***dt : 27/7/2022(Day-1)***

***Course Name : Full Stack Java / CoreJava***

***1.CoreJava - Language - 7 AM***

***2.AdvJava - Technology***

***3.Oracle (DataBase) - 6 PM***

***4.HTML/CSS/JS (UI) - 4 PM(3rd Aug - Online)***

***5.Anguler (UI)***

***6.Testing Basics - Selenium (Testing)***

***7.Spring(Core,MVC,JDBC,SpringBoot) - Framework***

***8.WebServices - Framework***

***9.Devops (Tools)***

***Advantages:***

***=>Project Development***

***=>100% placement - (Test - CoreJava,AdvJava and Oracle)***

***=>NASSCOM***

***========================================================***

***Duration : 5 to 6 Months***

***=======================================================***

***CoreJava : 55 to 60 workimg Session***

***======================================================***

***Language:***

***(i)Alphabets***

***(ii)Grammer***

***(iii)Construction Rules***

***Note:***

***=>Every Language will have its own Alphabets,Grammer and***

***Construction Rules.***

***======================================================***

***Part-1 : CoreJava - StandAlone Application***

***Part-2 : AdvJava - Web Application***

***Part-1 : CoreJava***

***1.Programming Components(Java Alphabets)***

***2.Programming Concepts***

***3.Object Oriented Programming features***

***Dt : 28/7/2022(Day-2)***

***1.Programming Components(Java Alphabets)***

***(a)Variables***

***(b)Methods***

***(c)Blocks***

***(d)Constructors***

***(e)Classes***

***(f)Interfaces***

***(g)AbstractClasses***

***2.Programming Concepts***

***(a)Object Oriented Programming***

***(b)Exception Handling process***

***(c)Multi-Threading process***

***(d)Java Collection Framework(JCF)***

***(Data Structures in Java)***

***(e)Files in Java***

***(f)Networking in Java***

***(Communication with TCP/IP)***

***3.Object Oriented Programming features***

***(a)Class***

***(b)Object***

***(c)Abstraction***

***(d)Encapsulation***

***(e)PolyMorphism***

***(f)Inheritance***

***--------------------------------------------------------***

***DT : 29/7/2022(Day-3)***

***Note:***

***=>Using CoreJava Components and Concepts we can develop***

***'StandAlone applications'.***

***faq:***

***define StandAlone Applications?***

***=>The applications which are installed in one computer and***

***performs actions in the same computer are known as StandAlone***

***applications or desktop applications or Windows applications.***

***Note:***

***According to developer StandAlone application means,***

***No HTML input***

***No Server Environment***

***No DataBase Storage***

***=======================================================***

***Part-2 : AdvJava***

***=>AdvJava uses the following three technologies to construct***

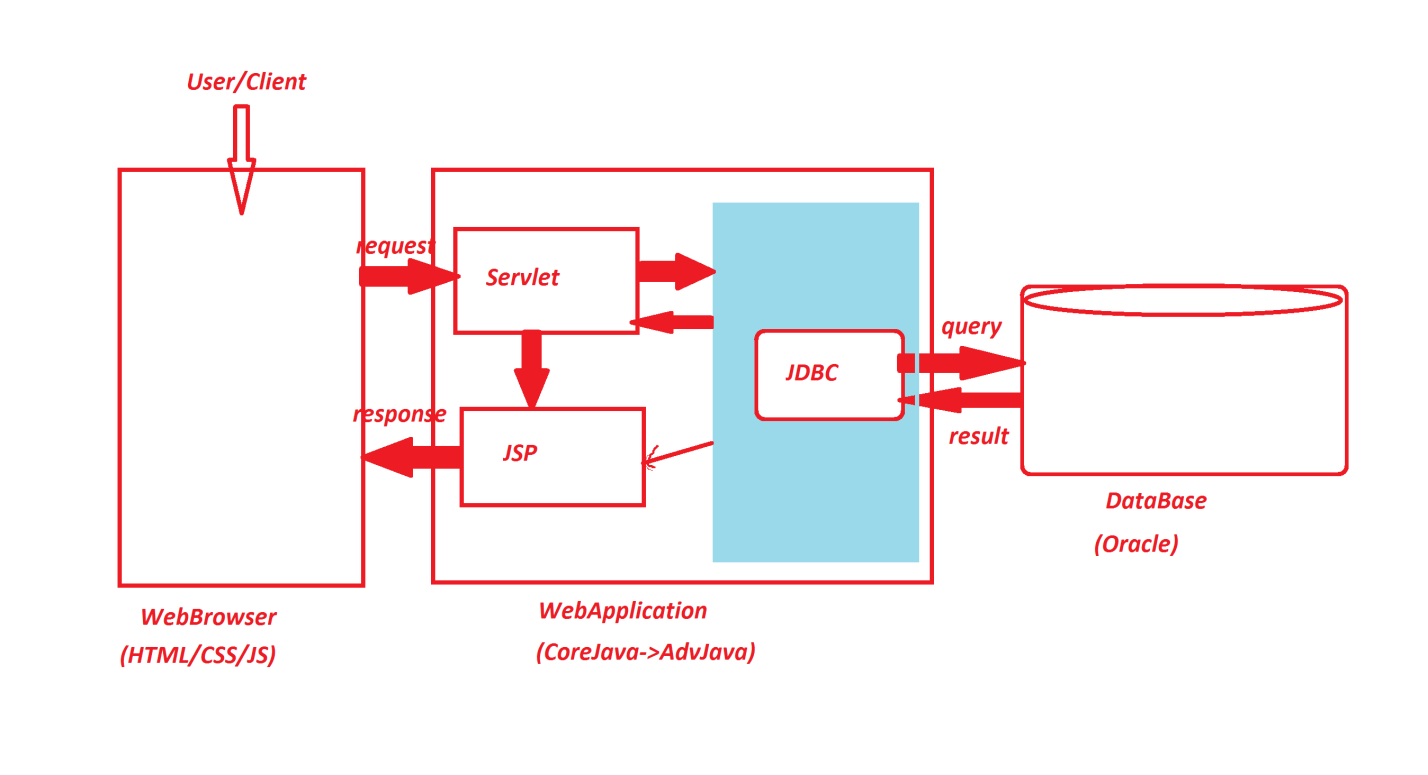
