total = 30

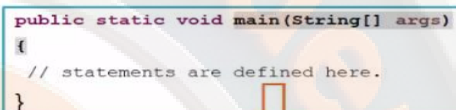
हमने run किया तो हमारे पास output available है इस तरह से हम return method का use कर सकते

क

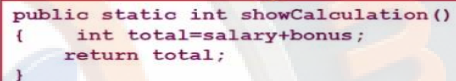
Name of methods

Methods can have any name but should follow java naming conventions.

- Methods names are **Case-sensitive** in Java.
- Methods name **can have Letter, digits** and two special characters **underscore and '\$'** sign.
- Name should with **alphabet at the start** , cannot start with number. Can use underscore and '\$' sign.
- **Following characters** can be letters, digits, \$ or _ character.
- **White space** , special characters like , * ; are **not allowed**.
- Methods name must **not be a keyword** (reserved word).
- If you are using long method name , you can use camelCasing to specify long names.



```
public static void main(String[] args)
{
    // statements are defined here.
}
```



```
public static int showCalculation()
{
    int total=salary+bonus;
    return total;
}
```

अब हम method के नाम रखने के बारे में जानेंगे यहाँ पर कुछ rules दिए गए हैं जिन्हें हम method का नाम रखते समय याद रखें यहाँ पर जो main है वह method का नाम है इसी तरह show calculation method name है यह rules हमने variables के name रखते हुए भी किए थे इस तरह हम इन्हें ध्यान में रखें

```

1 |
2 | public class NameExample {
3 |
4 |
5 |     public void showSum(int x, int y)
6 |     {
7 |         int total=x*y;
8 |         System.out.println("total is= "+total);
9 |     }
10 |
11 |
12 |
13 |     public static void main(String[] args) {
14 |
15 |         NameExample obj=new NameExample();
16 |         obj.showSum(10,30);
17 |     }
18 |
19 |
20 | }
21 |

```

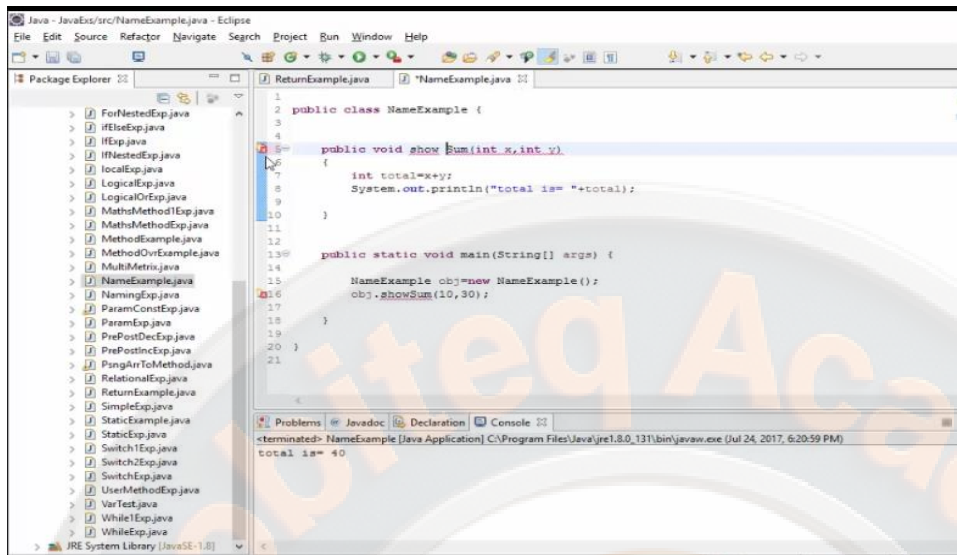
Problems | Javadoc | Declaration | Console

```

<terminated> ReturnExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 4:47:54 PM)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Type mismatch: cannot convert from void to int
    at ReturnExample.main(ReturnExample.java:27)

```

यहा पर हमारे पास example available है हमारे पास एक class है उसमे show sum method use किया है उसके बाद दो parameter है जिसमे हम value पास करेगे वह यह recive करेगे show sum method x और y को जो value recive होती है उसे total में assign करता है जोकि total में store हो जाएगा उसे print किया है अगर हम किसी method को call करते है तो हम उसी नाम का use करते है जो हमने ऊपर use किया था जोकि show sum है उसे हमने call किया है हमने एक object create किया है और value पास की है हमने run किया तो हमारे पास total 40 available हो जाएगा



अब अगर हम show sum में space दे तो हमारे पास error available होगा

इसी तरह अगर comma या dot का use करे तो भी error available होगा हम केवल underscore(_) का use कर सकते हैं

इसी तरह अगर method का name किसी number से शुरू करे तो error available होगा

इस प्रकार method का name रखते समय कुछ rules हैं जिन्हें follow करना होता है इस तरह से method को create कर सकते हैं

Parameter of methods

When you need to provide extra info to any method you can provide by using parameters. You can specify parameters for any method, which are passed when we call that method.

Parameters are optional and you can use them if you are required in programming, but need take care

1. Type
2. Order
3. Number of parameters of a method.

```

public static void main(String[] args)
{ MyClass obj=new MyClass();
  obj.showCalculation(10,20)
}

```

```

public int showCalculation( int a, int b)
{ int total=a+b;
  return total;
}

```

अब हम parameter के बारे में जानकारी लेंगे हम जब भी किसी method को declare करते हैं और यदि उस method को कोई extra information provide करना चाहते हैं जोकि उस method को call करते समय provide की जाती है तो उसके लिए हम parameters का use करते हैं हमने public static void main में string एक array लिया गया है जोकि arguments के नाम से है जोकि args के नाम से है इसी तरह हमने user defined method में लिया है जिसमे showcalculation method है जिसमे हमने दो variables parameter int a है और दूसरा int b है दोनों ही numeric है comma use किया है जोकि as a parameters है इन दोनों को जो value मिलती है जब call किया जाता है तब इसे value दी जाती है इसके rounded ब्रैकेट का use किया जाता है

parameters optional होते हैं हम इसे चाहे तो use कर सकते हैं और अगर जरूरत नहीं है तो इसे use नहीं करे यह हमारी programming requirement पर depend करता है हमें प्रयोग करते समय type का ध्यान रखना है जैसे हमने int डिफाइन किया है

हमें order का ध्यान रखना चाहिए अगर हम parameter में कुछ different variables use करने होते हैं जोकि different datatype के हैं जैसे की int और string है हम दोनों को use कर सकते हैं अगर पहले int use किया है और दूसरे पर string use किया है तो value देते समय जो value पहले दी जाए वह int के लिए है और दूसरी value string के लिए है और इसके साथ ही number of parameters भी जरूरी है अगर दो parameter है तो value भी दो हो इस तरह हम सभी का ध्यान में रखना चाहिए

```

1 public class ParamExp {
2
3     int total;
4
5     public void sum(int x,int y)
6     {
7         total=x+y;
8         System.out.println("total is= "+total);
9     }
10
11 }
12
13 public static void main(String[] args) {
14     ParamExp p=new ParamExp();
15     p.sum(10,20);
16 }
17
18 }
19
20
21
22
23

```

Problems | Javadoc | Declaration | Console

<terminated> NameExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 6:20:59 PM)

total is= 40

हमने यहा पर एक class बनाई है फिर int total लिया है उसके बाद void sum एक method बनाया है इसमें हमने दो parameter x और y पास किए हैं इसमें जो value होती उन्हें जोड़कर total में assign

किया है और फिर total को print किया है उसके बाद call करने के लिए p object create किया है फिर p. के साथ method को call कर सकते है जिसमे 10 x को assign होगा और 20 y को assign होगा दोनों का total करके store करके display कर दिया जाएगा

```

1 public class ParamExp {
2
3     int total;
4
5     public void sum(int x,int y)
6     {
7         total=x+y;
8         System.out.println("total is= "+total);
9     }
10
11     public static void main(String[] args) {
12         ParamExp p=new ParamExp();
13         p.sum(10,20);
14     }
15 }

```

terminated> NameExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 6:20:59 PM)
total is= 40

हमने run किया हमारे पास output available है

```

1 public class ParamExp {
2
3     int total;
4
5     public void sum(int x,int y,String n)
6     {
7         total=x+y;
8         System.out.println("total is= "+total);
9     }
10
11     public static void main(String[] args) {
12         ParamExp p=new ParamExp();
13         p.sum(10,20);
14     }
15 }

```

terminated> ParamExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 6:33:47 PM)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The method sum(int, int, String) in the type ParamExp is not applicable for the arguments:
at ParamExp.main(ParamExp.java:17)

अगर हम parameter में string का use करते हैं तो हमारे पास error available हो जाएगा क्योंकि हमने नीचे इसे कोई value assign नहीं की है इस तरह अगर हम तीन या चार args बनाते हैं तो उसमें हमें सभी args को fill करना है

```

1 public class ParamExp {
2
3     int total;
4
5     public void sum(int x,int y,String n)
6     {
7         total=x+y;
8         System.out.println("total is= "+total+" name is "+n);
9     }
10
11     public static void main(String[] args) {
12         ParamExp p=new ParamExp();
13
14         p.sum(10,20,"Varinder");
15     }
16 }

```

Problems | Javadoc | Declaration | Console

```

<terminated> ParamExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 6:34:41 PM)
total is= 30 name is Varinder

```

हमने यहाँ पर आपने error को ठीक करके run किया तो हमारे पास output available है इसी तरह हमें order का ध्यान रखना है अगर हम string पहले देते हैं और int बाद में तो भी error available होगा इस तरह से इन सभी को ध्यान में रखते हुए हम parameters का use कर सकते हैं

Method overloading

- ❑ When we create more than one method with same name but having different parameters. Then it is called method overloading.
- ❑ Compiler differentiate b/w methods with their parameters.
- ❑ It is easier for user and programmer to program when he has to work with one name of method.

MethodOvrExample.java

```

public void showData()
{
    System.out.println(" Without any parameter");
}

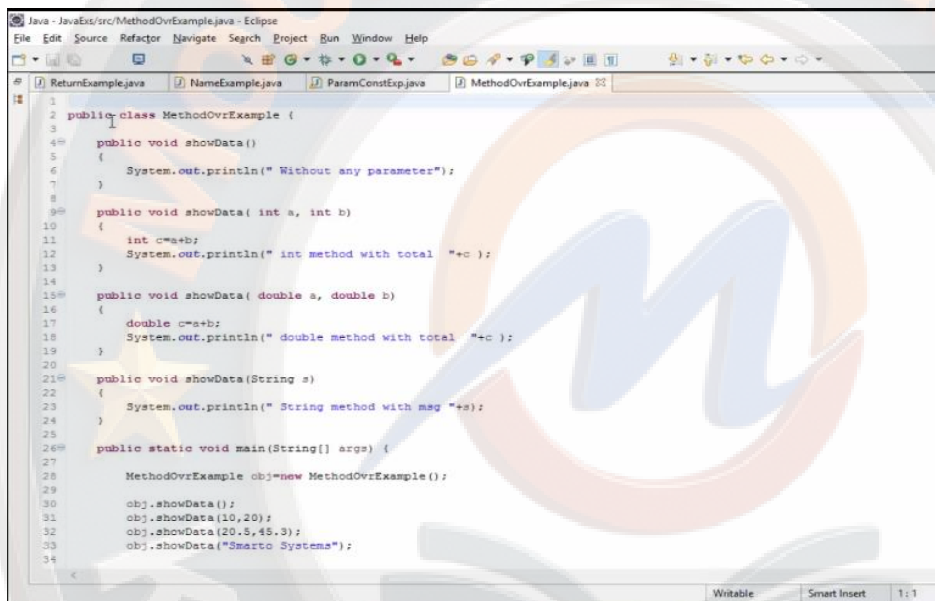
public void showData( int a, int b)
{
    int c=a+b;
    System.out.println(" int method with total "+c );
}

public void showData( double a, double b)
{
    double c=a+b;
    System.out.println(" double method with total "+c );
}

public void showData(String s)
{
    System.out.println(" String method with msg "+s);
}

```

अब हम method overloading के बारे में जानेंगे यह बहुत ही important रहती है हम यहा पर एक ही नाम से काफी सारे method create कर सकते हैं परन्तु parameters के आधार पर थोडा सा differentiate करते हुए use कर सकते हैं इसे java में method overloading कहते हैं जैसे कि हमने show data method create किया है हमने चार बार use किया है और हर बार parameters को अलग रखा है 1st में कोई भी parameter नहीं है 2nd में parameters के रूप में दो int variables हैं 3rd में parameters double के रूप में है और 4th में parameter string के रूप में use किया गया है इस तरह हम parameters में differentiate करके एक ही नाम से काफी सारे method बना सकते हैं compiler इन सभी parameters के बीच differentiate कर सकता है इस तरह user के लिए बहुत ही आसान रहता है



```

1 public class MethodOvrExample {
2
3     public void showData()
4     {
5         System.out.println(" Without any parameter");
6     }
7
8
9     public void showData( int a, int b)
10    {
11        int c=a+b;
12        System.out.println(" int method with total "+c );
13    }
14
15    public void showData( double a, double b)
16    {
17        double c=a+b;
18        System.out.println(" double method with total "+c );
19    }
20
21    public void showData(String s)
22    {
23        System.out.println(" String method with msg "+s);
24    }
25
26    public static void main(String[] args) {
27
28        MethodOvrExample obj=new MethodOvrExample();
29
30        obj.showData();
31        obj.showData(10,20);
32        obj.showData(20.5,45.3);
33        obj.showData("Smarto Systems");
34    }

```

हमने पहले class डिफाइन की है उसके बाद show data एक method डिफाइन किया है जिसमें कोई भी parameter नहीं है और फिर एक message print करवाया है फिर हमने same show data के नाम से method create किया है जिसमें दो variables पास किए हैं दोनों का total करके c को ट्रांसफर किया है फिर print किया है उसके बाद show data में दो double पास किए हैं इसी तरह 4th में string पास किया है हमने call करने के लिए object बनाया है फिर obj के साथ सभी को call किया है

```

1 public class MethodOvrExample {
2
3
4 public void showData()
5 {
6     System.out.println(" Without any parameter");
7 }
8
9 public void showData( int a, int b)
10 {
11     int c=a+b;
12     System.out.println(" int method with total "+c );
13 }
14
15 public void showData( double a, double b)
16 {
17     double c=a+b;
18     System.out.println(" double method with total "+c );
19 }
20
21 public void showData(String s)
22 {
23     System.out.println(" String method with msg "+s);
24 }
25

```

Problems | Javadoc | Declaration | Console

```

<terminated> MethodOvrExample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 6:45:08 PM)
Without any parameter
int method with total 30
double method with total 65.8
String method with msg Smarto Systems

```

हमने run किया और हमारे पास output available है इस तरह से हम एक नाम से अलग अलग method को create कर सकते है बस हमे parameters को differentiate करना है इस तरह हम इनका use कर सकते है

Constructor in java

- When we create obj in class , a constructor is called. Basically **constructor is a special method** which is called when an obj is created but having different feature as method.
 - It is used to initialize data members / variables or invoke important statement
 - It has same name as class name.
 - It has no return data type.
 - A constructor with no parameter is known as default constructor.
 - There is always a default constructor in your class , whether you define one or not, because Java provides a default constructor which initialize data members.
 - If we create own constructor , default constructor is not used
 - If you declare a constructor that accepts parameters is called Parameterized Constructor
 - If we still want to have an default constructor , we need to create and declare an empty constructor for the class.

```

public class ConstExp {
    public ConstExp ()
    {
        System.out.println("Hi I m Constructor");
    }

    public static void main(String[] args)
    {
        ConstExp obj=new ConstExp ();
    }
}

```

ParamConstExp.java

अब हम constructor method के बारे में जानेगे यह class का special method है जब भी हम class का object create करते है तो default रूप में constructor invoke हो जाता है तो automatically इसे call कर लिया जाता है method होने के साथ इसके कुछ feature है जो इसको ओर method से अलग बनाती है

इसका use data members या variables को initialize करने के लिए किया जाता है यानि की value प्रदान करने के लिए की जाती है

इसके अतिरिक्त कुछ important task है जिन्हें हमे बनाने की जरूरत ना हो वह automatically ही invoke हो जाए जैसे ही object create किया जाए वैसी statement हम constructor में डाल कर रखते है

इसका नाम class के नाम जैसा होता है इसकी return datatype नहीं होती है

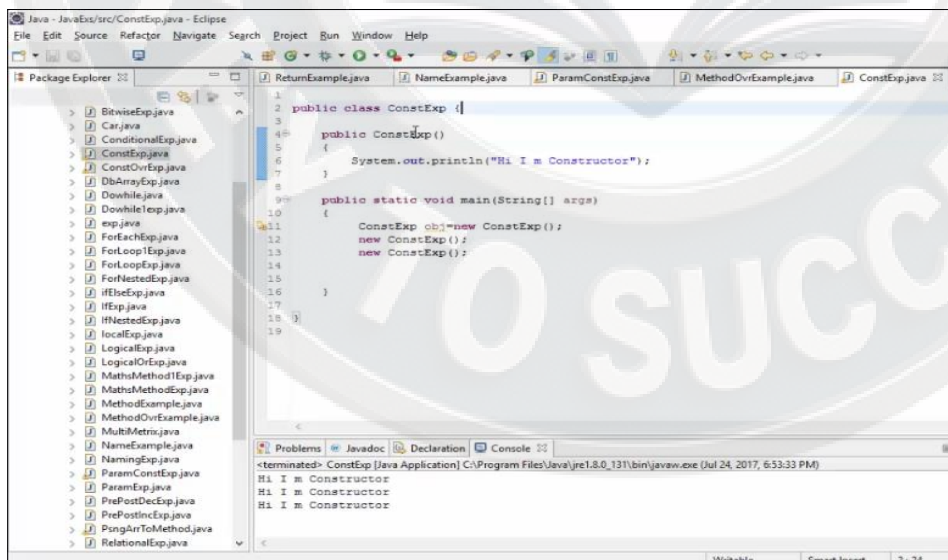
अगर हम constructor में कोई parameter नहीं देते है वह एक default constructor कहलाता है

जब भी हम कोई class create करते है तो एक default constructor पहले से ही available रहता है हम उसे डिफाइन करे या ना करे यह available रहता है

इसके अतिरिक्त हम आपने constructor भी create कर सकते है

अगर हम parameters base constructor create करते है तो वह parameterized constructor कहलाएगे इसका मतलब यदि हम constructor में कोई value पास करते है तो उसे parameterized constructor कहते है

अगर हम parameterized constructor और default constructor दोनों का use करते है तो हमे specifically empty constructor create करना होगा जोकि default constructor को represent करेगा



```

1 public class ConstExp {
2
3
4     public ConstExp()
5     {
6         System.out.println("Hi I m Constructor");
7     }
8
9     public static void main(String[] args)
10    {
11        ConstExp obj=new ConstExp();
12        new ConstExp();
13        new ConstExp();
14    }
15 }
16
17
18
19

```

Console Output:

```

<terminated>- ConstExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 6:53:33 PM)
Hi I m Constructor
Hi I m Constructor
Hi I m Constructor

```

हमने एक class create की है उसके बाद हमने एक special method भी create किया है क्योंकि इसका नाम class के नाम जैसा है हमने इसके साथ किसी return datatype का use नहीं किया है फिर

हमने एक line print करवाई है उसके बाद main method डिफाइन किया है इसमें हमने एक obj create किया है जब हम object create करते है तो यह constructor खुद ही invoke हो जाता है हमने run किया तो हमारे पास output available है हम एक से ज्यादा constructor create कर सकते है हमे उसके parameter में differentiate करना होता है इसका एक example देखते है

```

1 public class ParamConstExp {
2
3
4     public ParamConstExp()
5     {
6         System.out.println("Hi I m Constructor");
7     }
8     public ParamConstExp( String msg)
9     {
10        System.out.println("I m Parameter Constructor with msg " + msg);
11    }
12
13    public void ParamConstExp()
14    {
15        System.out.println("I m in Method with classname ");
16    }
17    public static void main(String[] args)
18    {
19        ParamConstExp obj=new ParamConstExp();
20        ParamConstExp obj1=new ParamConstExp("Smarto Systems");
21        obj.ParamConstExp();
22    }
23
24
25 }

```

Problems | Javadoc | Declaration | Console

```

<terminated> ParamConstExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 6:39:43 PM)
Hi I m Constructor
I m Parameter Constructor with msg Smarto Systems
I m in Method with classname

```

हमने एक class create की है उसके बाद हमने दो constructor डिफाइन किए है इसमें एक default constructor है जिसमे हमने कोई भी parameter पास नही किया है उसके बाद एक message print करवाया गया है उसके बाद दूसरा constructor जिसमे string को पास किया है जब यह invoke होगा तो

इसमें हम एक value दे सकते हैं जोकि string रहेगी फिर एक message print करवाया है फिर उसके बाद हमने class के नाम का एक method डिफाइन किया है इसमें method और constructor में अंतर होता है जब हम return datatype को introduce करेंगे class के नाम के साथ तो वह एक method माना जाएगा और जब return datatype use नहीं करते हैं तो वह constructor माना जाएगा फिर हमने इन्हें call किया है तो हमारे पास output available है इस तरह से हम constructor को create कर सकते हैं

Difference b/w constructor and method

Constructor	Method
Constructor has same name as the class name.	Method can have same as class name and can be different as per requirement.
Constructor is used to initialize the data members and startup tasks.	Method is used to define particular task for execution .
Constructor is automatically called when an obj is created.	We need to call method explicitly.
There is no return data type in constructor.	We must declare a return data type in methods.
There is always a default constructor provided by compiler.	There is no method provided by compiler.

यहा पर constructor और method के बीच के अंतर के बारे में बताया गया है इन दोनों में अंतर है जिस कारण यह एक दुसरे से अलग है

Constructor overloading

- When we create more than one constructors having different parameters. Then it is called constructor overloading.
- Compiler differentiate b/w constructors with their parameters.

```
public ConstOvrExp( String msg)
{
    System.out.println("I m in String Constructor with msg " + msg);
}
public ConstOvrExp( int a)
{
    System.out.println("I m in int Constructor with value " + a);
}
public ConstOvrExp( double a)
{
    System.out.println("I m in double Constructor with value " + a);
}
```

ConstOvrExp.java

अब हम constructor overloading के बारे में जानकारी लेंगे यदि हम आपनी class में एक से ज्यादा constructor को use करते हैं तो उसे constructor overloading कहते हैं इसमें नाम same ही रहता है क्योंकि एक constructor का नाम class name जैसा ही होता है तो हम एक ही नाम को बार बार use करते हैं परन्तु differentiate करने के लिए अलग अलग parameters का use करते हैं जैसे कि हम right box में देख सकते हैं compiler automatically constructor को parameters के आधार पर differentiate कर देता है जब हम कोई object बनाते हैं और जो value assign करते हैं object बनाते समय तो उस समय उस प्रकार के constructor को invoke किया जाता है जिस प्रकार की value parameter को assign की है इसको हम practical देखते हैं

```

1 public class ConstOvrExp {
2
3     public ConstOvrExp ()
4     {
5         System.out.println("Hi I m Constructor");
6     }
7
8     public ConstOvrExp( String msg)
9     {
10        System.out.println("I m in String Constructor with msg " + msg);
11    }
12
13    public ConstOvrExp( int a)
14    {
15        System.out.println("I m in int Constructor with value " + a);
16    }
17
18    public ConstOvrExp( double a)
19    {
20        System.out.println("I m in double Constructor with value " + a);
21    }
22
23    public void ConstOvrExp().
24    {
25        System.out.println("I m in Method with classname ");
26    }
27
28    public static void main(String[] args)
29    {
30        ConstOvrExp obj1=new ConstOvrExp();
31        ConstOvrExp obj2=new ConstOvrExp("Smarto Systems");
32        ConstOvrExp obj3=new ConstOvrExp(10);
33        ConstOvrExp obj4=new ConstOvrExp(14.23);
34        obj4.ConstOvrExp();
35    }
36 }

```

हमने यहां पर एक class create की है इसी नाम से हमने काफी सारे constructor create किए हैं और सभी में अलग अलग parameters पास किए हैं और फिर इन्हें call करने के लिए obj create करके call किया है फिर हमने इसे run किया है

```

11 System.out.println("I m in String Constructor with msg " + msg);
12 }
13 public ConstOvrExp( int a)
14 {
15     System.out.println("I m in int Constructor with value " + a);
16 }
17 public ConstOvrExp( double a)
18 {
19     System.out.println("I m in double Constructor with value " + a);
20 }
21 public void ConstOvrExp().
22 {
23     System.out.println("I m in Method with classname " );
24 }
25 public static void main(String[] args)
26 {

```

Console Output:

```

<terminated> ConstOvrExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 7:26:50 PM)
Hi I m Constructor
I m in String Constructor with msg Smarto Systems
I m in int Constructor with value 10
I m in double Constructor with value 14.23
I m in Method with classname

```

तो हमारे पास output available है इस तरह अगर हमारे पास एक से ज्यादा constructor रहते हैं और उनमें parameter में differentiate रहता है तो हम इस तरह से constructor overloading का use कर सकते हैं

Array in Java

- ❑ Array is a collection of same data type elements and having contiguous memory location.
- ❑ `int arr[]=new int[5];`
- ❑ Its indexing is start from 0th element.
- ❑ For sorting of data or code optimization array is best.
- ❑ You can create fixed size array and cannot change array size at runtime.
- ❑ You can store /access data in individual item using its index.
- ❑ You can create single and multidimensional array.

ArrayRequirement

0	1	2	3	4
10	20	30	40	50
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

ArrayExp.java

Array1Exp.java

अब हम array के बारे में जानेगे यहा हमारे पास array requirement का example available है

```

1 public class ArrayRequirement {
2
3
4
5
6 public static void main(String[] args) {
7     int a=10;
8     System.out.println("a ==>"+a);
9
10
11 }
12
13
14

```

हमारे पास practical available है यहा हमारे पास a 10 है जिसे print भी करवाया गया है और run करने पर 10 available है अगर हमे ओर variables बनाने है तो हमे b, c, etc use करने होंगे इस तरह अगर हमे 50 variables बनाने है तो यह थोडा typical हो जाएगा तो ऐसे case में एक ही datatype के काफी सारे variables बनाने हो तो हमारे पास array best option है यह हमारे पास example है कि हमे array की जरूरत किस लिए है अब हम array के बारे में जानते है

Array in Java

- ❑ Array is a collection of same data type elements and having contiguous memory location.
- ❑ `int arr[]=new int[5];`
- ❑ Its indexing is start from 0th element.
- ❑ For sorting of data or code optimization array is best.
- ❑ You can create fixed size array and cannot change array size at runtime.
- ❑ You can store /access data in individual item using its index.
- ❑ You can create single and multidimensional array.

ArrayRequirement

0	1	2	3	4
10	20	30	40	50
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

ArrayExp.java

Array1Exp.java

हम एक जैसे datatype के elements या variables एक साथ create कर सकते है इसी को हम array कहते है यह same datatype elements की एक collection है यह सारे elements एक contiguous memory location पर available रहते है जैसे कि right table में देखे 10 to 50 value store की है और जो location है वह contiguous location है एक साथ सभी जुड़े है हमे इसमें एक datatype डिफाइन करेगे array का नाम रखेगे और दो sequare ब्रैकेट्स का use करेगे यह array के size को

डिफाइन करेगी और = new int [5] इसमें हम 5 variables create करना चाहते हैं जिनका नाम array है जोकि arr है तो variables कुछ इस तरह से create हो जाएंगे सबसे पहले arr0 ,arr 1 ,2,3,4 है और फिर इनमे value store कर सकते हैं इनमे हमने 10, 20,30, 40, 50 store किया है इस सब के index 0 से है और 4 तक है इस तरह elements 5 है

यदि हम काफी सारे data store करना चाहते हैं तो array का use कर सकते हैं इसे आसानी से sort कर सकते हैं और यह code optimization के कारण भी यह important रहता है

array का fixed size रहता है

array के individual address को use कर सकते हैं

Two type of array

□ Single dimensional

```
int arr[] = new int[5];
```

0	1	2	3	4
10	20	30	40	50
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

ArrayExp.java

Array1Exp.java

□ Multi dimensional

```
int[][] arr = new int[3][3];
```

Multidimensional Array		
a[0][0]	a[0][1]	a[0][2]
a[1][0]	a[1][1]	a[1][2]
a[2][0]	a[2][1]	a[2][2]

Multidimensional Array		
1	2	3
4	5	6
7	8	9

java में हमारे पास array के दो types available है जोकि single dimensional और multi dimensional है

single dimensional का अर्थ है कि एक साथ एक ही index में हमारे पास सभी value का indexes का होना जैसे कि left side show किया जा रहा है यह एक साथ है और एक ही index में represent किए जा रहे हैं

multi dimensional में एक से ज्यादा indexes provide करवाते हैं जैसे कि हमने दो indexes use किए हैं यह कुछ इस तरह से कार्य करता है पहला index row के लिए और दूसरा index column के लिए काम करता है हमारा arr 0 से शुरू हो रहा है सबसे पहले value [0][0] होगी फिर row wise [0][1] है फिर [0] row में [2] column होगा इसी तरह जब line change होगी तो row 1 होगी इस तरह हट एक बार हर row के लिए जितने column हैं वह create किए जाएंगे और हम इनमें value store कर सकते हैं इस तरह हम इनका use करेंगे

Single dimensional array

```

1 public class ArrayExp {
2
3     public static void main(String args[])
4     {
5
6         int a[]={new int[5];
7         a[0]=10;
8         a[1]=20;
9         a[2]=30;
10        a[3]=40;
11        a[4]=50;
12
13
14        for(int i=0;i<a.length;i++)//length is the property of array
15        System.out.println("a["+i+"]= "+a[i]);
16
17    }
18 }

```

Console Output:

```

<terminated> ArrayExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 9:27:23 PM)
a[0]= 10
a[1]= 20
a[2]= 30
a[3]= 40

```

हमने यहाँ पर एक class डिफाइन की है फिर static void main में एक array create किया है जिसका नाम a रखा है जोकि square ब्रैकेट द्वारा show किया जा रहा है इसके बाद last ब्रैकेट में जो numeric 5 दिया है जो बताता है कि हमने 5 integer datatype के element create किए हैं जोकि एक साथ है जोकि a arr से connected है इन पाँचों में value assign कर सकते हैं जोकि हमने की है हम इनको individual print करवा सकते हैं और हमने print करवाने के लिए loop का use किया है उसके बाद

value को print करवाया है हमने run किया है हमारे पास output available है इस तरह से हम single dimensional array को create कर सकते है और use कर सकते है

multi dimensional array

```

3
4 public class DbArrayExp{
5
6 public static void main(String args[]){
7
8
9 int arr[][]={{1,2,3},{4,5,6},{7,8,9}};
10
11
12 for(int r=0;r<3;r++){
13 System.out.println();
14 for(int c=0;c<3;c++){
15 System.out.print(arr[r][c]+" ");
16 }
17
18
19
20
21 }
22
23 for(int r1=0;r1<3;r1++){
24 System.out.println();
25 for(int c1=0;c1<3;c1++){
26 System.out.print("arr["+r1+"["+c1+" ]");
27 }
28 }

```

हमने एक class create की है और void main में कुछ चीजे specify की है सबसे पहले हमने arr को डिफाइन किया है जिसमे हमने values को भी डिफाइन किया है हमने array को execute करने के लिए loop का use किया है 1 आउटर loop है जो rows को डिफाइन करती है जिसके लिए r का use किया है क्योंकि row 3 है इसलिए 3 का use किया है और r++ के साथ आगे बढ़ते जा रहे है एक ही statement है एक space create करने के लिए println को print करवाया है उसके बाद inner loop में column

को represent करती है जिसके लिए c का use किया है और फिर row और column variables का use करके सभी array की values को print करवाया है और जब run किया है तो हमारे पास output available हो गया है

```

1 public class DbArrayExp {
2
3
4 public static void main(String args[]) {
5
6 int arr[][] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
7
8
9 for (int r=0; r<3; r++) {
10 System.out.println();
11 for (int c=0; c<3; c++) {
12 System.out.print(arr[r][c] + " ");
13 }
14 }
15 }
16
17
18 }
19
20
21 }
22
23 for (int r1=0; r1<3; r1++) {
24 System.out.println();
25 for (int c1=0; c1<3; c1++) {
26 System.out.print("arr[" + r1 + "][" + c1 + "]: ");
27 }
28 }
29 }
30 }
31
32
33

```

Console Output:

```

1 2 3
4 5 6
7 8 9
arr[0][0] arr[0][1] arr[0][2]
arr[1][0] arr[1][1] arr[1][2]
arr[2][0] arr[2][1] arr[2][2]

```

यहा हमारे पास output available है लेकिन नीचे हमने एक loop ओर use की है

```

1
2
3
4 public class DbArrayExp {
5
6 public static void main(String args[]) {
7
8 int arr[][] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
9
10
11
12 for (int r=0; r<3; r++) {
13 System.out.println();
14 for (int c=0; c<3; c++) {
15 System.out.print(arr[r][c] + " ");
16 }
17 }
18 }
19
20
21 }
22
23 for (int r1=0; r1<3; r1++) {
24 System.out.println();
25 for (int c1=0; c1<3; c1++) {
26 System.out.print("arr[" + r1 + "][" + c1 + "]: ");
27 }
28 }
29 }
30 }
31
32
33

```

Console Output:

```

1 2 3
4 5 6
7 8 9
arr[0][0] arr[0][1] arr[0][2]
arr[1][0] arr[1][1] arr[1][2]
arr[2][0] arr[2][1] arr[2][2]

```

हमने run किया तो हमारे पास output available है यह सिर्फ समझने के लिए है कि यह किस तरह से काम करता है इस तरह हम double dimensional array का use कर सकते हैं

Multiplication of 2 Metrix

Multidimensional Array				Multidimensional Array				Multidimensional Array		
1	2	3	X	9	8	7	=	9	16	21
4	5	6		6	5	4		24	25	24
7	8	9		3	2	1		21	16	9

Here is program to multiply 2 Metrix of and storing result in 3rd Metrix

MultiMetrix.java

हम यहा पर multiplication array के बारे में जानेगे हमने यहा पर two dimensional array का use किया है हमने यहा दो metrix ली है हमने two dimensional array को लिया है जिसमे कुछ values को store किया है और दूसरे में value को reverse करके store किया है लेकिन हम दोनों के values को element by element जोड़ना चाहते है जैसे कि multi dimensionals के [0][0] को multiply करके तीसरे array multi dimensional array में result store किया जाए इसी तरह से दूसरे element की value को लिया जाए और multiply करके result store किया जाए इसी तरह सभी values के साथ किया जाए

```

1 public class MultiMatrix{
2
3
4     public static void main(String args[]){
5
6         int arr[][]={{1,2,3},{4,5,6},{7,8,9}};
7         int brr[][]={{9,8,7},{6,5,4},{3,2,1}};
8
9
10        int crr[][]=new int[3][3];
11
12
13        for(int r=0;r<3;r++){
14            System.out.println();
15            for(int c=0;c<3;c++){
16                {
17                    crr[r][c]=arr[r][c]*brr[r][c];
18                    System.out.print(crr[r][c]+"
19                }
20            }
21        }
22
23        for(int r1=0;r1<3;r1++){
24            System.out.println();
25            for(int c1=0;c1<3;c1++){
26                System.out.print(crr[r1][c1]+"*c1+" );
27            }
28        }
29    }
30 }
31 }

```

```

<terminated> MultiMatrix [Java Application] C:\Proje
16      21
24      25      9
21      16      9
crr[0][0] crr[0][1] crr[0][2]
crr[1][0] crr[1][1] crr[1][2]
crr[2][0] crr[2][1] crr[2][2]

```

हमने यहां पर multi dimensional array के example को practical किया है और हमारे पास output right side पर available है जिसमें की दोनों indexes value को multiply करके show किया गया है इस तरह double dimensional का use करके उनका output तीसरे में store करवाया है हम plus, minus का use भी कर सकते हैं इस तरह से अलग अलग example का use कर सकते हैं इस तरह हम multi matrix का use कर सकते हैं

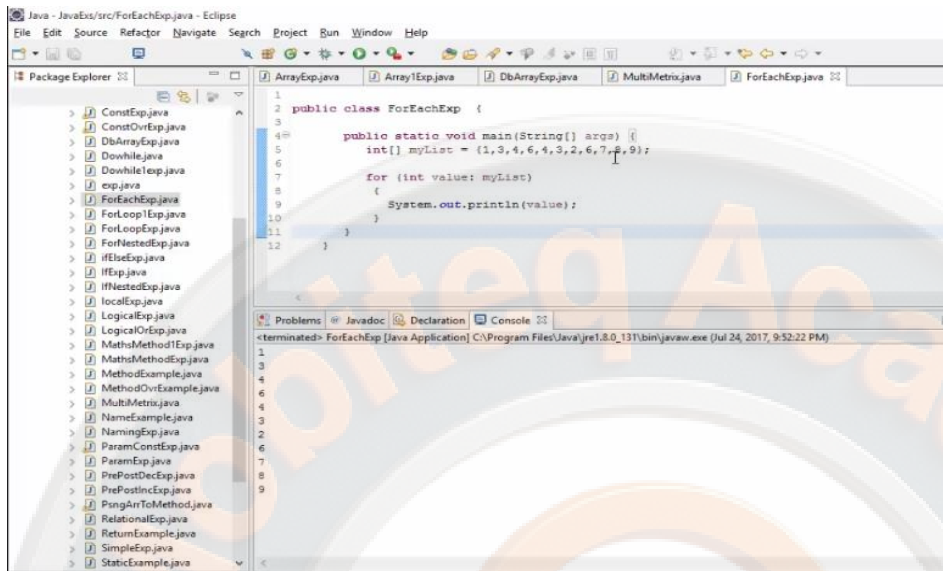
foreach Loop in Java

foreach loop is very beneficial loop to go through each element of any array, without knowing the total number of elements.

```
int[] myList = {1,3,4,6,4,3,2,6,7,8,9};  
  
for (int value: myList)  
{  
    System.out.println(value);  
}
```

ForEachExp.java

अब हम एक अलग तरह के loop का use करेगे जोकि मुख्यतः array के साथ use किया जाता है और यह बहुत ही important रहता है foreach loop बहुत ही beneficial loop है जोकि हर एक index की value को directly access करता है और एक variable में store करता है और हम directly value को print करवा सकते है example के लिए हमने एक array myList लिया है जिसमे हमने काफी सारे element store किए है इसमें फिर हमने for का use किया है फिर दो rounded ब्रैकेट का use किया है इसमें सबसे पहले एक variable int declare करवाए जिस तरह का ऊपर datatype लिया है उस तरह का यहा भी ले हमने int value लिया है value में हमने double column के बाद array का नाम use किया है जिस array की values को pick करना चाहते है अब यह variables की value को store करेगा और फिर इसे print करवाते जाएगे



```
1 public class ForEachExp {
2
3
4 public static void main(String[] args) {
5     int[] myList = {1,3,4,6,4,3,2,6,7,8,9};
6
7     for (int value: myList)
8     {
9         System.out.println(value);
10    }
11
12 }
```

terminated: ForEachExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 9:52:22 PM)

1
3
4
6
4
3
2
6
7
8
9

हमने यहां पर foreach loop के example को practical किया है और run करने के बाद output available है जिसमें value को print करवा दिया गया है इस तरह से foreach loop का use कर सकते हैं

Passing Array to Methods

- Just like we can send simple variable to methods, in same manner we can send array to methods.

```
public static void displayArray(int[] arr)
```

```
displayArray(brr);
```

```
PsnGArrToMethod.java
```

अब हम जानेगे कि किस प्रकार हम आपने method में array को as a parameter पास कर सकते हैं यहा पर हमने array को as a parameter पास किया है जब हम इस method को array के नाम से call करेगे जोकि display array है void main में, वहा पर हम कोई दूसरा array देंगे उसकी जो भी value है इस array में चली जाएगी और हम ऊपर वाले arr को print करवा सकते हैं

```

1 public class PngArrToMethod {
2
3
4
5     public static void displayArray(int[] arr)
6     {
7         for (int i = 0; i < arr.length; i++) {
8             System.out.print("value of Array:=" + arr[i] + " ");
9         }
10    }
11
12
13
14    public static void main(String[] args) {
15
16        int brr[]={1,2,3,4,5};
17
18        displayArray(brr);
19
20    }
21
22
23
24

```

Problems @ Javadoc Declaration Console

<terminated> PngArrToMethod [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 24, 2017, 10:03:47 PM)

value of Array:=1 value of Array:=2 value of Array:=3 value of Array:=4 value of Array:=5

हमने यहा पर एक class बनाई है फिर static void method में एक method create किया है जोकि display array है इसमें एक arr को parameter की तरह use किया गया है अब हम चाहते हैं कि अगर इस arr में कोई value आती है यह value हम तब देंगे जब हम इस array को call करेगे कोई value आती है तो $i = 0$ है arr की length को calculate करते हैं और आगे की ओर बढ़ते हुए i है जो नीचे value of array को represent करता है और print करवाता है जब हम element के बारे में ना पता हो कि कितने element आने वाले है तो हम arr.length का use करते हैं फिर हमारी loop खत्म है अब हम जब static void main method में जाते हैं तो हमने एक arr brr के नाम से create किया है और इसमें 5 element दे दिए हैं हमने static का use किया है इसीलिए हम direct call कर सकते हैं हमने call किया है और array में brr को पास कर दिया है हमने run किया है और हमारे पास output available हो गया है इस तरह हम किसी method में array को पास कर सकते हैं और उसकी value को access कर सकते हैं और साथ में array.length का use भी करना सीखा है

Primitive data types and Wrapper classes

Primitive data types

- We have already used primitive data types like int, char, float etc. in many examples in recent sessions.

Wrapper classes

- You can use **objects** to store same type of **data** and here wrapper classes come into the role.
- Wrapper classes are also important because they have their **own predefined functions** which provide important functionalities to our data.
- Wrapper classes are not primitive data type classes themselves as they just hold the value for primitive data types in objects.
- Whenever we **require precision** in data , we must use wrapper classes.

अब हम wrapper classes के बारे में जानेगे हमने अभी तक primitive datatype use की है



Data types and Wrapper classes

Primitive	Wrapper Class
boolean	Boolean
byte	Byte
short	Short
char	Character
int	Integer
float	Float
double	Double
long	Long

- In simple primitive data type we use
 - ▣ `int var=40;`
- But in wrapper classes
 - ▣ `Integer obj=new Integer(40);`
- But above both do the same thing.

String is not a primitive variable and it is a complex data type and we create objects of String class.

यहा पर primitive datatypes भी available है wrapper classes भी कुछ कुछ primitive classes की तरह है यहा पर 1st letter capital है बाकि नाम same है और int का पूरा नाम integer और char का पूरा नाम charecter use किया गया है जब हम primitive से data को store करते है तो वह as a variable store होता है लेकिन जब wrapper classes के साथ उसके दिए गए method के साथ data store करते है तो यह object के रूप में store होता है यह दोनों में मुख्य अंतर है जैसे की हम right side देख सकते है जैसे कि primitive datatype तो simple है जिसमे हम data store करते है लेकिन हम wrapper classes में हमने integer object declare करवाया है new integer जिसमे हमने constructor 40 पास किया है इस प्रकार हमने एक object interduse किया है दोनों के तरीके अलग अलग है लेकिन काम same है दोनों में 40 store किया जा रहा है तो हम wrapper classes का use क्यों करे

Primitive data types and Wrapper classes

Primitive data types

- We have already used primitive data types like int, char, float etc. in many examples in recent sessions.

Wrapper classes

- You can use **objects** to store same type **of data** and here wrapper classes come into the role.
- Wrapper classes are also important because they have their **own predefined functions** which provide important functionalities to our data.
- Wrapper classes are not primitive data type classes themselves as they just hold the value for primitive data types in objects.
- Whenever we **require precision** in data , we must use wrapper classes.

wrapper classes object में data store करती है जो data हम store करेंगे वह object बन जाएगा

wrapper classes में बहुत important और predefined function देती है जिनसे हम data पर work कर सकते हैं

wrapper classes कोई primitive classes नहीं है यह primitive data को रखने का एक तरीका है यह केवल value को object के रूप में store करती है

इसी तरह अगर हमें precision required हो अर्थात् बैंक में किसी तरह की कैलकुलेशन है जहाँ पर हम बहुत ही सही calculation करनी है काफी सारे दशमलव तक हमें value सही चाहिए तो हम wrapper classes का use कर सकते हैं क्योंकि वहाँ पर बहुत ज्यादा effective रहती है इन सभी को हम practical देखते हैं

```

1 public class PrimitiveExp {
2
3     public static void main(String[] args) {
4
5         double d=197.45d;
6
7         System.out.print("value of double d :=" + d);
8
9     }
10
11
12 }
13

```

Console Output:
<terminated> PrimitiveExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 9:09:26 AM)
value of double d :=197.45

यहा पर हमारे पास primitive classes का example available है जिसमे हमने एक class create की है और इसमें एक d variable create किया है जोकि double datatype का है और इसमें एक value store की है और value को print करवा दिया है हम run करे तो हमारे पास normal output available है ऐसे example हमने पहले भी बहुत से देखे है यह केवल दोनों में तुलना के लिए show किया गया है

wrapper classes

```

1 public class WrapperExp {
2
3
4     public static void main(String[] args) {
5         double d=197.45d;
6         System.out.println("value of double d :=" + d);
7         Double obj=new Double(d);
8         System.out.println("value of double Double obj :=" + obj);
9         byte b=obj.byteValue();
10        System.out.println("value of double to byte :=" + b);
11        int i=obj.intValue();
12        System.out.println("value of double to int :=" + i);
13        String s=obj.toString();
14        System.out.println("value of double to String :=" + s);
15
16    }
17
18 }
19

```

```

<terminated> WrapperExp [Java Application] C:\Program
value of double d :=197.45
value of double Double obj :=197.45
value of double to byte :=-59
value of double to int :=197
value of double to String :=197.45

```

हमने यहा पर wrapper classes के example को लिया है हमने एक class create की है फिर static void main में आकर d variable में एक value store की है और फिर print करवा लिया है जोकि output में print भी हुई है फिर हमने उसको d value को एक object में convert कर लिया है हमने इसे object में convert एक double wrapper class के द्वारा किया है और इसे भी print किया है जोकि output में print हुआ है इसके बाद हमने एक conversion शुरू किया है हमने obj के साथ byte value method लगाकर object में store value को byte के b variable में store करवा लिया है और उसे print करवाया है जैसे की byte -128 से 127 तक value store होती है तो 197 value बड़ी है तो value -59 में store हुई है इसी तरह हमने int में value को store करवाया है जोकि i में store है और output में 197 है इसमें .45 store नहीं किया गया है क्योंकि int factorional value को store नहीं करता है फिर हमने इसी object value को string में convert किया है इसके साथ हमने तो string method का use किया है जोकि output में available हो गई है इस तरह हम इसका use कर सकते है

AutoBoxing and Unboxing in wrapper classes

□ **autoboxing and unboxing** feature automatically converts

□ primitive into object

- `int x=10;`
- `Integer obj=Integer.valueOf(x);`
- `Integer obj=x;`

□ object into primitive

- `Integer obj=new Integer(10);`
- `int x=obj.intValue();`
- `int y=obj;`

`BoxUnboxExp.java`

अब हम wrapper classes के दो important concept कर बारे जानेगे जोकि auto boxing और unboxing है auto boxing जब हम primitive को किसी object में convert करे तो वह process auto boxing कहलाती है जैसे कि हमने एक example दिखाया है जिसमे int x में 10 store किया गया है इसमें x एक primitive datatype का variable है लेकिन जब हम इसे object में convert करना है तो हम इसके auto boxing का use करेगे इसके लिए integer जोकि wrapper class है उसका use करना है उसके साथ जिस value को हमने object में convert करना है उसे यहा पर पास करना है जैसे कि हमने x पास किया है अब हमारे पास integer wrapper class का एक method value of (x) है यह x की value को object में convert करेगा और उसे store करने के लिए हमने पहले से integer obj create किया हुआ है इस तरह हम object में convert कर सकते है इसके लिए हम direct integer obj=x का use भी कर सकते है

इसी तरह से unboxing का अर्थ है इससे हम object को primitive datatype में convert कर सकते है इसके लिए हमने यहा पर example show किया है कि किस तरह से हम object को primitive में convert कर सकते है इनमे से किसी भी तरीका का use कर सकते है

```

1 public class BoxUnboxExp {
2
3
4     public static void main(String[] args) {
5
6         int x=10;
7         Integer obj=Integer.valueOf(x);
8         System.out.println("value of Integer := " + obj);
9
10
11         Integer obj1=x;
12         System.out.println("value of Integer := " + obj1);
13
14
15         Integer obj2=new Integer(10);
16         System.out.println("value of Integer obj2 := " + obj2);
17         int x1=obj2.intValue();
18         System.out.println("value of int x1 := " + x1);
19         int y1=obj2;
20         System.out.println("value of int y1 := " + y1);
21
22     }
23
24
25
26
27
28
29

```

हमने यहा पर auto boxing का example लिया है हमने यहा पर boxunbox exp एक class बनाई है उसके बाद static void main method में एक variable int x create किया है फिर हमने auto boxing process का use किया है अर्थात हमने x जोकि primitive datatype की value को एक object में convert करेगा और फिर उस object को print करवा दिया है

```

<terminated> BoxUnboxExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 9:39:44 AM)
value of Integer :=10
value of Integer :=10
value of Integer obj2 :=10
value of int x1 :=10
value of int y1 :=10

```

हमने program को run किया तो हमारे पास output में 10 प्राप्त हुआ है लेकिन जब हमने इसे object में convert किया तो भी इसकी value 10 ही रहेगी इसी तरह हमने आपने program में directly auto boxing process को किया है जिसमे x की value को directly int class के एक ओर object obj1 में ट्रांसफर कर दिया गया है और print किया है यह process auto boxing का था

```

7 Integer obj3=Integer.valueOf(x);
8 System.out.println("value of Integer :="+ obj3);
9
10
11 Integer obj1=x;
12
13 System.out.println("value of Integer :="+ obj1);
14
15
16
17 Integer obj2=new Integer(10);
18
19 System.out.println("value of Integer obj2 :="+ obj2);
20 int x1=obj2.intValue();
21
22 System.out.println("value of int x1 :="+ x1);
23 int y1=obj2;
24
25 System.out.println("value of int y1 :="+ y1);
26
27
28 }
29

```

Console Output:

```

<terminated> BoxUnboxExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 9:39:44 AM)
value of Integer :=10
value of Integer obj3 :=10
value of Integer obj2 :=10
value of int x1 :=10
value of int y1 :=10

```

इसमें हमारे पास पहले से ही object obj 2 के रूप में available है और इसमें 10 value store है अब हम इस value को primitive में convert करना चाहते हैं तो इसके लिए हम क्योंकि हमने int class use की है तो इसका intValue method available है इसका use करेगे इसी के साथ जो obj2 की value जो 10 है उसे x में store कर देगा तो हमें primitive value मिल जाएगी इसी को हम directly भी कर सकते हैं और इसे print करवा लिया है और run करने पर output available है इस तरह यह दोनों बहुत ही important concept है जिनका use करना होता है इस तरह से हम इनका use कर सकते हैं

Strings in Java

- ❑ Strings are sequence of characters.
- ❑ Strings are very important to save textual information.
- ❑ String class is used to manipulate string values and make easy to work with strings.

```
char[] arr = { 'S', 'm', 'a', 'r', 't', 'o' };
String s = new String(arr);
System.out.println("Value Of String is "+ s );
```

अब हम java में strings के बारे में जानेगे string एक complex datatype है और objects में data store करता है यदि हम काफी सारे charecters को एक sequence में इकट्ठा कर दे तो वह strings कहलाता है string बहुत ही important रहते है जब हम textual information store करना चाहते है जैसे की किसी address को, किसी name को इस तरह की information के लिए strings का use करते है strings class पहले से java में available है जोकि बहुत सारे functions, methods देती है जिनसे हम आसानी से कार्य कर सकते है इसे समझाने के लिए हमने नीचे एक example लिया है जिसमे हमने char arr एक array लिया है इसमें हमने smarto word को store करवाया है लेकिन charecter एक time पर एक ही charecter store कर सकता है इसीलिए हमने charecter को इस तरह से store किया है हमने हर एक charecter को individually store किया है हमने single quotes का use किया है अब हम इस array को string के object में convert करना है तो इसके लिए हमे string s type करना है s एक object है और फिर new string class को initialize करेगे और arr को constructor में पास करेगे तो हमारे पास s string का object बनकर वापस आएगा और फिर value को print करवाया गया है इस तरह से simple charecter को string में convert कर सकते है

```
String s = "Smarto";
System.out.println("Value Of String is "+ s );
```

हम कुछ इस तरह से भी use कर सकते है जिसमे हमे array की जरूरत नही होगी

```

1 public class StringsExp {
2
3
4     public static void main(String[] args) {
5         char[] arr = { 'S', 'm', 'a', 'r', 't', 'o' };
6         String s = new String(arr);
7         System.out.println("Value Of String is "+ s );
8     }
9
10
11
12
13
14

```

Console Output: Value Of String is Smarto

यहा पर हमारे पास string का example available है इसमें हमने एक class create की है फिर static void main में जाकर एक arr declare किया है जोकि char datatype का है इसमें हमने अलग अलग character को store किया है और एक array create किया है जोकि 6 element के साथ हमारे पास available है इस string में convert करने के लिए string class का use किया है और s variable लिया है फिर इसे string object में convert किया है और इसे print करवा लिया है और run करने के बाद output available है

```

1 public class Strings1Exp {
2
3
4
5
6     public static void main(String[] args) {
7         String s = "Smarto";
8         System.out.println("Value Of String is "+ s );
9     }
10
11
12
13

```

Console Output: Value Of String is Smarto

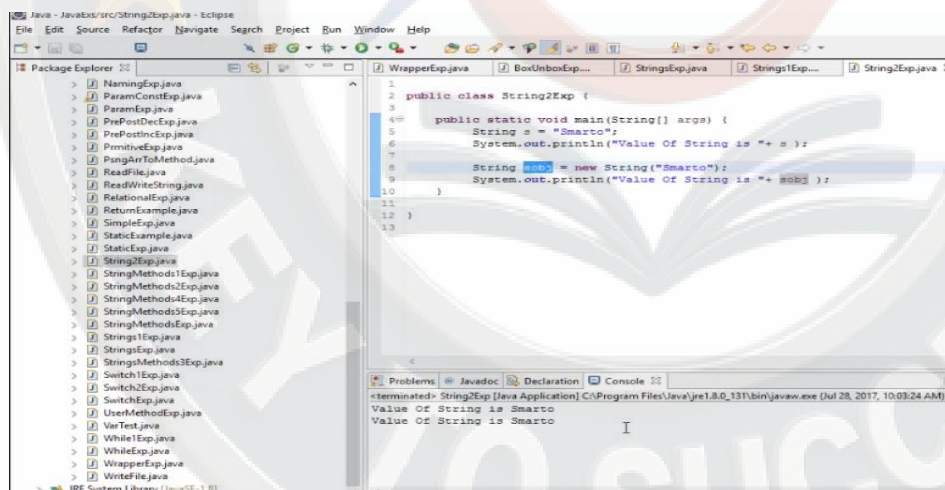
यहा पर हमने directly double quotes में text को type किया और string के object में store करवा लिया है और print करवा लिया है और run करने के बाद output भी available है इस तरह से strings को create कर सकते है इस तरह से textual informatio को store कर सकते है

Strings are Objects in java

- We have **String class** in java to manipulate strings.
- Strings variables are always treated as objects In Java.

```
String s = "Smarto";
System.out.println("Value Of String is "+ s );
String sobj = new String("Smarto");
System.out.println("Value Of String is "+ sobj );
```

अब हम string class के बारे में जानेगे यह पहले से predefined class है जब हम कोई variable declare करते है तो वह complex object के रूप में store किया जाता है जैसे कि हमने string s में smarto को store किया है परन्तु जब हम इसको as a object store करना चाहते है तो इस case में class का एक object बनाए string sobj और इसमें new string class को initialize करे और ब्रैकेट में double quotes में value या variable को पास करे इससे हमारे पास sobj एक object create हो गया है जोकि string class का है इसको हम normally print करवा सकते है



हमने यहा पर string class के example को practical किया है और run करने के बाद output available है इस तरह से किसी भी तरीके से का use कर सकते है दोनों में output same ही रहता है

String class has a lot of methods

- You can use plenty of predefined methods with string class to manipulate strings
 - ▣ **length() : to get the length in characters**
 - ▣ **concat() : to concatenate two or more strings**
 - ▣ **equals(): to compare strings**
 - Don't user ever to == equal operator to check equality in strings
 - ▣ **equalsIgnoreCase() to compare with ignore case values of strings :**

<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

string predefined methods available है जिनका use करके string की manipulation कर सकते है यहा पर हमारे पास कुछ important method available है इसमें नीचे जो URL available है इसमें strings के काफी सारे example available है जिनका use करके देख सकते है

length method

```

1 public class StringMethodsExp {
2
3
4     public static void main(String[] args) {
5
6
7         String str= new String("Smarto Systems");
8
9
10        int x=str.length();
11        System.out.println("Length is "+ x );
12        System.out.println("Length is "+ str.length() );
13
14
15        String str1 = "Our Website is ";
16        String str2 = "www.SmartoSystems.in";
17        String str3 = str1.concat(str2);
18
19        System.out.println(str3);
20
21    }
22
23 }

```

<terminated> StringMethodsExp [Java Application] C:\Program Files\Java\jre8.0_131\bin\javaw.exe (Jul 28, 2017, 11:21)

 Length is 14

 Length is 14

 Our Website is www.SmartoSystems.in

हमने यहा पर length method का example लिया है हमने यहा पर एक class create की है हमने str के नाम से एक variable create किया है यह एक object है इसमें हमने smarto system को assign किया है यदि हम इसकी length जानना चाहते है तो हम string class का method use करेगे हमने str के साथ dot लगा कर length method को use किया है यह integer value वापस करता है जैसा कि हम output में देख सकते है कि 14 available है इसमें smarto के 6 word और systems के 7 word है और इन दोनों के बीच में space available है इस कारण length 14 है हमने program में x को print करवाया है जोकि 14 है अगर हम space बढ़ा दे तो run करे तो हमारे पास length अलग available होगी अगर हम चाहे तो direct length को calculate कर सकते है इस तरह से length method का use कर सकते है

concatenate method

```

13
14
15     String str1 = "Our Website is ";
16     String str2 = "www.Smartsystems.in";
17     String str3 = str1.concat(str2);
18
19     System.out.println(str3);
20
21 }
22
23 }

```

Problems | Javadoc | Declaration | Console

```

<terminated> StringMethodsExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 11:22:56 AM)
Length is 17
Length is 17
Our Website is www.Smartsystems.in

```

इसके बाद हमारे पास 2nd concatenate method available है जोकि concat name से available है इसका अर्थ है कि हम एक से अधिक strings को आपस में जोड़ सकते है इसमें concat का use करना है यदि हम ऊपर length method देखे तो इसमें हमने plus का use किया है जोकि concatenate करने के लिए use किया है अब हम आपने concat के example को देखते है यहा पर हमने string class का str1 एक object बनाया है जिसमे हमने कुछ store किया है उसके str2 बनाया है उसमे भी हमने store किया है और फिर str3 बनाया है जिसमे हमने str1 को concat करके जो str2 आया है उसे str3 में store किया है इसमें हमने दोनों strings को जोड़ा है अगर हम इसे print करवाए तो हम output में दोनों को इकट्ठा देख सकते है इस तरह concat method को use कर सकते है

equals method

```

1 public class StringMethods2Exp {
2
3
4     public static void main(String[] args) {
5         String str1="Smarto Systems";
6         String str2="Smarto Systems";
7
8         if(str1.equals(str2))
9             System.out.println("Equal" );
10        else
11            System.out.println("Not Equal" );
12
13    }
14
15 }
16

```

Problems | Javadoc | Declaration | Console

```

<terminated> StringMethods1Exp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 11:27:0)
Equal
str3 Smarto Systems
str4 Smarto Systems
Not Equal

```

अगर string में equality check करनी है तो .equal method का use करना चाहिए इसमें अगर दोनों का output same है तो equal आएगा और अगर output same नहीं है तो not equal आएगा इस तरह से equality को check कर सकते हैं इस तरह से इसका use कर सकते हैं

ignore case method

```

1 public class StringsMethods3Exp {
2     }
3
4 public static void main(String[] args) {
5     String str1="SmartO Systems";
6     String str2="SmartO SystemS";
7
8     if(str1.equals(str2))
9         System.out.println("Equal" );
10    else
11        System.out.println("Not Equal" );
12
13    if(str1.equalsIgnoreCase(str2))
14        System.out.println("Equal" );
15    else
16        System.out.println("Not Equal" );
17
18
19 }
20
21
22

```

Problems | Javadoc | Declaration | Console

<terminated> StringsMethods3Exp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 11:34:2)

Equal

हमने यहा पर एक class create की है हमने यहा पर दो string str1 और str2 का use किया है और smarto systems store किया है अभी हमने दो strings को compare किया था जिसमे हमने .equal use किया था जिसमे output equal और not equal available था इसी को थोडा सा आगे बढ़ाते हुए अगर हम चाहते हैं किसी कारण से output ऐसा रहता है जिसमे कोई word बड़ा और कोई छोटा होता है जैसे की हमने ऊपर smartO में str2 में o capital है और इसी तरह str2 systemS में s capital है लेकिन str1 में सभी small case हैं तो ऐसे case में दोनों को compare करे और run करे तो हमारे पास output में 1 not equal रहेगा क्योंकि str2 में दो word capital है तो ऐसे case में हम एक और method ignore case का use कर सकते हैं अगर दो strings में किसी कारण से letter casing में का फर्क है तो यह उसे ignore करेगा और सही output available होगा जैसा की run करने के बाद equal output available है इस तरह से इनका use कर सकते हैं

Date class in java

- Date is a complex object in java , we have Date class in java to work with dates.
- We can work with date and time
 - ▣ To get current date and time we can use **Date** class.
 - ▣ We can compare date with DateFormat, Date class and compareTo() method.
 - ▣ **GregorianCalendar class** is used to add or subtract date and time.

DateExp.java

अब हम date class के बारे में जानेगे date class एक complex object के रूप में available है यह हमे काफी सारे methods provide करते है जिनसे हम dates पर कार्य कर सकते है हम इससे 1st तो यह जान सकते है कि current date क्या है तो हम date class का use करेगे

यदि हम dates को compare करना चाहते है तो date format , date class और compare to method का use कर सकते है

इसके इलावा हमारे पास Gregorian calendar class अलग available है जिससे हम date/time को subtract कर सकते है और add भी कर सकते है

How to display date and time in java

```
Date date = new Date();  
System.out.println(date.toString());
```

```
□ Output is  
  □ Thu Jul 27 13:04:59 IST 2017
```

DateExp.java

हमने यहा पर date class का एक example लिया है इससे हम date और time print करवा सकते है और यह हमारे system के according current date और time represent करेगा इसमें हम सबसे पहले Date class का एक object create करते है जैसे हमने date =new date create किया है इसको हम directly print नही करवा सकते है इसको हम पहले string में convert करते है हमने date.toString का use करके string में convert किया है तो अब as a string represent होगी हमारे पास output भी show किया गया है

```

1 import java.util.Date;
2
3 public class DateExp {
4
5     public static void main(String[] args) {
6
7         Date date = new Date();
8         System.out.println(date.toString());
9     }
10
11 }
12

```

Problems | Javadoc | Declaration | Console

<terminated> DateExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 12:18:25 PM)

हमने यहा पर सबसे पहले एक package include किया है अगर इसे include नही करे तो error available होगा package एक तरह collection of classes से है इस package में date class के method के बारे में बताया गया है उसके बाद हमने पहले Date class का एक object create किया है जैसे हमने date =new date create किया है हमने date.toString का use करके string में convert किया है हमने run किया और output में current date और time available है इस तरह से date को show कर सकते है

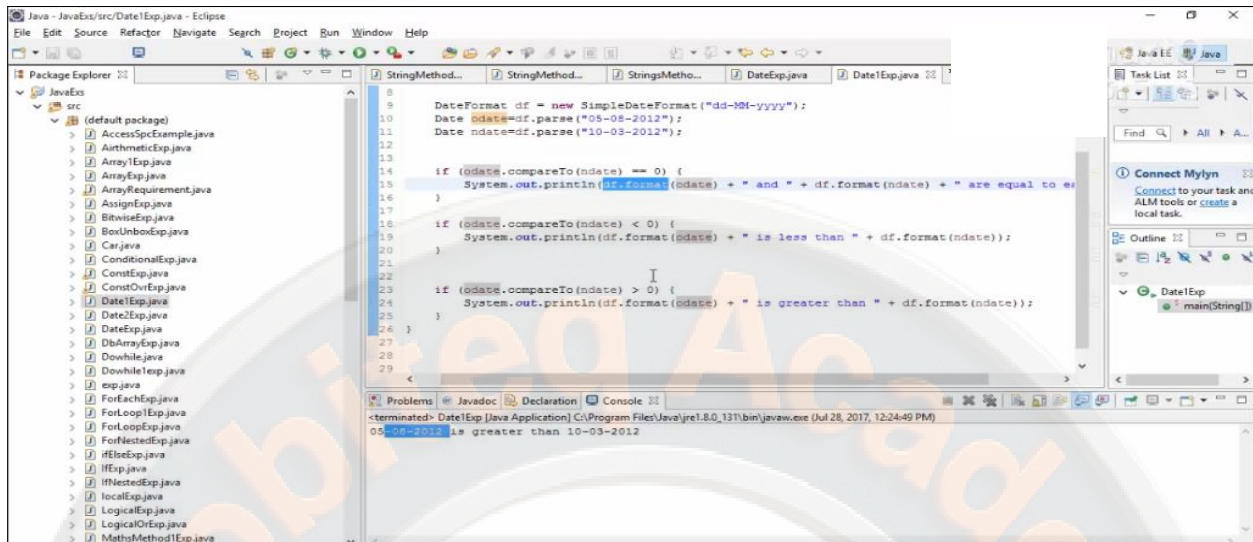
```

1  import java.text.*;
2
3  public class DateExp {
4      public static void main(String args[]) throws ParseException {
5
6
7
8
9          DateFormat df = new SimpleDateFormat("dd-MM-yyyy");
10         Date odate=df.parse("05-08-2012");
11         Date ndate=df.parse("10-03-2012");
12
13
14         if (odate.compareTo(ndate) == 0) {
15             System.out.println(df.format(odate) + " and " + df.format(ndate) + " are equal to each other");
16         }
17
18         if (odate.compareTo(ndate) < 0) {
19             System.out.println(df.format(odate) + " is less than " + df.format(ndate));
20         }
21
22         if (odate.compareTo(ndate) > 0) {
23             System.out.println(df.format(odate) + " is greater than " + df.format(ndate));
24         }
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

अब dates को compare करना सीखेंगे हमने यहा पर package include किया है फिर एक class create की है हमने static void main में exception को throw किया है इससे अगर कोई error expected रहती है किसी particular code से तो उस case में हम function के साथ particular exception को throw कर सकते है अगर कोई error हो तो इससे ठीक किया जा सकता है फिर हमने Date format एक class ली है जिसमे df variable का नाम रखा है date format से हम date का format decide कर सकते है जैसे हमने format set किया है इसके बाद हमने date class के दो object odate और ndate create किए है इनमे दो different date को store किया है store करने से पहले इसे date format से parse किया गया है इससे इसे particular date में convert किया जाएगा और date variable में store किया जाएगा इस तरह हमारे पास old और new date available हो जाएगी अब हमे इन दोनों dates को compare करना है कि कोनसी date बड़ी या पहले आएगी या बाद में तो इस case में if लगाकर compare to method का use कर सकते है और हमने इन्हें compare किया

७



The screenshot shows the Eclipse IDE with a Java project named 'Java' and a source file 'Date1Exp.java'. The code in the editor is as follows:

```
8   DateFormat df = new SimpleDateFormat("dd-MM-yyyy");
9   Date ddate=df.parse("05-05-2012");
10  Date ndate=df.parse("10-03-2012");
11
12
13
14  if (ddate.compareTo(ndate) == 0) {
15      System.out.println(df.format(ddate) + " and " + df.format(ndate) + " are equal to each other");
16  }
17
18  if (ddate.compareTo(ndate) < 0) {
19      System.out.println(df.format(ddate) + " is less than " + df.format(ndate));
20  }
21
22  if (ddate.compareTo(ndate) > 0) {
23      System.out.println(df.format(ddate) + " is greater than " + df.format(ndate));
24  }
25
26
27
28
29
```

The console output shows the result of the execution:

```
<terminated> Date1Exp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 12:24:49 PM)
05-05-2012 is greater than 10-03-2012
```

हमने run किया तो हमारे पास output available है जिसमें मैं एक case execute हुआ है इस तरह से dates को compare कर सकते हैं

Date class in java

□ Date is a complex object in java , we have Date class in java to work with dates.

□ **GregorianCalendar class** is used to add or subtract date and time.

- Using DAY_OF_MONTH
- Using MONTH
- Using YEAR

```
GregorianCalendar gc = (GregorianCalendar)
GregorianCalendar.getInstance();

System.out.println(" " + gc.getTime());

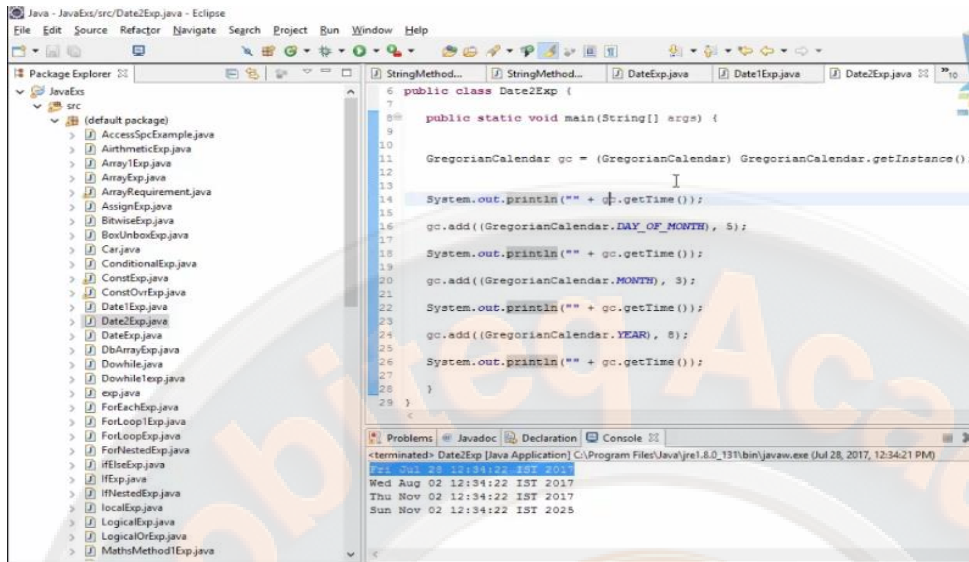
gc.add((GregorianCalendar.DAY_OF_MONTH), 5);
System.out.println(" " + gc.getTime());

gc.add((GregorianCalendar.MONTH), 3);
System.out.println(" " + gc.getTime());

gc.add((GregorianCalendar.YEAR), 8);
System.out.println(" " + gc.getTime());
```

Date2Exp.java

अब हम gregorian calendar class के बारे में जानेगे यह अलग से available class है इसका use यदि हम date के किसी पार्ट में कुछ जोड़ना चाहते है तो उसके लिए किया जाता है जैसे की कुछ दिन या महीने जोड़ना चाहते है तो ऐसे case में इसका use किया जाता है यह हमारे पास example भी available है यहा पर हमने gregorian calendar का gc एक object create किया है और इसे gregorian calendar का use करते हुए get instance के द्वारा initialize किया है इस तरह से यह date और time को setup करेगा अगर हम gc का simple use करेगे तो यह date time represent करेगा लेकिन अगर इसके कुछ constant use करेगे जैसे gc.add करके days of month में कुछ add कर सकते है इसके लिए हमे इस constant का use करना होगा इसमें, 5 लगाकर हमने 5 days add किए है उसके बाद get time और month और year में भी add किया गया है



```
6 public class Date2Exp {
7
8     public static void main(String[] args) {
9
10
11         GregorianCalendar gc = (GregorianCalendar) GregorianCalendar.getInstance();
12
13         // System.out.println(gc.getTime());
14         System.out.println("** + gc.getTime());
15
16         gc.add((GregorianCalendar.DAY_OF_MONTH), 5);
17
18         System.out.println("** + gc.getTime());
19
20         gc.add((GregorianCalendar.MONTH), 3);
21
22         System.out.println("** + gc.getTime());
23
24         gc.add((GregorianCalendar.YEAR), 8);
25
26         System.out.println("** + gc.getTime());
27     }
28 }
29 }
```

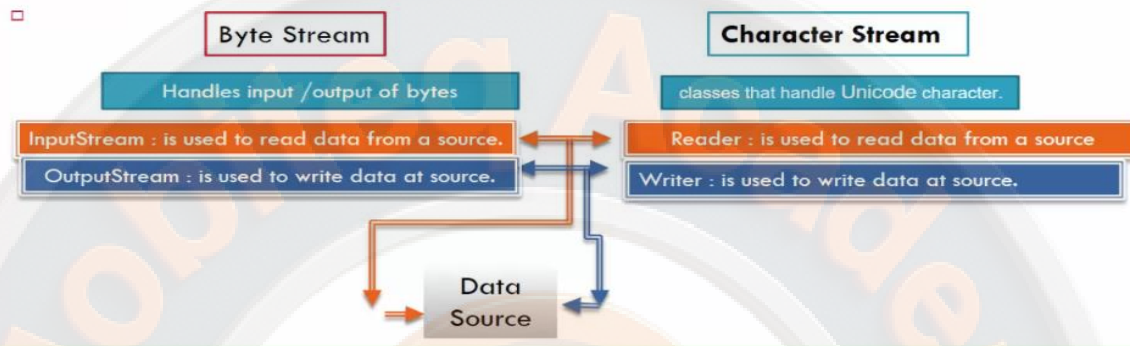
Problems | Javadoc | Declaration | Console

```
<terminated> Date2Exp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 12:34:21 PM)
Wed Aug 02 12:34:22 IST 2017
Thu Nov 02 12:34:22 IST 2017
Sun Nov 02 12:34:22 IST 2025
```

यहा पर gregorian calendar example को practical किया है और run करने के बाद output available है इस तरह date और time में कुछ add कर सकते है जिससे new dates को represent कर सकते है

Streams in Java

Streams are very important in java, Whenever we need to read or write data from different type of sources. We need stream classes and functions. Where



अब हम streams के बारे में जानकारी लेंगे streams का अर्थ है जब भी java में किसी source पर read करना है data या वहा से write करना है तो हमें streams की जरूरत रहती है इसके लिए हमें streams classes और उसके functions की जरूरत रहती है हमारे पास दो तरह की stream classes available है जोकि byte stream और charecter stream है यदि हम किसी source पर byte level पर कुछ read या write करना चाहते है तो byte stream का use कर सकते है और इसी तरह unicode charecter में कुछ read या write करना चाहते है तो charecter stream का use कर सकते है इनमें हमारे पास कुछ classes available है

Streams in Java

Byte Stream Classes

Stream class	Details
BufferedInputStream	To read Buffered Input Stream class.
BufferedOutputStream	To write Buffered Output Stream class.
DataInputStream	Class to read Standard datatype
DataOutputStream	Class to write Standard datatype
FileInputStream	Class to read from a file
FileOutputStream	Class to write to a file.
InputStream	Class for stream input.
OutputStream	Class for stream output.
PrintStream	Output Stream for printing output

WriteFile.java

Character Stream

Stream class	Details
BufferedReader	To read Buffered Input Stream class.
BufferedWriter	To write Buffered Output Stream class.
FileReader	Class to reads from file.
FileWriter	Class to writes to file.
InputStreamReader	Class translates byte to character for reading.
OutputStreamReader	Class to translate character to byte for writing.
Reader	Class for character stream input
Writer	Class for character stream output
PrintWriter	Output Stream for printing output

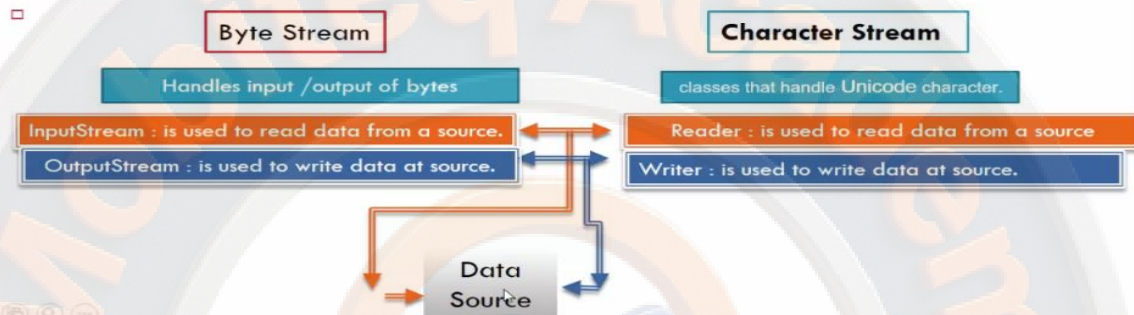
ReadFile.java

यहा पर दोनों की classes available है यहा पर byte stream में input read करने के लिए और output write करने के लिए है print stream से print कर सकते है इसी तरह से charecter stream भी available है इसमें अलग से classes available है

KEY TO SUCCESS

Streams in Java

Streams are very important in java, Whenever we need to read or write data from different type of sources. We need stream classes and functions. Where



हम यहा पर एक example से समझते है मान लीजिए कि हम इस data source file से कुछ read करना है अगर हम byte basic input output में read करना चाहते है तो इसके लिए byte की class का use कर सकते है जैसे input stream इससे हम आसानी से byte base पर कुछ भी read कर सकते है जैसे की arrow भी reflect कर रहे है इसी तरह अगर charecter stream से कुछ read करना चाहते है तो reader class available है इस तरह किसी भी source से आसानी से read कर सकते है इसी तरह write करने के लिए दोनों streams में class available है जिनका use कर सकते है

```

1
2 import java.io.*;
3 public class ReadWriteString
4 {
5     public static void main(String[] args) throws IOException
6     {
7         String s;
8         InputStreamReader in = new InputStreamReader(System.in);
9         BufferedReader br = new BufferedReader(in);
10        s = br.readLine(); //Reading from keyboard
11        System.out.println(s);
12    }
13 }

```

Problems | Javadoc | Declaration | Console

<terminated> Date2Exp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 12:34:21 PM)

Fri Jul 28 12:34:22 IST 2017
 Wed Aug 02 12:34:22 IST 2017
 Thu Nov 02 12:34:22 IST 2017
 Sun Nov 02 12:34:22 IST 2020

हमने पहले input और output के package को include किया है उसके बाद class का नाम है फिर static void main में एक exception throw की गई है फिर हमने string s लिया है एक input reader class create की है जिससे हम कुछ read करने वाले हैं हमने read करने के लिए जब input stream reader को initialize किया है तो constructor में system.in को पास किया है इससे जो भी input हम keyboard से देंगे वह system.in के द्वारा pick की जाएगी और वही input in object में store की जाएगी फिर हम buffer में read करना चाहते हैं कि line by line read करना चाहते हैं इसीलिए .in को buffer में पास किया है और यह br object में इकट्ठा करता रहेगा और यह br उस पूरी line को readline method के द्वारा read कर लेगा और उसे s variable में store कर देगा और फिर इसे print करवा लिया गया है जब हमने run किया तो हमें कुछ भी show नहीं किया गया है लेकिन फिर हमने कुछ type किया

```

1
2 import java.io.*;
3 public class ReadWriteString
4 {
5     public static void main(String[] args) throws IOException
6     {
7         String s;
8         InputStreamReader in = new InputStreamReader(System.in);
9         BufferedReader br = new BufferedReader(in);
10        s = br.readLine(); //Reading from keyboard
11        System.out.println(s);
12    }
13 }

```

Problems | Javadoc | Declaration | Console

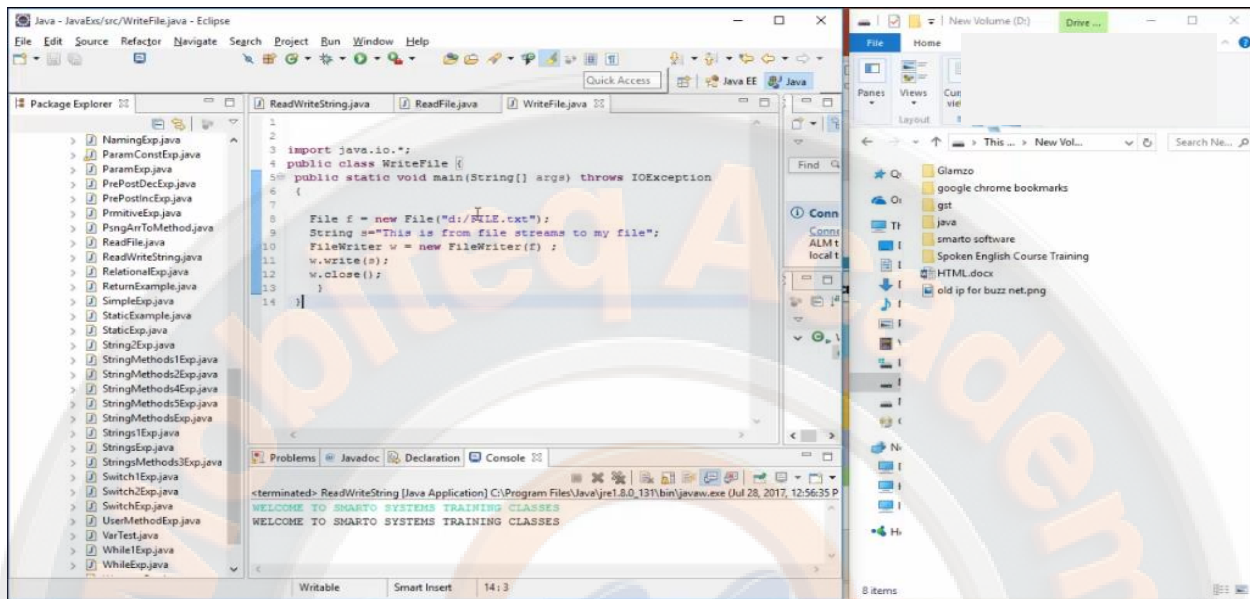
<terminated> ReadWriteString [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Jul 28, 2017, 12:56:35 PM)

WELCOME TO SMARTO SYSTEMS TRAINING CLASSES

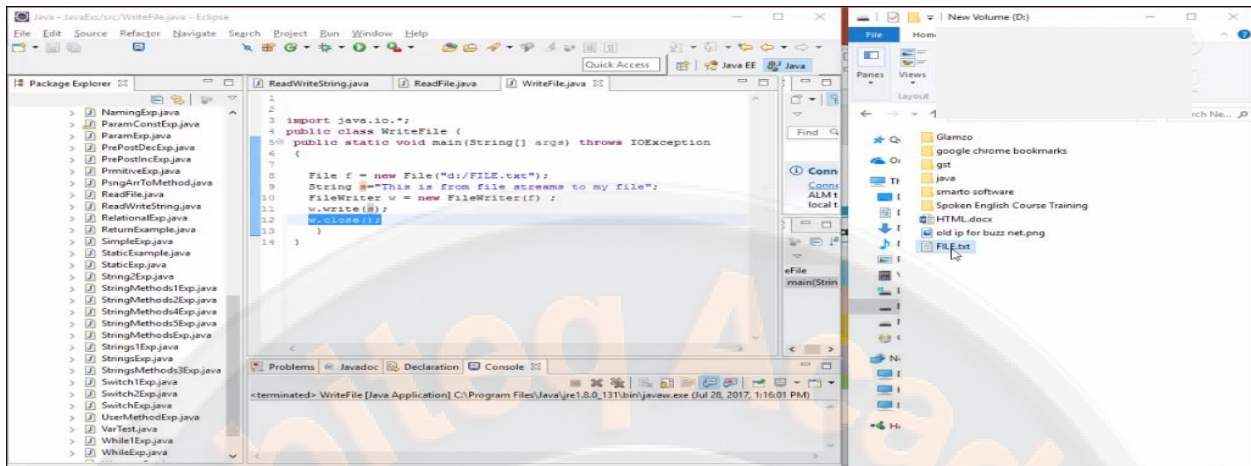
WELCOME TO SMARTO SYSTEMS TRAINING CLASSES

हमने कुछ type किया और enter press किया तो जितना भी हमने type किया था वह सारा input stream reader के द्वारा read कर लिया गया है और फिर buffer reader के द्वारा इकठ्ठा एक line में उठा लिया गया है और s में store करवाकर print करवाया गया है इस प्रकार keyboard से आसानी से read और write कर सकते हैं

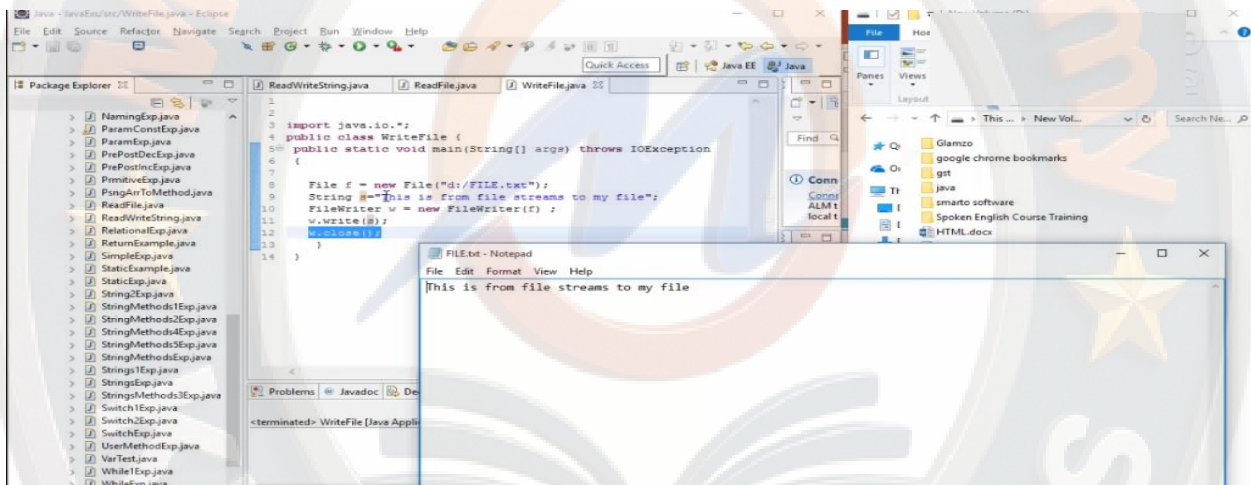
Writing and Reading on Files



हमने यहा पर left side पर writefile program को open किया है और right side पर file explorer को open किया है इसमें किसी भी प्रकार की file.txt available नहीं है हमने आपने program में package को include किया है और write file class डिफाइन की है फिर एक exception throw की है फिर file class का एक f object create किया है जिसमे एक file create की है जोकि d drive में है फिर एक string लिया है जिसमे कुछ store किया है अब हमे इस string को file पर write करना है इसके लिए filewriter class create की है और जिसका w एक object create किया है और जिस file पर write करेगी इसके एक object f पास किया है अब हम कुछ भी write करेगे तो वह direct file में write होती जाएगी इसके बाद string write करना है तो w हमारे पास function है जिससे हम आसानी से इस writer से file को write कर सकते है w.write और फिर string पास किया है इसके बाद close किया है



हमने program को run किया तो हमारे पास console में कुछ available नहीं हुआ है लेकिन file explorer में file.txt available हो गई है हमने इसे double क्लिक करके open किया



तो जो message हमने वहा पर type किया था वह available है इस तरह हमने एक file को create करके उसमे कुछ text को input किया है अभी हमने file को write किया है

```

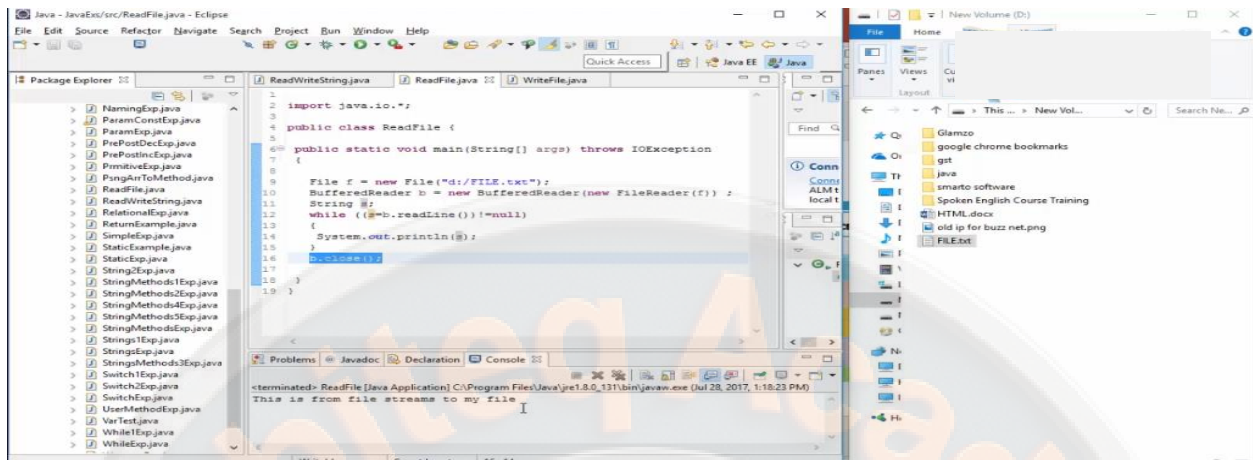
1 import java.io.*;
2
3
4 public class ReadFile {
5
6     public static void main(String[] args) throws IOException
7     {
8
9         File f = new File("d:/FILE.txt");
10        BufferedReader b = new BufferedReader(new FileReader(f));
11        String s;
12        while ((s=b.readLine())!=null)
13        {
14            System.out.println(s);
15        }
16        b.close();
17    }
18 }
19 }

```

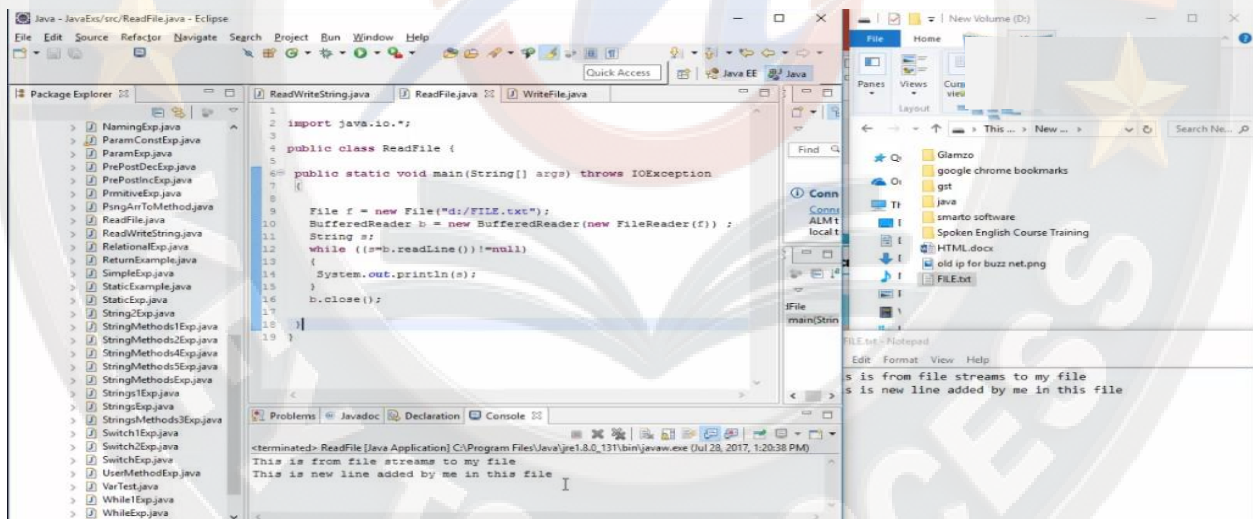
Problems | Javadoc | Declaration | Console

<terminated> WriteFile [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\java.exe (Jul 28, 2017, 1:16:01 PM)

यहा पर हमने read file program को open किया है यहा पर हमने एक file को f object से pick किया है और फिर file का reference दे दिया है कि इस file से हमे कुछ read करना होगा तो इस file को हमने f object में ट्रांसफर किया है फिर हमने buffer reader class का use किया है इसके लिए constructor में एक file reader का object पास किया है इस data को उठाने के लिए एक buffer reader चाहिए तो इसके लिए b का use object के रूप में किया है इसके बाद string s लिया है यह जो read होगा उसे s में save करता जाएगा उसके बाद loop का use किया है क्योंकि हमे file के content के बारे में पता नही है इस loop में हमने यह डिफाइन किया है कि जब तक string में completely file का information ना आ जाए तब तक यह line read करता रहे और यह line s में इकठ्ठा available हो जाएगा और फिर उसे print कर सकते है और फिर close किया है



हमने run किया तो हमारे पास console panel में class का information available हो गया है इस तरह file को read कर सकते हैं



हमने आपनी file txt में manually जाकर कुछ type करके save कर दिया है और हमने फिर से read file program को run किया तो वह उस line को भी read कर रहा है जोकि हमने बाद में add की है इस प्रकार file से read और write कर सकते हैं

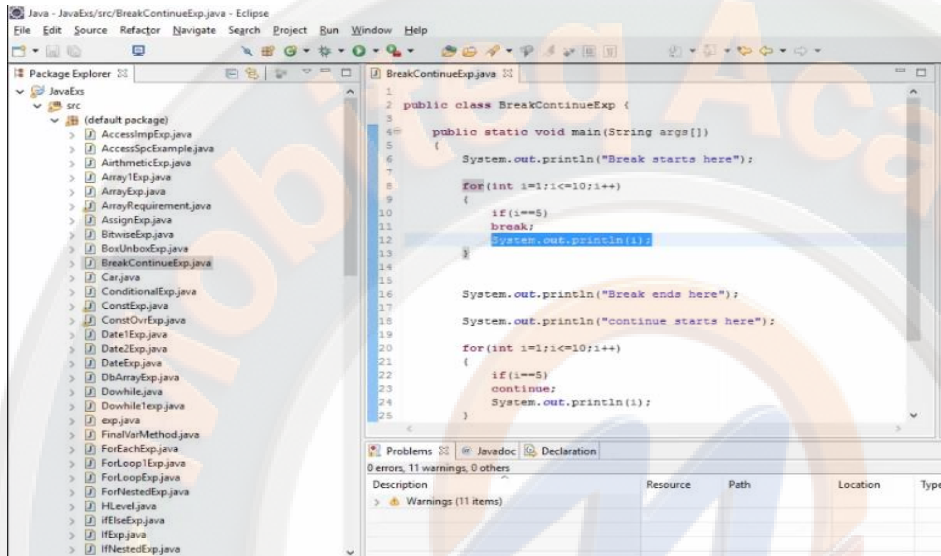
Break And continue in Java.

- | | |
|--|--|
| <ul style="list-style-type: none"> ❑ break keyword is used to stop a loop execution. It is used mainly for while loop do while, for loop and for each loop. ❑ Once break comes in loop, it does not execute further. | <ul style="list-style-type: none"> ❑ continue keyword is used to skip specific part under continue in loop execution. It is used mainly for while loop do while, for loop and for each loop. ❑ Once continue comes in loop, it executes further but skips the execution of under continue part. |
|--|--|

अब हम break and continue के बारे में जानेगे यह बहुत ही important keyword है

Break keyword -: इसका use किसी भी समय loop की execution को रोकने के लिए किया जा सकता है जैसे ही किसी loop में break मिलता है वह loop वहा पर stop हो जाती है और वह उसके आगे नहीं बढ़ती है

Continue keyword -: इसका use अगर continue के नीचे कुछ statement है उसे skip करने के लिए किया जाता है अगर हम किसी particular condition को satisfied करते है ऐसे में यह भी loop के साथ काम करता है हम किसी भी loop में इसका use करते है जब loop के execution में continue आता है तो condition सही रहती है तब continue का नीचे वाला पार्ट execute नहीं होता है परन्तु loop अपना पूरा execution करता है



```

1 public class BreakContinueExp {
2
3     public static void main(String args[])
4     {
5         System.out.println("Break starts here");
6
7         for(int i=1;i<=10;i++)
8         {
9             if(i==5)
10            break;
11            System.out.println(i);
12
13
14
15
16            System.out.println("Break ends here");
17
18            System.out.println("continue starts here");
19
20            for(int i=1;i<=10;i++)
21            {
22                if(i==5)
23                    continue;
24                System.out.println(i);
25            }
26        }
27    }
28 }

```

यहाँ हमने break and continue का example लिया है हमने यहाँ पर class को डिफाइन किया है इसके बाद static void main में एक message print करवाया है उसके बाद एक loop है loop में हमें 1 से 10 तक जाना है और 1,1 जोड़ते जाना है परन्तु अगर loop में i अगर 5 के बराबर हो जाता है तो यह loop वहाँ पर break हो जाए और फिर i को print करवाया है अब यह देखना है break keyword के बाद यह line को execution करेगा या नहीं इसके लिए हमने इसे run किया है और right side console panel को देखते हैं इसमें हमारी 1st line available है उसके बाद loop 1 से 4 तक चलता है क्योंकि इसे break keyword मिल गया है और उसकी आगे execution नहीं होती है इसके बाद for loop execute नहीं हुई है यह process condition satisfied होने के साथ ही बंद हो जाती है

```

8      for (int i=1;i<=10;i++)
9      {
10     {
11     if (i==5)
12     break;
13     System.out.println(i);
14     }
15     }
16     System.out.println("Break ends here");
17     }
18     System.out.println("Continue starts here");
19     }
20     for (int i=1;i<=10;i++)
21     {
22     if (i==5)
23     continue;
24     System.out.println(i);
25     }
26     System.out.println("Continue ends here");
27     }
28     }
29     }
30     }
31     }
32     }

```

इसके बाद continue के बारे में जानते हैं इसमें हमने एक message print करवाया है जोकि output में show भी किया जा रहा है फिर वही same loop है इसमें हमने बस अगर $i=5$ हो जाए तो उसके बाद continue keyword का use किया है और output में पूरी loop execute हुई है इसमें केवल 4 के बाद 5 नहीं आया है क्योंकि हमने loop में $i=5$ हो तो continue कर दिया जाए और फिर उसे print करवाया गया है ऐसे में जब 5 हुआ तो यही से loop को continue कर दिया गया अर्थात् continue के नीचे वाले पार्ट को ignore कर दिया गया है जब condition true रहती है इस तरह break और continue में फर्क रहता है इन दोनों process में दोनों keyword का use के जैसे तरीके से ही किया जा रहा है इनमें फर्क केवल executional पार्ट में है break loop को वहीं पर खत्म कर देता है और continue अपने नीचे आने वाली statement को skip कर देता है उनको execute नहीं करता है यह दोनों में main फर्क है और इस तरह से इनका use कर सकते हैं

Inheritance in Java

- ❑ Inheritance is an important feature of OOPS(Object Oriented Programming System)
- ❑ Inheritance means a process where one class gets the properties (methods and fields) of another class.
- ❑ Now in inheritance there are more than one classes required.
- ❑ The class whose properties are inherited is called as **superclass /base class/parent class**.
- ❑ The class which gets/receives/ inherits the properties of other class is called as **subclass /derived class/child class**.
- ❑ **extends** is the keyword we use to create sub or super classes and get properties of a other class.
- ❑ So we can easily reuse code of one class (super) into another class (sub class) means we can reuse one class (super) members like variable and methods in another class (sub) .

```
class SuperClass
{
    (methods and fields)
}

class SubClass extends SuperClass{
    (methods and fields)
}
```

SuperCalculator.java

अब हम java में inheritance के बारे में जानेगे यह बहुत ही important concept है क्योंकि यह object oriented programming system से related है उसका एक important feature है जिसे हम OOPS कहते है

यदि हम एक class की properties दूसरी class को provide करवाते है तो यह inheritance कहलाता है इस में एक class की properties जिसमे methods और fields दोनों होते है उनको दूसरी class के द्वारा use किया जा सकता है

inheritance में एक से ज्यादा classes होती है

वह class जिसकी properties प्राप्त की जाती है अर्थात जो class properties provide करती है उसे हम superclass/ baseclass / parent class कहते है

वह class जो properties को प्राप्त करती है जो इसका use करती है उसे subclass / derived / child class कहते हैं

इसके लिए हमने right side पर example भी show किया है

इसके लिए हमें extends keyword का use किया जाता है इससे आसानी से superclass और subclass को create कर सकते हैं जैसे की example में हमने use भी किया है

इन सबका फायदा यह रहता है कि जो field या method हमने super class में create कर लिए हैं उसे हम sub class में reuse कर सकते हैं हमें दोबारा बनाने की जरूरत नहीं होती है हमें बार बार code create नहीं करना पड़ता है

```

1 class BasicCalculator
2 {
3     protected //private
4     int z;
5
6     public void addNumber(int x, int y) {
7         z = x + y;
8         System.out.println("The sum of the given numbers:"+z);
9     }
10
11    public void SubtractNumber(int x, int y) {
12        z = x - y;
13        System.out.println("The difference between the given numbers:"+z);}
14    }
15
16
17
18 }
19 public class SuperCalculator extends BasicCalculator {
20
21    public void multiplyNumber(int x, int y)
22    {
23        z = x * y;        System.out.println("The product of the given nu
24    }
25    public void DivideNumber(int x, int y)
26    {
27        z = x / y;
28    }
29 }

```

Problems Javadoc Declaration Console

```

<terminated> SuperCalculator [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 2, 2017, 2:09:13 AM)
The sum of the given numbers:60
The difference between the given numbers:40
The product of the given numbers:500

```

हमने आपने program को open कर लिया है हमने यहा पर class create की है इसमें हमने एक method public void addnumber create किया है इसमें हम x और y को parameter की तरह

receive करते हैं और जोड़कर print कर देते हैं इसी तरह हमने subtract number method भी create किया है इसमें भी हमने x और y को receive करके total करके print किया है

इसके बाद हमने supercalculate class create की है इसको हमने basic calculator की subclass बनाया है जोकि हमने extends keyword के साथ बनाई है यहा पर left side की class subclass और extend के right side class superclass कहलाएगी इस प्रकार हम basic calculator की सभी चीजे हम नीचे use करेंगे

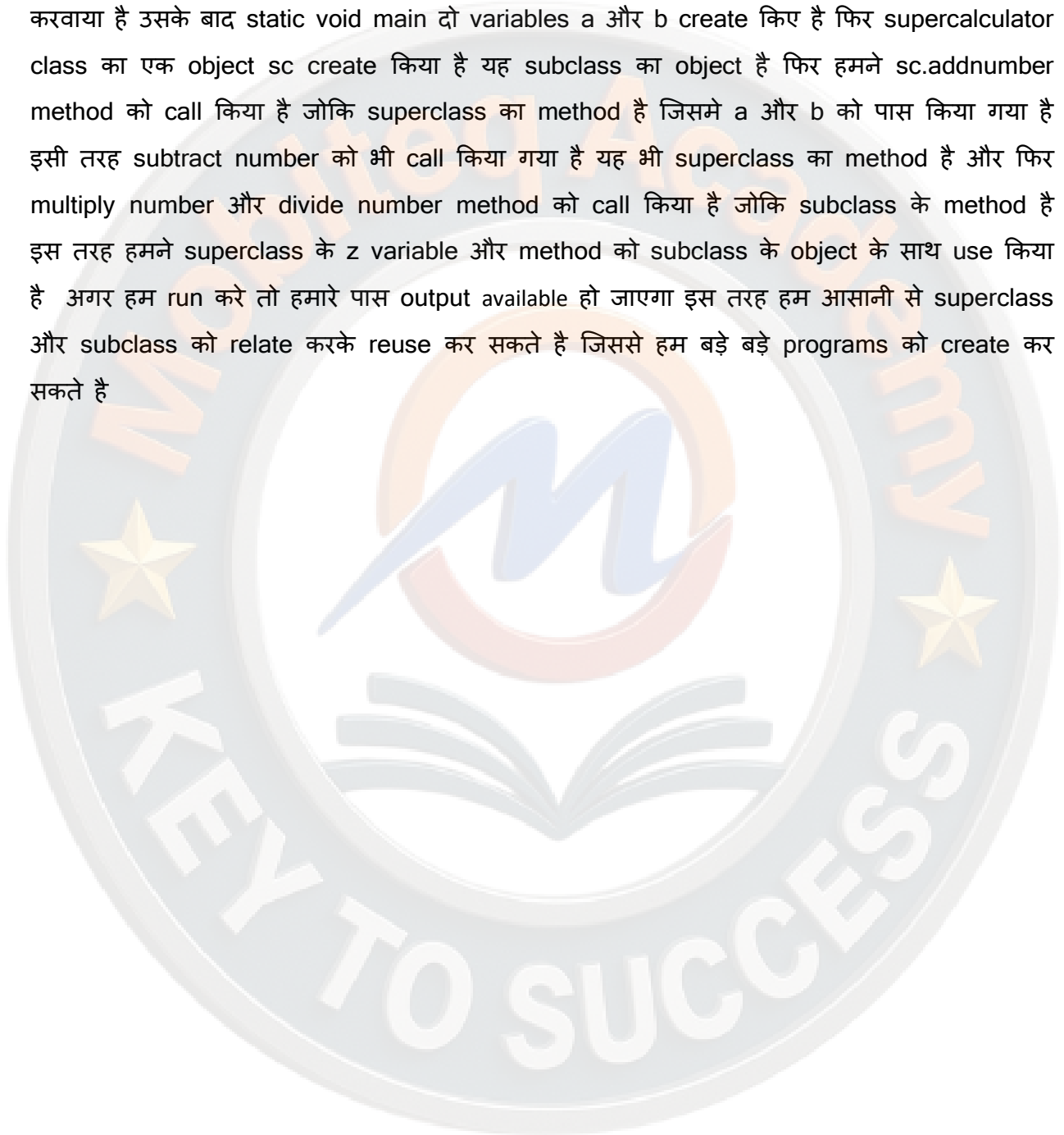
```

15
16
17
18
19 public class SuperCalculator extends BasicCalculator {
20
21     public void multiplyNumber(int x, int y)
22     {
23         z = x * y;
24         System.out.println("The product of the given numbers:"+z);
25     }
26
27     public void DivideNumber(int x, int y)
28     {
29         z = x / y;
30         System.out.println("The Division of the given numbers:"+z);
31     }
32
33     public static void main(String args[]) {
34         int a = 60, b = 40;
35         SuperCalculator sc = new SuperCalculator();
36         sc.addNumber(a, b);
37         sc.SubtractNumber(a, b);
38         sc.multiplyNumber(a, b);
39         sc.DivideNumber(a, b);
40     }
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

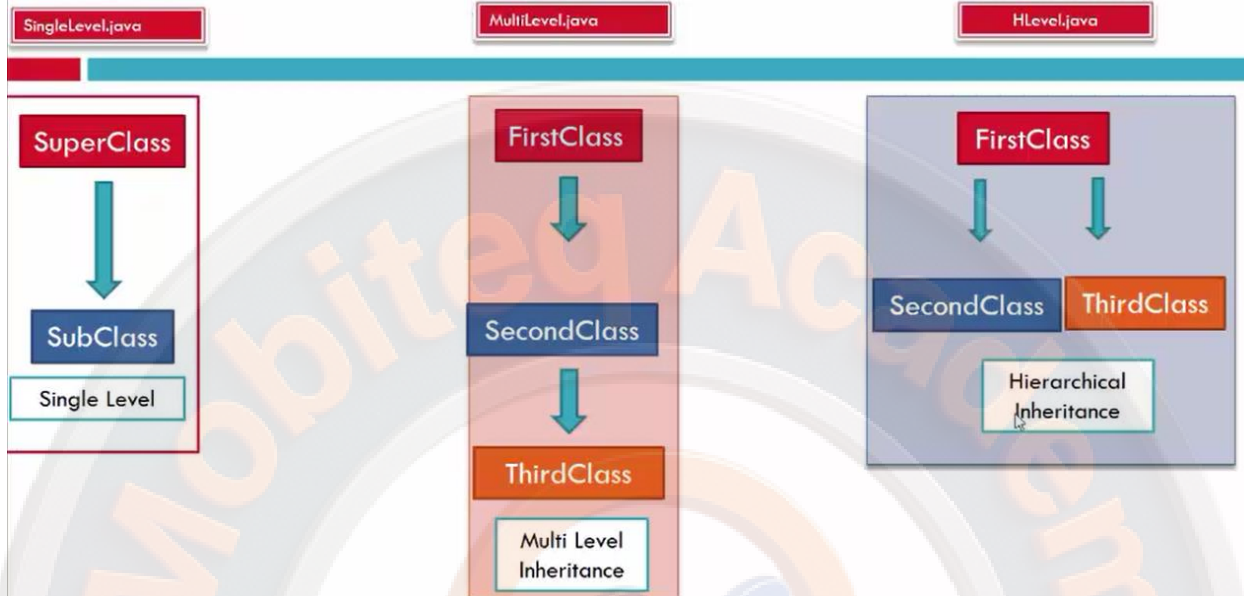
```

<terminated>- SuperCalculator [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 2, 2017, 2:09:13 AM)
 The sum of the given numbers:60
 The difference between the given numbers:40
 The product of the given numbers:500

इसके लिए हमने multiply number method और divide method का use किया है हमने इन दोनों में कहीं भी z को declare नहीं किया है हमने z ऊपर use किया है z basic calculator class का variable है और इसे हम subclass में use किया है हमने यहाँ पर calculation किया और print करवाया है उसके बाद static void main दो variables a और b create किए हैं फिर supercalculator class का एक object sc create किया है यह subclass का object है फिर हमने sc.addnumber method को call किया है जोकि superclass का method है जिसमें a और b को पास किया गया है इसी तरह subtract number को भी call किया गया है यह भी superclass का method है और फिर multiply number और divide number method को call किया है जोकि subclass के method हैं इस तरह हमने superclass के z variable और method को subclass के object के साथ use किया है अगर हम run करें तो हमारे पास output available हो जाएगा इस तरह हम आसानी से superclass और subclass को relate करके reuse कर सकते हैं जिससे हम बड़े बड़े programs को create कर सकते हैं



Types of Inheritance in Java



अब हम inheritance के प्रकार के बारे में जानेगे

Single level

```

1 class First
2 {
3     protected //private
4     int z;
5
6     public void addNumber(int x, int y) {
7         z = x + y;
8         System.out.println("The sum of the given numbers:"+z);
9     }
10
11    public void SubtractNumber(int x, int y) {
12        z = x - y;
13        System.out.println("The difference between the given numbers:"+z);
14    }
15
16    }
17
18 }
19 public class SingleLevel extends First{
20
21    public void multiplyNumber(int x, int y)

```

<terminated> SingleLevel [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 2, 2017, 2:25:55 AM)
 The sum of the given numbers:60
 The difference between the given numbers:40
 The product of the given numbers:500
 The Division of the given numbers:

यह example कुछ पिछले example की तरह ही है हमने यहा पर first class create की है फी एक variable declare किया है और फिर दो method को use किया है इसके बाद फिर single level class create की है जिसने first को extends किया है इसके बाद हमने दो method का use किया है हमने इन दोनों में कही भी z को declare नहीं किया है हमने z ऊपर use किया है z first class का variable है ओर इसे हमने subclass में use किया है हमने यहा पर calculation किया और print करवाया है उसके बाद static void main दो variables a और b create किए है फिर single class का एक object sc create किया है यह subclass का object है फिर हमने sc.addnumber method को call किया है जोकि superclass का method है जिसमे a और b को पास किया गया है इसी तरह subtract number को भी call किया गया है यह भी superclass का method है और फिर multiply number और divide number method को call किया है जोकि subclass के method है इस तरह हमने superclass के z variable और method को subclass के object के साथ use किया है अगर हम run करे तो हमारे पास output available हो जाएगा इस relationship को single level कहते है क्योंकि इसमें एक class superclass और एक subclass है और यह process एक class तक ही limited है

Multi level

```

1
2 class FirstClass {
3     int z;
4
5     public void addNumber(int x, int y) {
6         z = x + y;
7         System.out.println("The sum of the given numbers:"+z);
8     }
9 }
10
11 class SecondClass extends FirstClass
12 {
13
14
15     public void SubtractNumber(int x, int y) {
16         z = x - y;
17         System.out.println("The difference between the given numbers:"+z);
18     }
19 }
20
21
22
23
24 public class MultiLevel extends SecondClass {
25
26
27     public void multiplyNumber(int x, int y)
28     {
29         z = x * y;
30         System.out.println("The product of the given numbers:"+z);
31     }
32     public void DivideNumber(int x, int y)
33     {
34         z = x / y;
35         System.out.println("The Division of the given numbers:"+z);
36     }
37
38     public static void main(String args[]) {
39         int a = 50, b = 10;
40         MultiLevel sc = new MultiLevel();
41         sc.addNumber(a, b);
42         sc.SubtractNumber(a, b);
43         sc.multiplyNumber(a, b);
44         sc.DivideNumber(a, b);
45     }
46 }
47
48
49
50

```

Problems | Javadoc | Declaration | Console

```

<terminated> SingleLevel [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 2, 2017, 2:25:55 AM)
The sum of the given numbers:60
The difference between the given numbers:40
The product of the given numbers:500
The Division of the given numbers:5

```

हमने यहा पर एक class create की है इसमें int z है और फिर addnumber method लिया है और दो number add करके display किया है उसके बाद second class create की है जिसने first class को extends किया है इस तरह first class superclass है और secondclass subclass है इसमें subtract method का use किया है इसमें z का direct use किया गया है इसके बाद हमने multi level class create की है जिसने second class को extends कर लिया है इस तरह second class सुपर class और multi level subclass है इस तरह first class की सभी चीजे second में available हुई फिर second की सभी चीजे multi level class को available हुए फिर हम public static void main में a और b को value की तरह use किया है फिर multi level का एक object sc create किया है फिर हमने methods को call किया है हमने program को run किया तो हमारे पास output available है इस तरह अगर एक class की properties दूसरी class inherit करती है और दूसरी class की properties तीसरी class inherit करती है तो इस chain को multi level inheritance कहते है इसमें भी एक class ही properties देती है और दूसरी class प्राप्त करती है एक से अधिक class को inherit नहीं कर सकते है

Hierarchical inheritance

The screenshot displays the Eclipse IDE with a Java project named 'Java - JavaBst/src/HLevel.java'. The Package Explorer on the left shows a hierarchy of classes, including 'DateExp.java', 'DateExp.java', 'DbArrayExp.java', 'Dowhile.java', 'DowhileExp.java', 'exp.java', 'FinalVarMethod.java', 'ForEachExp.java', 'ForLoopExp.java', 'ForNestedExp.java', 'HLevel.java', 'HLevelExp.java', 'HExp.java', 'HNestedExp.java', 'localExp.java', 'LogicalExp.java', 'LogicalOrExp.java', 'MathsMethodExp.java', 'DowhileExp.java', 'exp.java', 'FinalVarMethod.java', 'ForEachExp.java', 'ForLoopExp.java', 'ForNestedExp.java', 'HLevel.java', 'HLevelExp.java', 'HExp.java', 'HNestedExp.java', 'localExp.java', 'LogicalExp.java', 'LogicalOrExp.java', 'DateExp.java', 'DateExp.java', 'DbArrayExp.java', 'Dowhile.java', 'DowhileExp.java', 'exp.java', 'FinalVarMethod.java', 'ForEachExp.java', 'ForLoopExp.java', 'ForNestedExp.java', 'HLevel.java', 'HLevelExp.java', 'HExp.java', 'HNestedExp.java', 'localExp.java', 'LogicalExp.java', 'LogicalOrExp.java', 'MathsMethodExp.java', 'MathsMethodExp.java', 'MethodExample.java', 'MethodOverExample.java', 'MultiLevel.java', 'MultiMatrix.java', 'NameExample.java', 'NamingExp.java', 'OverridExp.java', 'ParamConstExp.java', 'ParamExp.java', 'PrePostDecExp.java', 'PrePostIncExp.java', 'PrimitiveExp.java', and 'PngAnToMethod.java'.

The main editor shows the following Java code:

```

1  class FirstClass {
2      int z;
3
4
5      public void addNumber(int x, int y) {
6          z = x + y;
7          System.out.println("The sum of the given numbers:"+z);
8      }
9  }
10
11 class SecondClass extends FirstClass
12 {
13
14
15      public void SubtractNumber(int x, int y) {
16          z = x - y;
17          System.out.println("The difference between the given numbers:"+z);
18      }
19  }
20
21
22
23
24 class ThirdClass extends FirstClass
25 {
26
27      public void multiplyNumber(int x, int y)
28      {
29          z = x * y;
30          System.out.println("The product of the given numbers:"+z);
31      }
32
33
34
35
36
37 public class HLevel extends SecondClass {
38
39
40
41      public void DivideNumber(int x, int y)
42      {
43          z = x / y;
44          System.out.println("The Division of the given numbers:"+z);
45      }
46
47      public static void main(String args[]) {
48          int a = 50, b = 10;
49          HLevel sc = new HLevel();
50          sc.addNumber(a, b);
51          sc.SubtractNumber(a, b);
52          sc.DivideNumber(a, b);
53      }
54  }
55
56

```

The console output shows the following results:

```

<terminated> MultiLevel [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 2, 2017, 2:30:14 AM)
The sum of the given numbers:60
The difference between the given numbers:40
The product of the given numbers:500
The Division of the given numbers:5

```

यहा पर हमने first class1 create की है जिसमे एक method का use किया है और फिर एक second class1 create की है जोकि first class1 को extends करती है इसका भी एक method है इसके बाद third class1 create की है जोकि first class1 को दोबारा inherit करती है इस तरह first class1 का method और z third class1 में भी आ गया है इस तरह से किसी superclass की एक से ज्यादा subclass create कर सकते है फिर हमने hlevel एक class create की है इसमें second को extends किया है इसमें एक method का use किया है इस तरह से हमने hierarchical structure create किया है इसी को हम hierarchical inheritance कहते है

Sub class and Super class

- ❑ We can inherit members from super class and also create or add new members in subclass:
- ❑ We can use inherited fields/methods directly.
- ❑ We can declare new fields and new methods in the subclass that are not in the superclass.
- ❑ We can add same name method in subclass which is already present in super class, this process is called overriding.
- ❑ We can create a subclass constructor which invokes the constructor of the superclass and can be invoked using the keyword super.

यहा पर superclass और subclass के कुछ ओर features के बारे में बताया गया है



Access specifiers

Defines the accessibility of any class/ methods/variables .

1. public : Accessibility to public means you can access variable/ methods from any class in Java anywhere.
2. private : Accessibility to private means you can access variable/ methods within only its declared class.
3. protected: protected means you can access variable/ methods within only its declared class or in its subclass.
4. default: means you can access variable/only in your package but not in another package.



```
public static void main(String[] args)
{
    // statements are defined here.
}
```

अब हम access specifier के बारे में जानकारी लेंगे हमने पहले भी इसके बारे में जानकारी ली थी जिसमें हमने public के बारे में जाना था जोकि हमने breif format में जाना था अब हम इसके बारे में properly

जानेगे हम access specifier से accessibility को डिफाइन करते हैं यह चार प्रकार के होते हैं अगर हम कोई भी access specifier नहीं लगाए तो वह default माना जाता है

1st public यदि किसी variable या method को public डिफाइन करते हैं तो हम उसे कहीं भी access कर सकते हैं आपनी class, subclass में कहीं भी कर सकते हैं

private यदि किसी variable या method को private डिफाइन करते हैं तो वह केवल आपनी ही class में use किया जाता है इसे किसी दूसरी subclass में use नहीं कर सकते हैं

protected इसके variable और method को आपनी class और इसकी subclass में use कर सकते हैं

default यह कोई access specifier वाला keyword नहीं है अगर हम कोई भी access specifier नहीं लगाए तो वह default माना जाता है यह आपने package में available रहता है लेकिन दूसरे package में available नहीं होता है

हमने एक example show किया है जिसमें हमने public access specifier का use किया है जोकि उस method की accessibility को डिफाइन करता है



Access Specifiers In Java

Access Modifier	Within Class Same Pkg	Within Other Class Same Package	Outside Package By Subclass Only	Outside Package
Public	Y	Y	Y	Y
Protected	Y	Y	Y	N
Default	Y	Y	N	N
Private	Y	N	N	N

AccessImpExp.java

AcsSpcExp.java

यहा पर हमारे पास एक चार्ट available है जिससे हम access specifier को आसानी से समझ सकते है इसमें जहा जहा specifier को use कर सकते है वह y पार्ट द्वारा बताया गया है और जो yellow पार्ट है यानि कि n पार्ट है वहा use नही किए जा सकते है अब हम इन सभी के practical example की समझेगे



Access Specifiers In Java

package one;

Class FirstClass

```
{
}
```

Class SecondClass

```
{
}
```

package two;

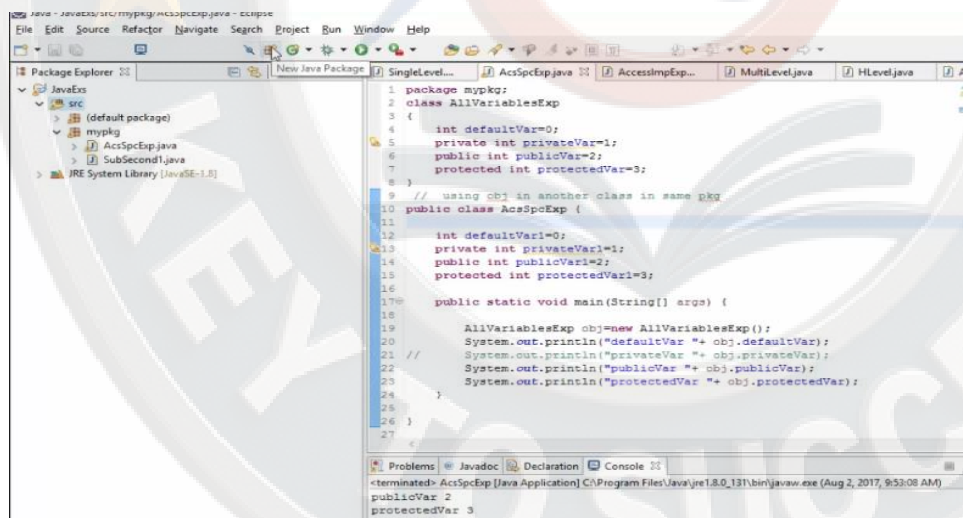
Class ThirdClass

```
{
}
```

Class FourthClass

```
{
}
```

package को एक फोल्डर की तरह समझ सकते हैं यहा पर हमारे पास दो package available हैं यह दोनों अलग अलग package हैं



```

1 package mypkg;
2 class AllVariablesExp
3 {
4     int defaultVar=0;
5     private int privateVar=1;
6     public int publicVar=2;
7     protected int protectedVar=3;
8 }
9 // using obj in another class in same pkg
10 public class AcSpExp {
11
12     int defaultVar1=0;
13     private int privateVar1=1;
14     public int publicVar1=2;
15     protected int protectedVar1=3;
16
17     public static void main(String[] args) {
18
19         AllVariablesExp obj=new AllVariablesExp();
20         System.out.println("defaultVar "+ obj.defaultVar);
21         System.out.println("privateVar "+ obj.privateVar);
22         System.out.println("publicVar "+ obj.publicVar);
23         System.out.println("protectedVar "+ obj.protectedVar);
24     }
25
26 }
27

```

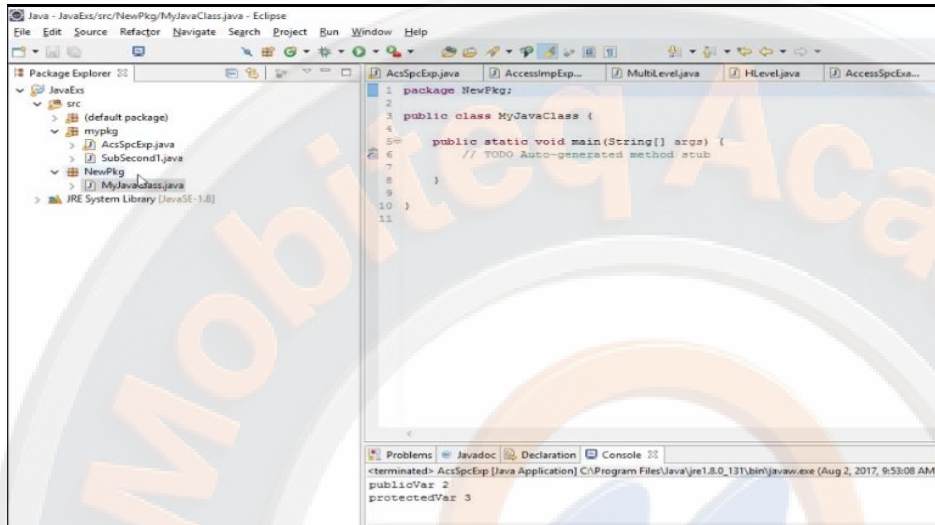
Problems | Javadoc | Declaration | Console

<terminated> AcSpExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\java.exe (Aug 2, 2017, 9:53:08 AM)

publicVar 2
protectedVar 3

अगर हम अपना एक package बनाना चाहते हैं तो हम left side src पर क्लिक करके ऊपर से new java package icon पर क्लिक करे एक box available होगा इसमें आपने package का नाम type करे और finish पर क्लिक करे तो left side पर package ओर जुड़ गया है यहा पर mypackage में हमारे package का name available हो गया है अगर इस package में कोई class create करना चाहते हैं तो right क्लिक करके new पर जाकर class पर क्लिक करे एक box available होगा यहा पर हमारे

package का name available है हमें यहाँ पर class का name type करना है और static void main option पर क्लिक करके finish करना है



यहाँ पर हम देख सकते हैं कि package में एक class available हो गई है हम program में देखें तो सबसे पहले package फिर package का नाम है उसके बाद class create की है फिर सारा structure same है इस तरह से package डिफाइन कर सकते हैं और package के अंदर classes भी डिफाइन कर सकते हैं अगर हमें एक package की class किसी दूसरे package में use करनी है तो हमें उसके लिए import keyword का use करना होगा और उसके साथ .* का use करना होगा जैसे कि import mypkg.* इससे जितनी भी classes हैं उनसे related जो भी function है वह हमारे पास available हो जाएगी अगर हमें किसी एक class का use करना है तो हम dot के बाद उस class का नाम दे सकते हैं इस तरह से एक package की class का दूसरे package की class में use कर सकते हैं

Private Access Specifier

```

1 package mypkg;
2 class AllVariablesExp
3 {
4     int defaultVar=0;
5     private int privateVar=1;
6     public int publicVar=2;
7     protected int protectedVar=3;
8 }
9 // using obj in another class in same pkg
10 public class AcSpcExp {
11
12     int defaultVar=0;
13     private int privateVar=1;
14     public int publicVar=2;
15     protected int protectedVar=3;
16
17 @
18     public static void main(String[] args) {
19         AllVariablesExp obj=new AllVariablesExp();
20         System.out.println("defaultVar "+ obj.defaultVar);
21         // System.out.println("privateVar "+ obj.privateVar);
22         System.out.println("publicVar "+ obj.publicVar);
23         System.out.println("protectedVar "+ obj.protectedVar);
24     }
25 }
26 }
27

```

Problems | Javadoc | Declaration | Console

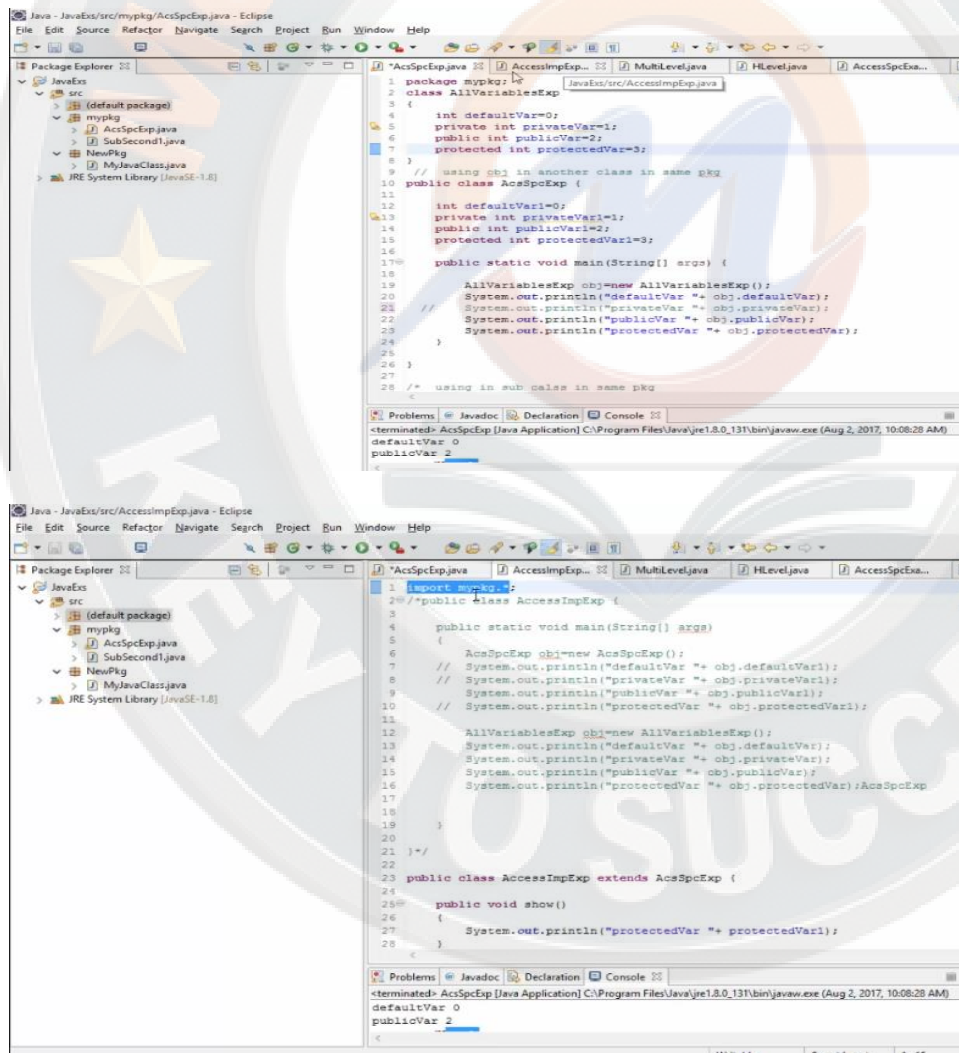
```

<terminated> AcSpcExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\java.exe (Aug 2, 2017, 9:53:08 AM)
publicVar 2
protectedVar 3

```

अब हम access specifier example को देखते हैं यहा पर हमने package include किया है फिर एक class all variableexp create की है इसमें हमने चार तरह के variables लिए है एक default variable है एक private, एक public और एक protected variable है यहा पर default कोई भी keyword नहीं होता है अगर हम कोई भी keyword ना लगाए तो वह default माना जाता है अब हम इस class का variable दूसरी class में use करना चाहते हैं तो इसका एक object create कर सकते हैं या फिर एक subclass बना ले अभी हमने नीचे एक class acSpcExp एक class create की है हमने वही variables दोबारा लिए है फिर static void main में जाकर एक object create किया है और variables को call किया है अगर हम run करे तो हमारे पास output में public protected और default output available है इसमें हमारे पास private output available नहीं है क्योंकि हमने उसे comment बनाया हुआ है अगर हम इसे comment से हटा देते हैं तो हमारे पास error available हो जाता है कि हम private variable को यहा पर use नहीं कर सकते हैं क्योंकि private केवल आपनी class में ही accessible रहता है इसे हम किसी दूसरी class में use नहीं कर सकते हैं हमने आपने variable को ऊपर class में declare किया है और वह उसी block तक ही available है इस तरह private का use हम उसी class में कर सकते हैं जहा पर उसे declare किया गया है

Default Access Specifier



ऊपर image में हमने default variable declare किया है और उसे हमने नीचे call किया है और वह वहा पर accessible है क्योंकि यह same package में है अब हम किसी दुसरे package में देखते है

हमने public class में भी चारों variables को declare किया हुआ है हम देखना चाहते हैं कि इस class के variables दूसरे package में ले जाए तो क्या इसके variables create करके use कर सकते हैं

यहा पर एक दूसरा package भी use किया है जिसमे हमने import mypkg.* का use किया है

```

15 System.out.println("publicVar "+ obj.publicVar);
16 System.out.println("protectedVar "+ obj.protectedVar);AcSpExp
17 }
18
19
20 }*/
21
22 public class AccessImpExp extends AcSpExp {
23
24     public void show()
25     {
26         System.out.println("protectedVar "+ protectedVar);
27     }
28
29     public static void main(String[] args)
30     {
31         AcSpExp obj=new AcSpExp();
32         AccessImpExp obj1=new AccessImpExp();
33
34         obj1.show();
35         // System.out.println("defaultVar "+ obj.defaultVar);
36         // System.out.println("privateVar "+ obj.privateVar);
37         // System.out.println("publicVar "+ obj.publicVar);
38         // System.out.println("protectedVar "+ obj.protectedVar);
39
40         // AllVariablesExp obj=new AllVariablesExp();
41         // System.out.println("defaultVar "+ obj.defaultVar);
42         // System.out.println("privateVar "+ obj.privateVar);

```

```

<terminated> AcSpExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\java.exe (Aug 2, 2017, 10:08:28 AM)
defaultVar 0
protectedVar 2

```

इसके लिए हम दूसरे package में गए यहा पर या तो हम उस class जोकि acsSpCexp है उसे extends करके subclass बना दे यहा पर acsSpCexp की subclass accessimpexp बन गई या फिर हम access imp exp का object create कर ले हमने एक obj create किया है और फिर इस obj के साथ particular variable को call करना चाहते हैं

```

21
22 public class AccessImpExp extends AcSpExp {
23
24     public void show()
25     {
26         System.out.println("protectedVar "+ protectedVar1);
27     }
28
29     public static void main(String[] args)
30     {
31         AcSpExp obj=new AcSpExp();
32         AccessImpExp obj1=new AccessImpExp();
33
34         obj1.show();
35         System.out.println("defaultVar "+ obj.defaultVar1);
36         System.out.println("privateVar "+ obj1.privateVar);
37         System.out.println("publicVar "+ obj1.publicVar);
38         System.out.println("protectedVar "+ obj1.protectedVar);
39
40         /* AllVariablesExp obj=new AllVariablesExp();
41         System.out.println("defaultVar "+ obj1.defaultVar);
42         System.out.println("privateVar "+ obj1.privateVar);
43         System.out.println("publicVar "+ obj1.publicVar);
44         System.out.println("protectedVar "+ obj1.protectedVar);AcSpExp
45
46     }
47

```

हमने सभी statement को uncomment किया तो केवल public ही accessible है बाकी के सामने error available हो गया है default variable केवल आपने package में ही accessible होता है जैसे ही किसी दूसरे package में जाते हैं यह चलता नहीं है इस कारण यह चल नहीं रहा हम इसे commented कर देते हैं इसी तरह से protected variable को भी किसी दूसरे package में use नहीं कर सकते हैं इसीलिए error available हो जाता है हम इसे किसी दूसरे package में subclass बनाकर use कर सकते हैं इस तरह से हम इन variables की accessibility के according package में use कर सकते हैं public variables का use हम कही भी कर सकते हैं वह कही भी accessible है उसे directly use कर सकते हैं इस तरह हम access specifier का use कर सकते हैं

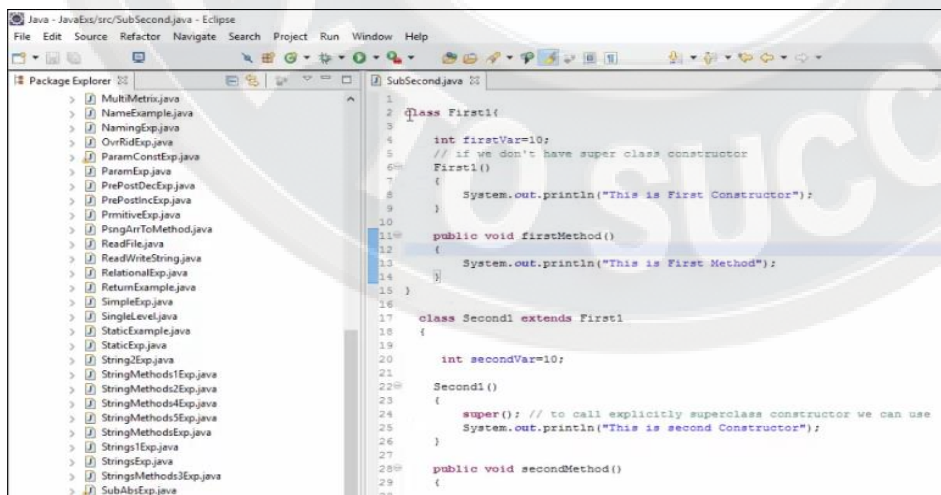
“super” keyword in java

- **“super”** keyword is a reference variable which is used to called its immediate parent class object.
- That means with super we can call immediate parent class instance variable/method/class constructor.

SubSecond.java

SubSecond1.java

अब हम super keyword के बारे में जानेगे यदि हमे कभी आपनी immediate super class के किसी variable को use करना है उसके किसी method को use करना है या किसी constructor का use करना है तो हम super keyword का use कर सकते है यह बहुत ही important keyword रहता है



```

1
2 class First1{
3
4     int firstVar=10;
5     // if we don't have super class constructor
6     First1()
7     {
8         System.out.println("This is First Constructor");
9     }
10
11     public void firstMethod()
12     {
13         System.out.println("This is First Method");
14     }
15 }
16
17 class Second1 extends First1
18 {
19
20     int secondVar=10;
21
22     Second1()
23     {
24         super(); // to call explicitly superclass constructor we can use
25         System.out.println("This is second Constructor");
26     }
27
28     public void secondMethod()
29     {
30
31

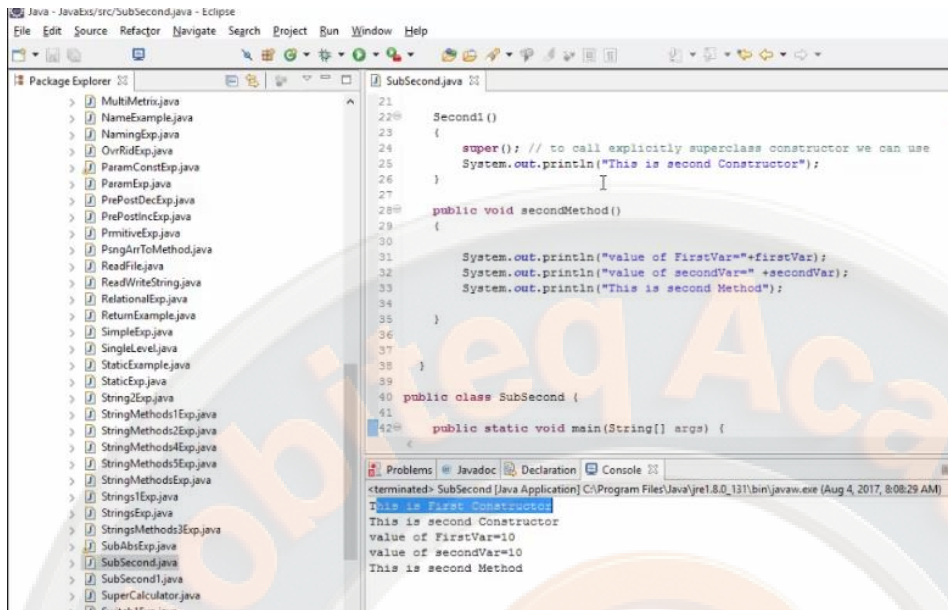
```

```

30
31     System.out.println("value of FirstVar="+firstVar);
32     System.out.println("value of secondVar="+secondVar);
33     System.out.println("This is second Method");
34
35 }
36
37
38 }
39
40 public class SubSecond {
41
42     public static void main(String[] args) {
43         Second1 obj=new Second1();
44         obj.secondMethod();
45
46
47     }
48
49 }
50

```

हमने सबसे पहले class create की है इसमें एक variable create किया है जिसमें 10 value assign की है फिर first1 class का एक constructor first1 है जिसमें हमने एक message represent किया है इसी तरह first class के method firstmethod में एक message represent किया है यहा तक class1 st है इसके बाद second1 class में जाते है जोकि first1 की subclass है इसमें उसका अपना variable, constructor और method है फिर हमने एक तीसरी class भी डिफाइन की है लेकिन हम second 1 class के बारे में जान लेते है हमने उसमें एक variable declare किया है फिर एक constructor भी है अगर हम first1 का constructor भी use करना चाहते है तो हम super का use कर सकते है यह हमारी 1st line होनी चाहिए तो यह first1 constructor को execute करेगा और फिर second class के message को print करेगा उसके बाद हमने print करवाया है फिर हमारे पास subsecond class available है इसमें static void main में second1 का एक object create किया है फिर इसके साथ second method को call किया है हमने object create किया है तो automatically constructor execute हो जाएगा और method को call किया है तो वह भी execute हो जाएगा



```
21
22 Secondi ()
23 {
24     super(); // to call explicitly superclass constructor we can use
25     System.out.println("This is second Constructor");
26 }
27
28 public void secondMethod()
29 {
30
31     System.out.println("value of FirstVar="+firstVar);
32     System.out.println("value of secondVar=" +secondVar);
33     System.out.println("This is second Method");
34 }
35
36 }
37
38 }
39
40 public class SubSecond {
41
42     public static void main(String[] args) {
```

<terminated> SubSecond [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 4, 2017, 8:08:29 AM)

This is second Constructor
value of FirstVar=10
value of secondVar=10
This is second Method

हमने program को run किया तो हमारे पास output available है इस तरह से super keyword का use किया जा सकता है

Overriding methods in java

A Sub class inherits a members of its Super class, then we can redefine methods of super class in subclass , this is called overriding and be remember that method should not be marked final to override.

Why we override methods , some times sub class requires to modify the existing method code which is given in super class. Than we can easily modify the super class method in subclass using overriding.

OvrRidExp.java

```
class SuperClass
{
    void show()

    {
        System.out.println("SuperClass")
    }
}
```

```
class SubClass extends SuperClass
{
    void show()

    {
        System.out.println("SubClass")
    }
}
```

अब हम overriding method के बारे में जानेगे जब हम किसी super class के method को दोबारा subclass में redefined करे तो इसे हम overriding कहते हैं जैसे की example में show भी किया जा रहा है इसमें void show method को दोनों class में use किया गया है जब हम super class में किसी method को डिफाइन करते हैं तो subclass में उसी method को redefined इसीलिए किया जाए कि subclass की अलग requirement हो इसलिए हम इस overriding process का use करते हैं

Rules for Overriding methods

1. If there is argument list , then it should be exactly the same as that of the superclass method.
2. The return type should be the same in sub and superclass.
3. The access level cannot be more restrictive than the overridden method's access level. For example: If the superclass method is declared public then the overriding method in the sub class cannot be either private or protected.
4. Constructors cannot be overridden
5. Only subclasses can override methods of superclasses.
6. If we declare any method with final keyword than it cannot be overridden.
7. static method cannot be overridden but we can re-declare in subclass.
8. Method which we cannot inherit, cannot be overridden by subclass.

OvrRidExp.java

इसके कुछ basic rules है जिनके बारे में जानकारी होना जरूरी है

```

1 class SuperOne
2 {
3     int a=10;
4     int b=20;
5
6     public void showVar()
7     {
8         System.out.println("value of a = "+a);
9         System.out.println("value of a = "+b);
10    }
11
12 }
13
14
15 class SubOne extends SuperOne
16 {
17     int z=30;
18     public void showVar() // overriding super class method at here
19     {
20         // super.showVar();
21
22         System.out.println("value of a = "+a);
23         System.out.println("value of a = "+b);
24         System.out.println("value of a = "+z);
25     }
26
27 }
28
29
30 }
31
32 public class OvrRidExp {
33
34
35
36     public static void main(String[] args) {
37         SubOne obj=new SubOne();
38         obj.showVar();
39     }
40
41 }
42

```

हमने यहा पर superone class डिफाइन की है इसमें हमने a और b दो variables लिए है और showvar method से दोनों को print करवा किया है इसी तरह से subone class है जिसे superone से extends किया है इसमें int z =30 लिया है फिर हमने showvar method को use किया है इसके बाद हम override कर रहे है हमने ऊपर a और b को print करवाया था लेकिन अब हम z को भी print करवाना चाहते है तो हमने तीनों को print करवाया है इस तरह हमने showvar method को subclass में override कर लिया है उसके बाद हमने एक ovrridexp class ली है और subone class का obj create किया है और उस object से showvar method को call किया है हम run करे तो हमारे पास output available है इस तरह से हम override का use कर सकते है

Difference B/W Overriding And Overloading

Method Overloading	Method Overriding
We use method overloading to provide easiness and better readability.	1 A Sub class inherits a members of its Super class, then we can redefine methods of super class in subclass , this is called overriding
It is implemented in one class and same class.	2 We implement Method overriding super and sub classes inheritance.
In case of method overloading, parameters are different in method overloading to implement it .	3 In case of method overriding, parameter are same.
By changing return type of the method we can not implement it because Return type can be same or different in method overloading.	4 We need to keep same data Return type to implement method overriding.

यहा पर overriding और overloading method के बीच difference के बारे में बताया गया है

Final Keyword In Java

- **"final"** is a non-access modifier can be implemented **on a variable/ a method /a class**. We can restrict the user for accessibility implementation of a **variable/ a method /a class** with **final keyword**.
- Final can be: variable / method /class
 - ▣ `final int var=10;`
 - Final variable once assigned a value can never be changed. It will be constant.
 - ▣ `final void show(){}`
 - We cannot override final method in java.
 - ▣ `final class Employee{}`
 - We cannot extend final class in java.

FinalVarMethod.java

अब हम final keyword के बारे में जानेगे यह एक non access modifier है इससे किसी variable, method, और class के access mode को change कर सकते है इसका मतलब है कि restriction बढ़ा सकते है जैसे की यदि हम किसी final को variable के साथ use करे तो ऐसे case में variable की value को change नहीं कर पाएगे value fixed रहेगी यह पुरे program में constant value रहेगी इसी तरह अगर किसी method के साथ final को use करे तो उस case में उस method को किसी subclass में override नहीं कर पाएगे अगर हम final का use किसी class के साथ करे तो हम उस class की subclass नहीं बना पाएगे इस तरह से यह तीन use है

```

1 final class FinalSuper {
2     |
3     final int var=10;
4     |
5     final void show()
6     {
7         |
8         System.out.println("We are in final show method");
9     }
10    |
11    void change()
12    {
13        // var=20;
14    }
15    |
16 }
17 public class FinalVarMethod extends FinalSuper {
18     |
19     |
20     /*
21     void show()
22     {
23         System.out.println("Overriding method ");
24     }
25     */
26     |
27     /*
28     public static void main(String[] args) {
29         FinalVarMethod obj=new FinalVarMethod();
30         //obj.var=30;

```

Console Output:
<terminated> OvrRidExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 4, 2017, 9:13:29 AM)
value of a = 10

हमने finalsuper एक class create की है इसमें एक variable और एक method create किया है और एक void change simple method है उसके बाद जब हम इसकी subclass बनाते है तो हमारे पास error available हो जाता है इसी तरह अगर किसी method को redefined करते है तो error available हो जाता है अगर हम obj के साथ show को call करते है तो हमारे पास error नही आता है अगर हम कोई value change करते है तो error available हो जाता है इस तरह से हमने तीनों के बारे में जानकारी ली है और साथ ही final keyword का use किया है

“this” keyword in java

- We can refer current class object with "this" keyword.
- We can invoke and call current class method and current class constructor.
- We can use "this" as an argument in the method and as argument in the constructor call.
- We can get current object back from any method if we set the return datatype of current class of that method.

```
int x;  
void setValue(int x)  
{  
    // x=x;  
    this.x=x;  
}
```

अब हम this keyword के बारे में जानेगे this का use करते हुए current class के object को denote कर सकते है

हम किसी current class के method और constructor को invoke कर सकते है

this को हम as an argument भी किसी method में पास कर सकते है

हम इससे किसी current method के object को अर्थात this को वापस भी कर सकते है

```

1 public class ThisExp {
2
3     int x;
4
5     void setValue(int x)
6     {
7         x=x;
8         //use this keyword
9         // this.x=x;
10    }
11
12    void showValue( )
13    {
14        System.out.println(" value of x = "+x);
15    }
16
17    public static void main(String[] args)
18    {
19        ThisExp obj=new ThisExp();
20        obj.setValue(10);
21        obj.showValue ();
22    }
23
24
25
26
27
28
29
30

```

Problems | Javadoc | Declaration | Console

<terminated> ThisExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 4, 2017, 10:33:44 AM)

value of x = 0

हमने एक class डिफाइन की है उसके बाद एक variable x डिफाइन किया है उसके बाद एक method डिफाइन किया है इस method के द्वारा हमने ऊपर variable की value को उस particular class में डिफाइन करना चाहते हैं यानि की जो इस x को value मिलेगी वह हम thisexp के x को देना चाहते हैं फिर void showvalue में x को print करवाया है फिर static void main में एक obj create किया है फिर set value में 10 डाल दिया है फिर object के साथ showvalue को call किया है हमने run किया तो हमारे पास output 0 available हुआ है क्योंकि compiler confuse हो गया है कि x कौनसा है इसको denote नहीं कर रहा है इस कारण x में कोई value assign नहीं है इसलिए default value print हो गई है

```

1 public class ThisExp {
2
3     int x;
4
5     void setValue(int x)
6     {
7         // x=x;
8         //use this keyword
9         this.x=x;
10    }
11
12
13    void showValue( )
14    {
15        System.out.println(" value of x = "+x);
16    }
17
18    public static void main(String[] args)
19    {
20
21        ThisExp obj=new ThisExp();
22        obj.setValue(10);
23        obj.showValue();
24    }
25 }
26
27
28

```

<terminated> ThisExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 4, 2017, 10:36:14 AM)
 value of x = 10

अगर हम this keyword का use करते हैं जैसे कि हमने this.x=x का use करते हैं इससे हमने यह बता दिया कि x data member है और = के बाद का x local variable है अगर अब run करे तो output 10 available है इस तरह से this का use कर सकते हैं

Get And Set Methods

- To secure data in java we can provide one more level of option in the form of Get and Set methods.
- These are not specific methods but are user created methods.
- We use set method to setup the value of variables.
- We use get method to get back value of variables.
- By using get/set methods user cannot directly interact with variables of any class and data is always secured.

GetAndSetExp.java

अब हम get और set method के बारे में जानेगे

```

1  class BankData {
2      private String name;
3      private double bal;
4
5      public void display() {
6          System.out.println("Customer name is " + getCustomerName());
7          System.out.println("    Balance is " + getCustomerBalance());
8      }
9
10     public void setCustomerName( String name ) {
11         this.name = name;
12     }
13
14     public void setCustomerBalance( double bal ) {
15         this.bal = bal;
16     }
17
18     public String getCustomerName () {
19         return this.name;
20     }
21
22     public double getCustomerBalance () {
23         return this.bal;
24     }
25 }
26
27 }
28
29 public class GetAndSetExp {
30     public static void main(String[] args) {
31         String name="Sachin";
32         double bal=1874.00;
33         BankData obj=new BankData();
34         obj.setCustomerName(name);
35         obj.setCustomerBalance(bal);
36         obj.display();
37     }
38 }
39
40

```

terminated: ThisExp [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (Aug 4, 2017, 10:36:14 AM)
value of x = 1874

हमने यहा पर एक class डिफाइन की है इसमें दो variable डिफाइन किए है जिसमे name और bal को store करेगे value को store करने के लिए set method का use किया है इसमें हम जिस name को पास करते है उसको class के name variable में store किया जाता है this.name की सहायता से किया जाता है इसी तरह customer balance में भी किया है अगर हम इस data को वापस चाहते है तो get method का use करेगे दोनों का return datatype है यहा पर name और balance वापस आने वाला है जोकि this के साथ है इसकी value को हमने public void display में use किया है हमने directly use किया है उसके बाद एक class ओर डिफाइन की है उसमे string लिया है उसमे नाम sachin दिया है और एक double variable लिया है जिसका नाम bal है उसमे value है फिर ऊपर वाली class का एक object बनाया है उसके साथ customer name और फिर bal को डिफाइन किया है फिर display method को call किया है हम run करे तो हमारे पास output available हो जाएगा इस तरह set और get का use कर सकते है और आपने data को secure कर सकते है

Abstract Methods and Abstract Class

- ❑ An abstract class is a some type of template for other classes, and most commonly used for inheritance , so that sub class can follow the template set by abstract class.
- ❑ Other classes which inherits abstract class have to follow implementation of methods given in abstract classes , although other classes or sub classes of abstract class can add other methods also to define their functionality.
- ❑ An abstract class is a class which is declared using abstract method.
- ❑ An abstract class can or can not have abstract methods.
- ❑ We cannot create objects of Abstract classes.
- ❑ We can create subclasses of Abstract classes.

Calculator.java

```

abstract class Calculator
{
    abstract void addData();
    abstract void subtractData();
    public void showData(int x)
}

public class MyCalculator extends Calculator
{
    void showData()
    {
    }
    void addData()
    {
    }
    void subtractData()
    {
    }
    void multiplyData()
    {
    }
}

```

अब हम abstract method और abstract classes के बारे में जानेगे abstract classes जब भी हम subclass के लिए कोई rule डिफाइन करना चाहते है तो abstract classes से कर सकते है example में हमने abstract class calculator create की है जिसमे दो method भी है हम class के आगे abstract लगाकर class को abstract और method के आगे abstract लगाकर method को abstract बना सकते है फिर हमने एक class और create की है जिसे हमने calculator class से extends किया है और हमने फिर उसके method को भी override किया है यह जरूरी होता है इसके बाद इसके आपने method भी है इस तरह से कुछ rules को follow करते है इस तरह एक rule को डिफाइन किया जाता

है की जोभी class इसकी subclass बने उसके method को override करते हुए आगे बड़े abstract classes का object बनाकर use नहीं कर सकते है अगर हम calculator का object create करे तो error available हो जाएगा हम abstract classes की subclass बना सकते है जैसे की हमने mycalculator class create की है

Abstract Methods and Classes

- Abstract class in Java
 - We can declare abstract class by adding “abstract” keyword. Abstract class needs to be inherited by extending it. It cannot be instantiated.


```
abstract class Calculator
{
}
```
- Abstract method
 - A method that is declared with “abstract” keyword and it does not have any coding in it and is used by subclasses to override it.


```
abstract void showData();
```
- Abstract class can have its own variable/ normal methods/ abstract methods and constructor.

SubAbsExp.java

एक abstract class बनाने के लिए abstract keyword का use करना है हम इसके object नहीं बना सकते हैं इसे inherit किया जा सकता है इसी तरह abstract method बनाना चाहते हैं उसके लिए भी abstract keyword का use करेंगे यह लगाने के साथ ही method जरूरी हो जाएगा उस class को inherit करना जो class इस abstract class को inherit करती है abstract class के पास आपने variables, method, abstract methods और constructor सब हो सकते हैं इसे पूरी एक class की तरह use कर सकते हैं लेकिन इसे object बनाकर use नहीं कर सकते हैं

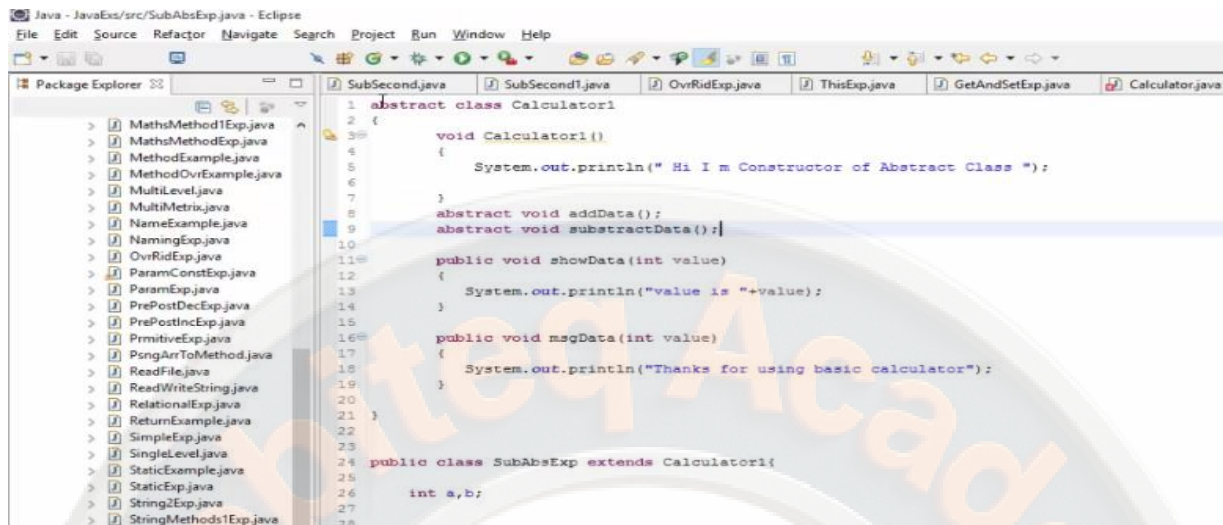
```

1 abstract class Calculator
2 {
3
4     abstract void addData();
5     abstract void subtractData();
6
7     public void showData(int value)
8     {
9         System.out.println("value is "+value);
10    }
11
12    public void magData(int value)
13    {
14        System.out.println("Thanks for using basic calculator");
15    }
16
17    public static void main(String[] args)
18    {
19        Calculator obj=new Calculator();
20        obj.addData();
21        obj.subtractData();
22    }
23
24
25

```

Problems | Javadoc | Declaration | Console

<terminated> GetAndSetExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 4, 2017, 10:50:01 Z)
Customer name is Sachin and Balance is 18874.0



```

1  abstract class Calculator1
2  {
3      void Calculator1()
4      {
5          System.out.println(" Hi I m Constructor of Abstract Class ");
6      }
7      }
8      abstract void addData();
9      abstract void subtractData();
10     }
11     public void showData(int value)
12     {
13         System.out.println("value is "+value);
14     }
15     public void msgData(int value)
16     {
17         System.out.println("Thanks for using basic calculator");
18     }
19     }
20     }
21     }
22     }
23     }
24     public class SubAbsExp extends Calculator1{
25     int a,b;
26     }
27     }
28     }

```

यहाँ एक abstract class available है फिर हमने दो methods abstract डिफाइन किए हैं फिर नीचे एक show data method use किया है इसमें हमने value को print भी करवाया है फिर एक ओर method msgdata के नाम से use किया है इस तरह हमारे पास चार methods available हैं इसमें से दो abstract method हैं और दो body के साथ properly डिफाइन हैं जिसमें एक में value को print करवाया है और दूसरे में एक message को print करवाया है फिर हमने public static void main में हमने एक class की तरह simple object create किया है फिर adddata और subtract data methods को call किया है अगर हम इसे run करें तो हमारे पास error available होगा क्योंकि हम abstract class का object नहीं बना सकते हैं इस कारण error available हो गया है इस तरह हम abstract class का object create नहीं कर सकते हैं



```

29  void addData ()
30  {
31      a=10;
32      b=20;
33      showData (a+b);
34  }
35
36  void substractData ()
37  {
38      a=10;
39      b=20;
40      showData (b-a);
41  }
42
43  public static void main (String [] args)
44  {
45      SubAbeExp obj=new SubAbeExp ();
46      obj.addData ();
47      obj.substractData ();
48  }
49
50
51
52
53

```

हमने यहा पर एक ओर example लिया है इसमें हमने एक class डिफाइन की है इसमें फिर एक constructor डिफाइन किया है इसका मतलब की abstract class का constructor हो सकता है इसके बाद दो abstract method भी डिफाइन किए है यहा जरूरी है कि जो subclass बनेगी उसे इन दोनों methods को override करना पड़ेगा इसके बाद दो method ओर डिफाइन किए है जिसमे एक में value को और दुसरे में message को print करवाया गया है हमने क public class subabaexp ली है जिसने calculator1 को extends किया है इसके बाद दो variables int a और b लिए है अब हमे calculator1 class के दोनों abstract methods को subclass में override करना होगा इसके लिए हमें adddata और substract data दोनों methods को override किया है फिर static void main में subclass का object create किया है फिर adddata और substract data को call किया है अगर हम run करे तो हमारे पास output available हो गया है इस तरह से abstract class का use कर सकते है

Interface and use of interfaces

- An interface is a some type of template for other classes, and most commonly used for inheritance , so that sub class can follow the template set by interface.
- Interface seems like any other class but it is not a class, it is a interface .
- An interface can have only abstract methods with no signature, no body just declaration.
- Other classes which implement interface have to follow implementation of abstract methods declared in interface definition, although other classes or sub classes of interfaces can add other methods also to define their functionality.
- All the methods are public abstract by default in interfaces.
- When we declare Variables ,they declared in an interface are public, static & final by default.
- A class can implement more than one interfaces.
- If a class implements one or more interface, all methods defined by that interface must be used by its subclass to compile successfully.

IntClassExp.java

```
Interface Calculator
{
    void addData();
    void subtractData();
}
```

```
public class MyCalculator implements Interface
{
    void showData()
    {
        void addData()
        {
            void subtractData()
            {
            }
        }
    }
}
```

IntClass1 Exp.java

अब हम interface बारे में जानेगे जैसे की हमने अभी abstract classes के बारे में जाना था वैसा ही concept interface भी है यहा पर हमारे पास example में interface calculator एक class है इसमें दो methods है जो भी class इस class को implement करेगी उसे इन दोनों methods को override करना जरूरी है इस तरह से एक rule create हो गया है इस तरह interface किसी दूसरी class के लिए एक template की तरह है जोकि inheritance के लिए ही use होने वाला है

interface दिखने में एक class की तरह है लेकिन यह class की तरह नहीं है क्योंकि जब इसमें methods डिफाइन करते है तो उसकी body डिफाइन नहीं रहती है इस कारण यह एक rule डिफाइन करता है

interface के सभी methods abstract methods होते है इसका मतलब की उन्हें declare कर सकते है और उन्हें override करना जरूरी है उसकी body नहीं होती है केवल declaration होती है उसको डिफाइन नहीं किया जाता है जैसे adddata method create तो किया गया है लेकिन इसका use नीचे किया गया है

जो भी class interface को implement करेगी उसे सभी methods को override करना होगा एक sub class के आपने methods हो सकते है

interface में जो variables होते है वह सभी default रूप में public static और final रहते है यह तीनों features interface के variables के साथ जुडी रहती है

एक class एक से ज्यादा interfaces का use कर सकती है अर्थात एक class के द्वारा एक से ज्यादा interfaces को implement किया जा सकता है ऐसे case में interface के जितने methods होंगे वह सभी class को override करने होंगे

```

1 interface MyInterface
2 {
3     public void addData(int a, int b);
4     public void subtractData(int a, int b);
5 }
6 class IntClassExp implements MyInterface
7 {
8     public void addData(int a ,int b )
9     {
10        System.out.println("add data is "+(a+b));
11    }
12    public void subtractData(int a, int b)
13    {
14        System.out.println("subtraction is "+(b-a) );
15    }
16    public static void main(String arg[])
17    {
18        IntClassExp obj =new IntClassExp();
19        obj.addData(10,20);
20        obj.subtractData(20, 5);
21    }
22 }
23
24
25

```

Problems | Javadoc | Declaration | Console

```

<terminated> IntClassExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 5, 2017, 7:30:44 AM)
add data is 30
subtraction is -15

```

हमने यहा पर एक interface declare किया है जिसके लिए interface keyword का use किया है उसके बाद block में दो methods को डिफाइन किया गया है दोनों में दो parameters भी है यह दोनों default रूप से abstract methods है इसीलिए इन methods को override करना जरूरी है उसके बाद हमने एक class intclassexp create की है जब हम किसी interface का use करना चाहते हैं तो हमे implement keyword का use करना होता है और साथ में interface का नाम देते है जिसको implement करना चाहते है अब दोनों methods को डिफाइन करना जरूरी है इसीलिए हमने दोनों methods को डिफाइन किया है फिर static void main में class का object create किया है हमने adddata को pick किया और साथ में value को पास किया है इसी तरह से subtract data को भी value पास की है अब हमने इसे run किया तो हमारे पास output available है

```

1  Interface basicInterface
2  {
3  }
4  public void addData(int a, int b);
5  public void subtractData(int a, int b);
6  }
7  interface advInterface
8  {
9  }
10 public void multiplyData(int a, int b);
11 public void divideData(int a, int b);
12 }
13 }
14 }
15 }
16 class IntClass1Exp implements basicInterface,advInterface
17 {
18 }
19 public void addData(int a ,int b )
20 {
21     System.out.println("add data is "+(a+b));
22 }
23 public void subtractData(int a, int b)
24 {
25     System.out.println("subtraction is "+(b-a) );
26 }

```

Problems | Javadoc | Declaration | Console

```

<terminated> IntClassExp [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 5, 2017, 7:31:47 AM)
add data is 30
subtraction is -15

```

यहा पर हमारे पास एक ओर example available है इसमें हमने दो interface का use किया है यहा पर basic interface 1st interface है जिसमे दो methods को डिफाइन किया है इसी के साथ हमने advinterface लिया है इसमें भी दो methods को डिफाइन किया है फिर एक intclass1 exp है इसने दोनों interface को implement किया है इसके लिए हमने दोनों के बीच comma का use किया है इस तरह हम एक से ज्यादा interface को use कर सकते है फिर हमने दोनों interface के methods को override किया है फिर static void main में चारो methods को call किया है जब हम इसे run करे तो हमारे पास output available हो गया है इस तरह से एक से ज्यादा interface का use कर सकते है

Difference Between An Interface And An Abstract Class

Abstract class	S.No	Interface
An abstract class is a some type of template for other classes, and most commonly used for inheritance , so that sub class can follow the template set by abstract class.	1	An Interface is a some type of template for other classes, and most commonly used for inheritance , so that sub class can follow the template set by interface. And it is must for subclass to provide all methods provided by Interface.
We use abstract keyword to create abstract classes or abstract method.	2	We use interface keyword to create abstract classes or abstract method.
Abstract class can have non-abstract methods also which are optional for subclass to implement. so in abstract class which methods subclass requires , it can implement (abstract) and which it doesn't want to use , it can leave them.	3	All methods are abstract in Interface and it can not have non-abstract methods. Subclass has to implement all methods of interface. So Interfaces are useful in a situation that all methods provided by interface should be implemented by subclass.
You can extend only one abstract class at a time.	4	You can implement many interfaces at one time.

अब हम interface और abstract class के बीच में difference के बारे में जानेंगे

abstract class एक template की तरह use की जाती है जोकि एक rule को डिफाइन करती है और जोभी class इस class को inherit करती है यानि की जो subclass बनती है उसे सभी abstract methods को override करना जरूरी होगा इसी तरह interface भी एक rule को डिफाइन करता है इसमें जो template डिफाइन किया जाता है उसमें किसी भी method का body डिफाइन नहीं किया जाता है

abstract class create करने के लिए abstract keyword का use किया जाता है इसी तरह interface के लिए interface keyword का use किया जाता है

abstract class के पास non abstract methods हो सकते है इसी तरह interface में non abstract methods नहीं होते है इसमें केवल abstract method ही रहते है इसमें सभी methods को override करना जरूरी होता है abstract class में एक ही class को inherit कर सकते है उसके लिए extends keyword का use किया जाता है इससे एक class को inherit कर सकते है परन्तु interface में एक से ज्यादा interface को implement कर सकते है और सभी interface के methods को inherit करना भी

जरूरी होगा इसके लिए हम implement keyword का use करते हैं इस तरह से दोनों में difference रहते हैं और दोनों का use किया जा सकता है

Try Catch In Java To Find Exceptions

□ What is exception ?

- Lets take example of dividing a number /0 , what will you expect to get. It will give an exception.

- We can cover expected exceptions using try{catch} block

```
a=10;
c=a/b;
```



```
Output:
java.lang.ArithmeticException: / by zero
at ExcpExp1.show(ExcpExp1.java:8)
at ExcpExp1.main(ExcpExp1.java:20)
```

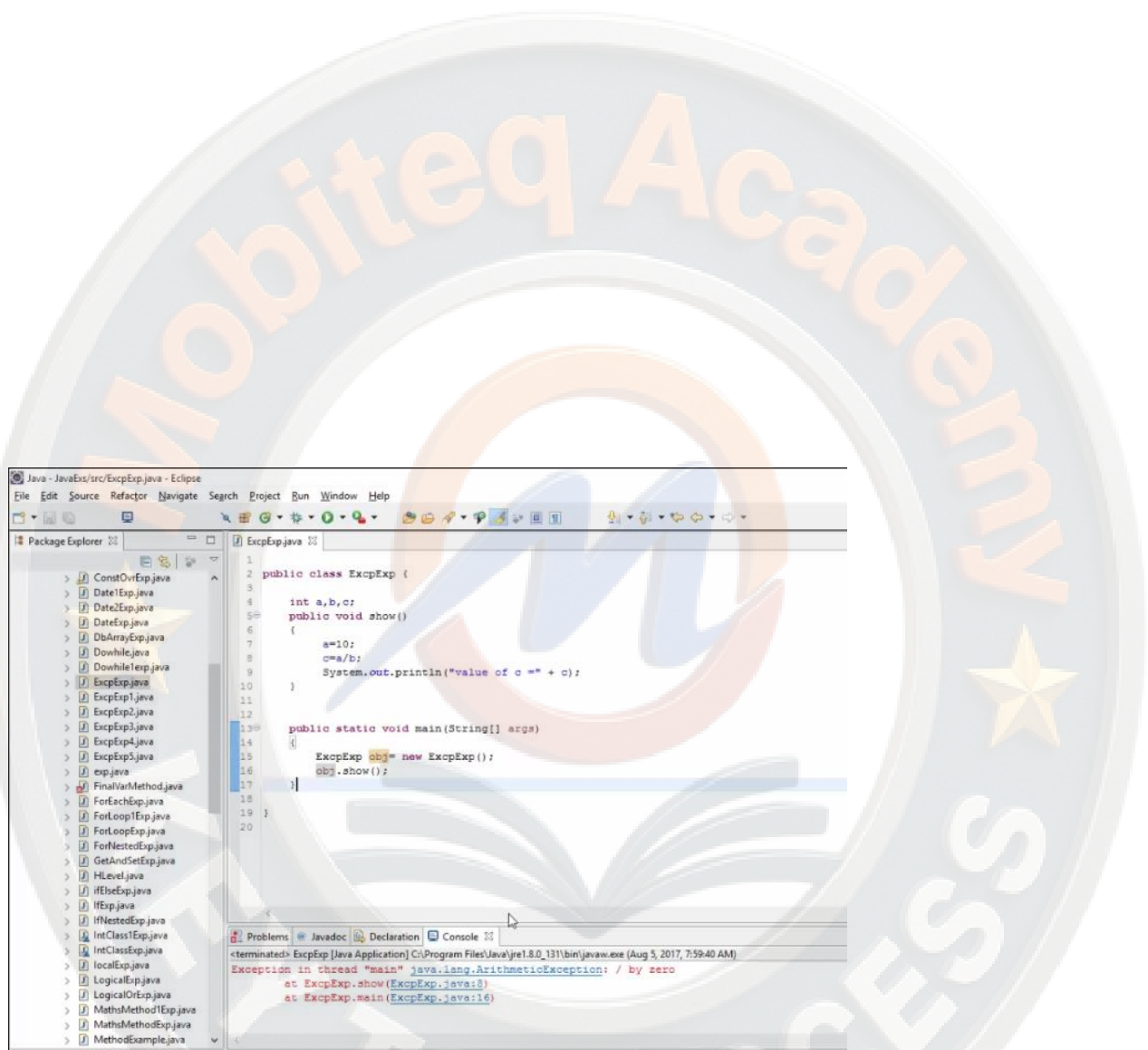
ExcpExp.java

ExcpExp1.java

ExcpExp2.java

ExcpExp3.java

अब हम try और catch के बारे में जानेगे जब हम कुछ error के साथ कुछ ऐसी चीजो के साथ जो program को चलने नहीं देते हैं जब हमारे पास ऐसा कुछ होता है तो उन चीजो को दूर करने के लिए try catch का use कर सकते हैं इसके लिए हम के prectical example को देख लेते हैं



यहा पर हमने एक simple program को लिया है जिसमे एक class को डिफाइन किया है फिर तीन variables को declare किया है फिर एक method का use किया है फिर static void main में object create किया है और method को call किया है अब हम इसे run करते है तो हमारे पास exception के रूप में error available हो गया है हम ने b में कोई value store नहीं की है और हम 0 से किसी value को भाग नहीं कर सकते है इस तरह कुछ input user द्वारा भी डाला जा सकता है इसीलिए इस तरह के error के कारण program चलना बंद हो जाता है ऐसे case में try और catch का use कर सकते है

Try Catch In Java To Find Exceptions

□ What is exception ?

- Lets take example of dividing a number /0 , what will you expect to get. It will give an exception.

```
a=10;
c=a/b;
```



```
Output:
java.lang.ArithmeticException: / by zero
at ExcpExp1.show(ExcpExp1.java:8)
at ExcpExp1.main(ExcpExp1.java:20)
```

- We can cover expected exceptions using `try{}catch{} block`

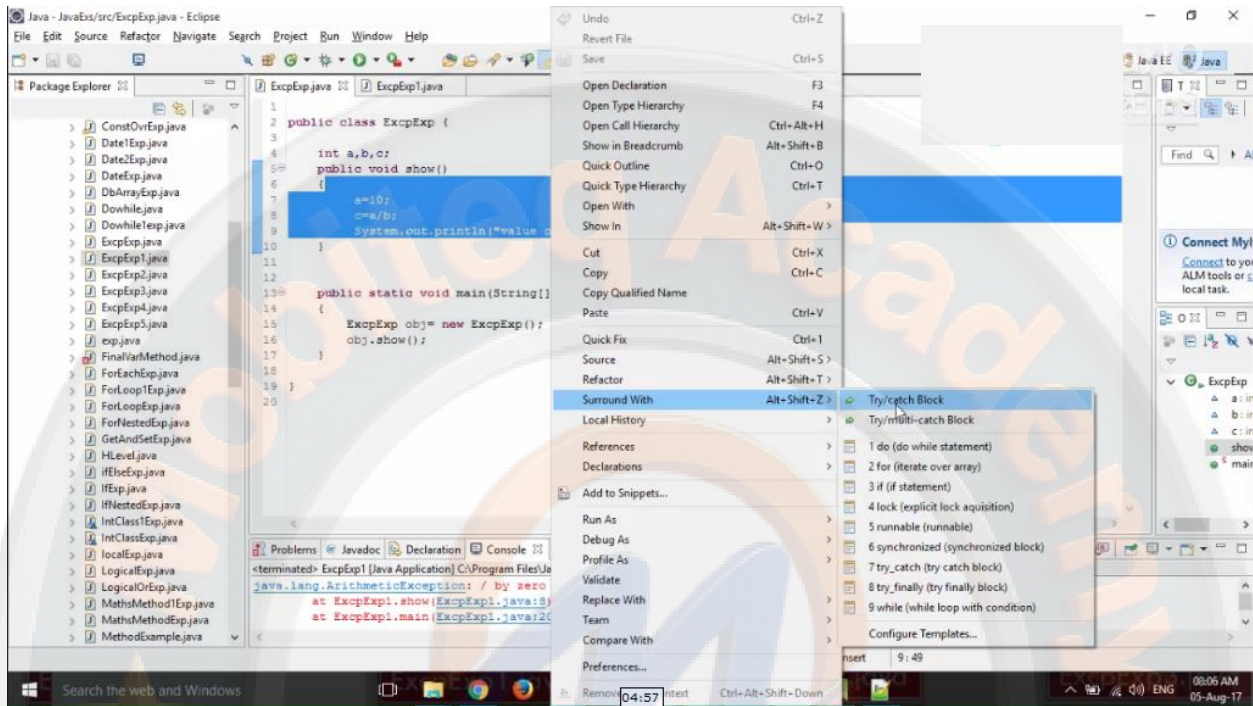
ExcpExp.java

ExcpExp1.java

ExcpExp2.java

ExcpExp3.java

try और catch दो block है जिनका use करके हम exception को overcom कर सकते है इस तरह इन exception से deal करने के लिए try catch का use कर सकते है ताकि हमारा program चलता रहे अभी तक हमने arithmetic exception के बारे में जाना है



हमने आपने block को select करके right क्लिक करके surround with पर जाकर try/catch block पर क्लिक करे तो हमारी statement try और catch block में आ गई है यहा पर catch में e.print stack trace हमे error डिफाइन करेगा जो error या exception आने वाली है अभी run करे तो भी same output रहता है

```

1 public class ExcpExpl {
2
3     int a,b,c;
4     public void show()
5     {
6         try {
7             a=10;
8             c=a/b;
9             System.out.println("value of c = " + c);
10        } catch (Exception e) {
11            // e.printStackTrace();
12            System.out.println("Program has some error in calculation");
13        }
14    }
15
16    public static void main(String[] args)
17    {
18        ExcpExpl obj= new ExcpExpl();
19        obj.show();
20    }
21 }
22

```

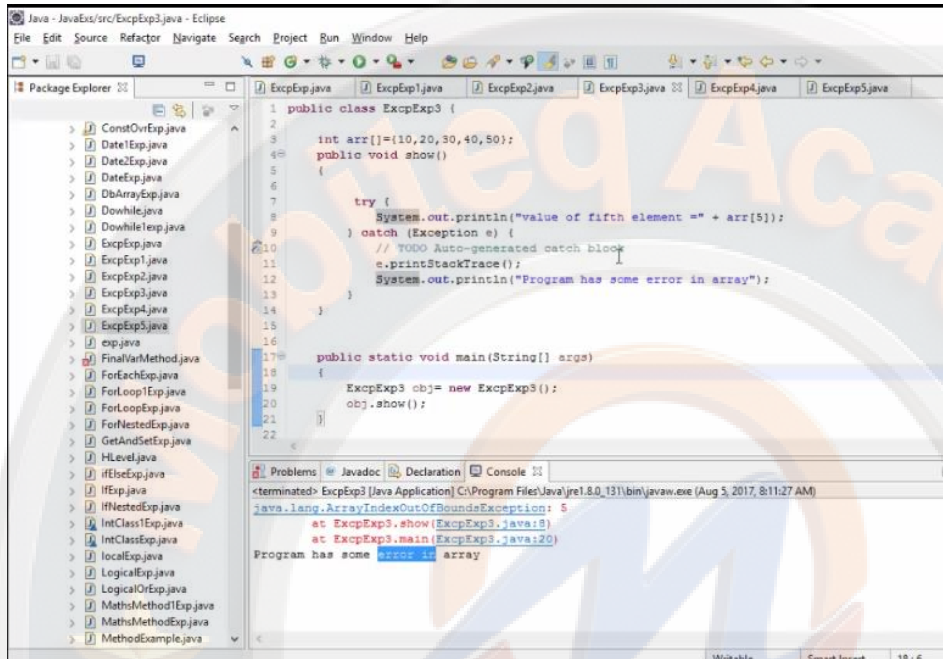
Problems | Javadoc | Declaration | Console

```

<terminated> ExcpExpl [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 5, 2017, 8:07:28 AM)
Program has some error in calculation

```

हम किसी भी exception को आपने message के साथ display कर सकते हैं हमने यहा पर e.printStackTrace method को commented कर दिया है और एक line को add कर लिया है अगर हम run करे तो वह line output में available हो गई है अगर हम काफी सारे try और catch block का use करते है तो ऐसे case में specifically कहाँ पर किस पार्ट में problem आई है उसको हम आसानी से जान सकते है यदि हम इस तरह का message provide करते है इस तरह से try और catch का use करके हम exception को pick कर सकते है



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left lists various Java files. The main editor displays the source code for `ExcpExp3.java`. The code defines a class `ExcpExp3` with a `show()` method that attempts to access an array element at index 5, which is out of bounds. A `try-catch` block is used to handle the exception, printing the stack trace and a message: "Program has some error in array". The `main` method creates an instance of `ExcpExp3` and calls `show()`. The Console window at the bottom shows the output of the program, including the stack trace for `java.lang.ArrayIndexOutOfBoundsException: 5` and the message "Program has some error in array".

```
1 public class ExcpExp3 {
2
3     int arr[]={10,20,30,40,50};
4     public void show()
5     {
6
7         try {
8             System.out.println("value of fifth element =" + arr[5]);
9         } catch (Exception e) {
10            // TODO Auto-generated catch block
11            e.printStackTrace();
12            System.out.println("Program has some error in array");
13        }
14    }
15
16
17    public static void main(String[] args)
18    {
19        ExcpExp3 obj= new ExcpExp3();
20        obj.show();
21    }
22 }
```

Console Output:

```
<terminated> ExcpExp3 [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (Aug 5, 2017, 8:11:27 AM)
java.lang.ArrayIndexOutOfBoundsException: 5
    at ExcpExp3.show(ExcpExp3.java:9)
    at ExcpExp3.main(ExcpExp3.java:20)
Program has some error in array
```

इसी तरह exception के लिए हमने यहा पर array का example भी practical किया है उसमे एक message print किया है और run करने पर error show किया जाता है इस तरह हम पता चल जाएगा की कहाँ पर problem आ रही है इस तरह से बहुत बड़े program में coding को आसान कर सकते है

Rules Using Try Catch In Java

- **What is try {} catch{} Block?**
- The try block is a block used to cover statements where an exception might occur. A try block should have a catch block, which has the code to handle the exception that occurs in its try block.
- If more than one exceptions are expected to occur than we can use multiple catch blocks to handle those exceptions.
- A try block must have at least one catch block or and can have more than once catch block.
- A try block can have one **finally** block .

```
try{
    int arr[]=new int[5];
    arr[5]=20/5;
    System.out.println("We are in try block");
}
catch(ArithmeticException e){
    System.out.println("Exception is ArithmeticException");
}
catch(ArrayIndexOutOfBoundsException e){
    System.out.println("Exception is ArrayIndexOutOfBoundsException");
}
catch(Exception e){
    System.out.println("There is some other issue");
}
```

ExcptExp4.java

अब हम try और catch के rules के बारे में जानेगे try और catch block का use उन exception के लिए किया जाता है जोकि expected रूप में आ सकती है इस तरह try block मुख्य block रहता है जो उस code को occurs करता है जहा पर exception arise हो सकती है जैसे की example में try उन line of codes को block में ले रहा है जहा पर expected exception आ सकती है हर try के साथ एक catch लगा होना जरूरी है जो उस particular try के code में आए हुए exception को pick करता है अब एक से ज्यादा catch हो सकते है जैसे की हमने example में तीन catch का use किया है क्योंकि code में एक से ज्यादा exception आ सकती है इस तरह एक try के साथ एक से ज्यादा exception का use कर सकते है

try में finally block भी लगाया जाता है यदि try में कोई exception arise होता है तो catch block को execute किया जाता है अगर कोई exception arise नहीं होगा तो कोई catch भी execute नहीं होगा परन्तु अगर हम चाहते है की कोई coding चले ही चले तो हम ऐसे में finally block को use कर सकते है

```

2 public class ExcpExp4 {
3     public void show()
4     {
5         try {
6             int arr[] = new int[5];
7             arr[2] = 10/5;
8             System.out.println("Value in array "+arr[2]);
9             System.out.println("We are in try block");
10        }
11        catch (ArithmeticException e) {
12            System.out.println("Exception is ArithmeticException");
13        }
14        catch (ArrayIndexOutOfBoundsException e) {
15            System.out.println("Exception is ArrayIndexOutOfBoundsException");
16        }
17        catch (Exception e) {
18            System.out.println("There is some other issue");
19        }
20        System.out.println("Now We are out of try catch blocks");
21    }
22    public static void main(String args[])
23    {
24        ExcpExp4 obj = new ExcpExp4();
25        obj.show();
26    }
27 }

```

Console Output:

```

Value in array 2
We are in try block
Now We are out of try catch blocks

```

हमने यहा पर exception का एक ओर example लिया है हमने run किया तो हमारे पास output available है जोकि right side में console panel में available है यहा पर कोई भी catch execute नहीं हुआ है क्योंकि कोई भी exception arise नहीं हुई है

```

4 public class excpexp4 {
5     public void show()
6     {
7         try {
8             int arr[] = new int[5];
9             arr[2] = 10/0;
10            System.out.println("Value in array "+arr[2]);
11            System.out.println("We are in try block");
12        }
13        catch (ArithmeticException e) {
14            System.out.println("Exception is ArithmeticException");
15        }
16        catch (ArrayIndexOutOfBoundsException e) {
17            System.out.println("Exception is ArrayIndexOutOfBoundsException");
18        }
19        catch (Exception e) {
20            System.out.println("There is some other issue");
21        }
22        System.out.println("Now We are out of try catch blocks");
23    }
24    public static void main(String args[])
25    {
26        ExcpExp4 obj = new ExcpExp4();
27        obj.show();
28    }
29 }

```

Console Output:

```

Exception is ArithmeticException
Now We are out of try catch blocks

```

हमने program में थोड़ा सा change किया है हमने 10/0 कर दिया है और run किया है तो हमारे पास output में arithmetic exception show की गई है इस तरह आसानी से हम exception को जान सकते हैं

```

> [J] Cons A 2 public class ExcpExp4 {
> [J] Date 3     public void show() {
> [J] Date 4     {
> [J] Date 5         try {
> [J] Date 6             int arr[] = new int[5];
> [J] Date 7             arr[5] = 10/5;
> [J] Date 8             System.out.println("Value in array "+arr[2]);
> [J] Date 9             System.out.println("We are in try block");
> [J] Excp1 10         }
> [J] Excp1 11         catch (ArithmeticException e) {
> [J] Excp1 12             System.out.println("Exception is ArithmeticException");
> [J] Excp1 13         }
> [J] Excp1 14         catch (ArrayIndexOutOfBoundsException e) {
> [J] Excp1 15             System.out.println("Exception is ArrayIndexOutOfBoundsException");
> [J] Excp1 16         }
> [J] Final 17         catch (Exception e) {
> [J] ForEr 18             System.out.println("There is some other issue");
> [J] ForEr 19         }
> [J] ForEr 20     }
> [J] ForEr 21     System.out.println("Now We are out of try catch blocks");
> [J] GetA 22 }
> [J] HLev 23 public static void main(String args[]) {
> [J] HLev 24     ExcpExp4 obj = new ExcpExp4();
> [J] HExp 25     obj.show();
> [J] HRes 26 }
> [J] InfCl 27 }
> [J] InfCl 28 }
> [J] local 29 }
> [J] local 30 }

```

```

<terminated> ExcpExp4 [Java Application] C:\Program Files\Ja
Exception is ArithmeticException
Now We are out of try catch blocks

```

इसी तरह हमने arr [5] = 10/5 किया है और run किया तो हमारे पास output में एक message show किया गया है इस तरह यह खुद ही pick करके उससे related message को show कर देता है इसी तरह अगर हम किसी exception के बारे में पता ही नहीं है तो उस case में last catch execute होगा इसमें हमारे पास हर बार last में एक message print हो रहा है जोकि हमारे program में available है इसका मतलब की हमारे program को flow टूटता नहीं है अगर हम try और catch का use करते हैं इस तरह से try और catch का use कर सकते हैं

Finally Block In Java

- We use Finally block in java to do some finalize task which are important.
- Doesn't matter if any exception has been raises or not , try or catch executed or not, Finally block will always execute.

```
try {
    System.out.println("value of fifth element =" + arr[2]);
}
catch (Exception e)
{
    //e.printStackTrace();
    System.out.println("Program has some error in array");
}
finally
{
    System.out.println("We are in finally block");
}
```

ExcpExp5.java

try और catch के साथ जुड़ा हुआ finally block भी है हमारे पास example में try और catch block available है और finally block को भी introduce किया है यह एक तीसरा block है यह optional है हम चाहे तो use कर सकते है और नही तो हमे use करना कोई जरूरी नही है जब भी finally का use करते है तो हम इसमें वह statement include करते है जो हम execute करवाना ही चाहते है तो ऐसे case में चाहे exception arise या ना हो तो finally block execute होता है तो जो important code हो उसे हम finally का use करके execute कर सकते है

```

2 int arr[]={10,20,30,40,50};
3 public void show()
4 {
5     try {
6         System.out.println("value of fifth element =" + arr[5]);
7     }
8     catch (Exception e)
9     {
10        // TODO Auto-generated catch block
11        //e.printStackTrace();
12        System.out.println("Program has some error in array");
13    }
14    finally
15    {
16        System.out.println("We are in finally block");
17    }
18 }
19
20 public static void main(String[] args)
21 {
22     ExcepExp5 obj= new ExcepExp5();
23     obj.show();
24 }
25
26 }
27
28 }
29
30 }
31
32 }
33
34 }
35
36 }
37
38 }
39
40 }
41
42 }
43
44 }
45
46 }
47
48 }
49
50 }
51
52 }
53
54 }
55
56 }
57
58 }
59
60 }
61
62 }
63
64 }
65
66 }
67
68 }
69
70 }
71
72 }
73
74 }
75
76 }
77
78 }
79
80 }
81
82 }
83
84 }
85
86 }
87
88 }
89
90 }
91
92 }
93
94 }
95
96 }
97
98 }
99
100 }

```

हमने यहा पर finally example को practical किया है हमने यहा पर finally block भी include किया है और program को run किया तो हमारे पास output available है इसमें exception आने के बाद भी finally block execute हुआ है

```

1 public class ExcepExp5 {
2     int arr[]={10,20,30,40,50};
3     public void show()
4     {
5         try {
6             System.out.println("value of fifth element =" + arr[5]);
7         }
8         catch (Exception e)
9         {
10            // TODO Auto-generated catch block
11            //e.printStackTrace();
12            System.out.println("Program has some error in array");
13        }
14        finally
15        {
16            System.out.println("We are in finally block");
17        }
18    }
19
20    public static void main(String[] args)
21    {
22        ExcepExp5 obj= new ExcepExp5();
23        obj.show();
24    }
25
26 }
27
28 }
29
30 }
31
32 }
33
34 }
35
36 }
37
38 }
39
40 }
41
42 }
43
44 }
45
46 }
47
48 }
49
50 }
51
52 }
53
54 }
55
56 }
57
58 }
59
60 }
61
62 }
63
64 }
65
66 }
67
68 }
69
70 }
71
72 }
73
74 }
75
76 }
77
78 }
79
80 }
81
82 }
83
84 }
85
86 }
87
88 }
89
90 }
91
92 }
93
94 }
95
96 }
97
98 }
99
100 }

```

इसी तरह अगर कोई exception नहीं आती है तो भी finally block execute हुआ है इस तरह से finally block का use कर सकते हैं

```

1 import java.io.*;
2
3 public class ReadFile {
4
5     public static void main(String[] args) throws IOException
6     {
7         File f = new File("d://FILE.txt");
8         BufferedReader b = new BufferedReader(new FileReader(f));
9         String s;
10        while ((s=b.readLine())!=null)
11        {
12            System.out.println(s);
13        }
14        b.close();
15    }
16 }
17
18
19 }

```

Output: This is from file streams to my... This is new line added by me in

अभी तक हमने try और catch exception के बारे में जाना है इसके इलावा भी हमारे पास exception से deal करने के लिए throws statement available है हम error को समझने के लिए streams के अंतर्गत एक example को देखते हैं इसके लिए हम readfile का example देखते हैं हमने यहा पर कुछ package को import किया हुआ है फिर एक class create की है इसमें हम एक file को read करना चाहते हैं जोकि d drive पर available है और इस file में जो भी है उसे हमने while loop के साथ print करवा लिया है हमने run किया तो हमारे पास output में file में जो कुछ था वह show किया गया है

```

1 import java.io.*;
2
3 public class ReadFile {
4
5     public static void main(String[] args) throws IOException
6     {
7         File f = new File("d://FILE6.txt");
8         BufferedReader b = new BufferedReader(new FileReader(f));
9         String s;
10        while ((s=b.readLine())!=null)
11        {
12            System.out.println(s);
13        }
14        b.close();
15    }
16 }
17
18
19 }

```

Exception: java.io.FileNotFoundException: d://FILE6.txt (No such file or directory)

हमने यह पर file का नाम file6 कर दिया है जोकि available नहीं है और हमने run किया तो हमारे पास exception available हो गई है ऐसे exception को cover करने के लिए try catch तो available है

परन्तु इसके लिए throws keyword का use कर सकते हैं इसके साथ हम जो unexpected exception को cover कर सकते हैं

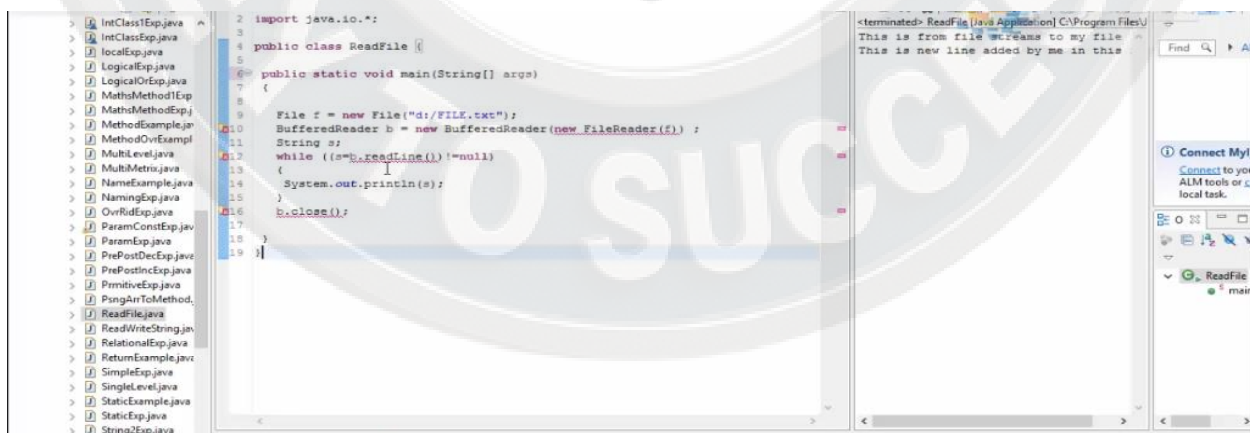
Throws in Java

We can use **throws keyword** to declare an exception. Which is very helpful to programmer to find where exception has occurred and he can easily work out on exception and easy work out on flow of program.

```
public void readFile() throws IOException
{
    File f = new File("d:/MYFILE.txt");
    BufferedReader b = new BufferedReader(new FileReader(f));
    String s;
    while ((s=b.readLine())!=null)
    {
        System.out.println(s);
    }
    b.close();
}
```

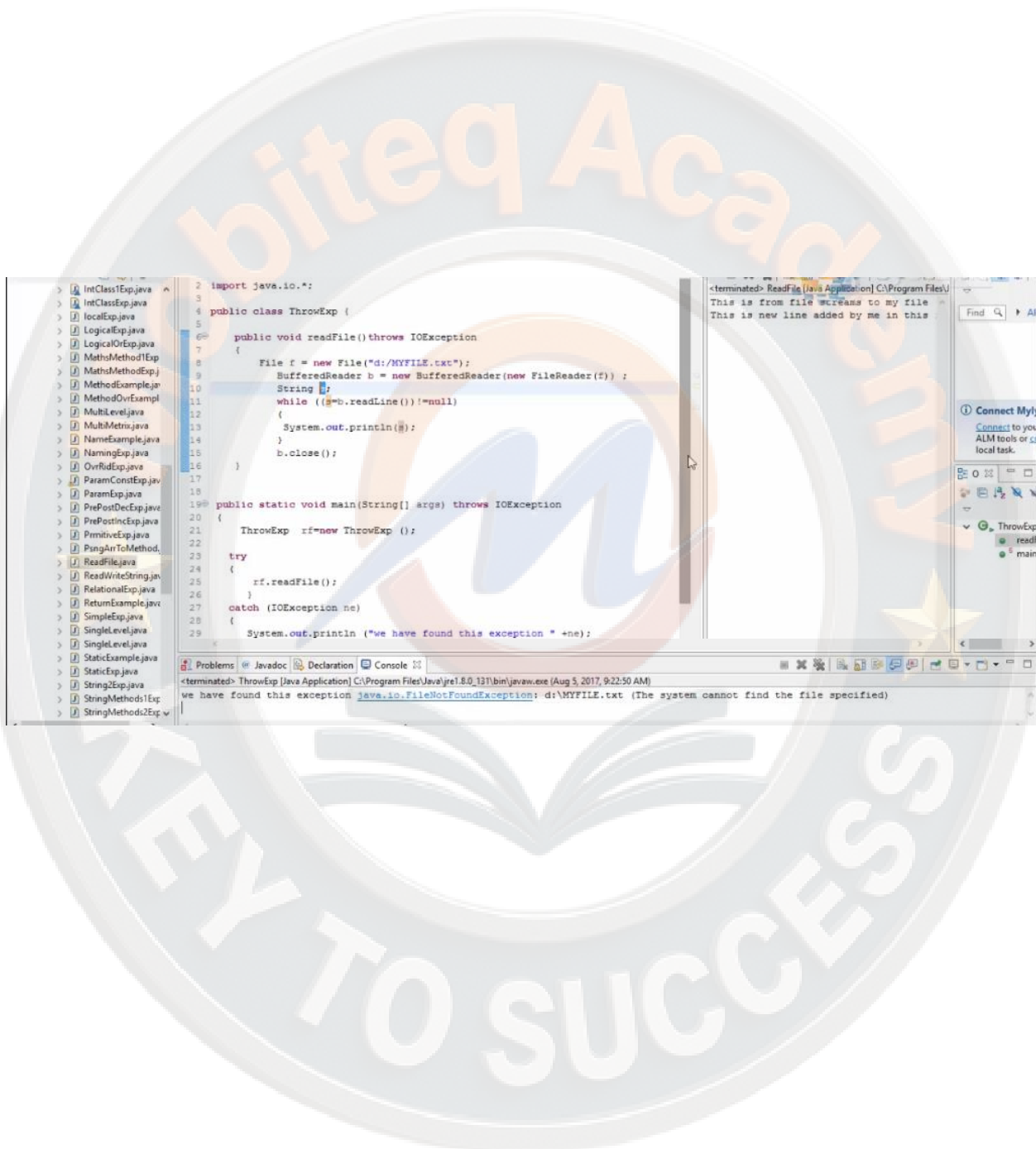
ThrowExp.java

throws एक keyword है जिससे exception को declare कर सकते हैं इससे programmer आसानी से exception को जान सकता है कि किस पार्ट में issue आ रहा है और जिससे उसके program का flow भी चलता रहता है कुछ जगह पर java throws को जरूरी मानता है



जैसे की हमने यहा पर throws exception को हटा दिया है तो हमारे पास error available हो गए हैं हलाकि अभी वह directly program से जुडी हुई भी नहीं है इस तरह java कुछ input output throws

exception को जरूरी रखता है इस तरह हमें input output throws exception को throws करना जरूरी होता है



इसी तरह हमने यहा पर throws के एक ओर example को practical किया है इसमें हमने जो file नहीं है उससे read करने की कोशिश की है अर्थात java को वह file नहीं मिलेगी ऐसे case में input output

exception arise की जाएगी जो throws के द्वारा प्राप्त की जाएगी और catch को दी जाएगी और फिर catch उस प्राप्त की गई exception को print करवाकर show करेगा कि issue कहाँ पर है इस तरह से throws का use exception को pick करने के लिए और उन्हें solve करने के लिए कर सकते हैं