***WebApplications:***

***(a)JDBC***

***(b)Servlet***

***(c)JSP***

***Diagram:***

******

***(a)JDBC:***

***=>JDBC Stands for 'Java DataBase Connectivity' and which is***

***used to establish communication b/w JavaProgram and DataBase***

***product.***

***(b)Servlet:***

***=>Servlet means 'Server program' and which accepts the request***

***from the User or client.***

***(c)JSP:***

***=>JSP stands for 'Java Server page' and which is response***

***from WebApplication.***

***====================================================***

***Dt : 1/8/2022***

***faq:***

***define Web Application?***

***=>The application which is executed in WebEnvironment or***

***Internet Environment is known as WebApplication or Internet***

***Application.***

***===============================================***

***faq:***

***wt is the diff b/w***

***(i)Language***

***(ii)Technology***

***(iii)Framework***

***(i)Language:***

***=>Language will provide programming components and concepts***

***to construct programs.***

***Ex:***

***CoreJava***

***(ii)Technology:***

***=>The process of applying the knowledge to realtime world***

***application development is known as Technology.***

***Ex:***

***AdvJava***

***(iii)Framework:***

***=>The structure which is ready constructed and available for***

***application development is known as Framework.***

***Ex:***

***Spring***

***WebServices***

***=======================================================***

***faq:***

***define Enterprise Applications?***

***=>The applications which are executing in distributed***

***environment and depending on the features like 'Security',***

***'Loadbalancing' and 'Clustering' are known as Enterprise***

***applications or Enterprise Distributed Applications.***

***=======================================================***

***Level-1 : CoreJava***

***Level-2 : AdvJava***

***Level-3 : Spring&WebServices***

***=========================================================***

***Dt : 2/8/2022(Syllabus)***

***\*imp***

***define Program?***

***=>Program is a set-of-Instructions.***

***define Programming?***

***=>The process of writing programs is known as programming,***

***which means converting 'Analysis and Design' into Code form.***

***Project:(SDLC - S/W Development Life Cycle)***

***stage-1 : Analysis***

***stage-2 : Design***

***stage-3 : Coding/Construction***

***stage-4 : Testing***

***stage-5 : Deployment***

***define programmer?***

***=>The person who writes(construct) the programs is known as***

***Programmer or Code writer(Developer).***

***------------------------------------------------------***

***Note:***

***=>After writing the program,save the program with language***

***extention.***

***Ex:***

***Test.c***

***Test.cpp***

***Test.java***

***=>The program will have the following two stages to generate***

***result:***

***1.Compilation process***

***2.Execution process***

***1.Compilation process:***

***=>The process of checking the program constructed according***

***to the rules of language or not,is known as Compilation process.***

***=>After Compilation process is Successfull,***

***=>c and c++ programs generate Objective Code.***

***=>Java programs generate Byte Code***

***2.Execution process:***

***=>The process of running the compiled code and checking the***

***required output is generated or not,is known as Execution process***

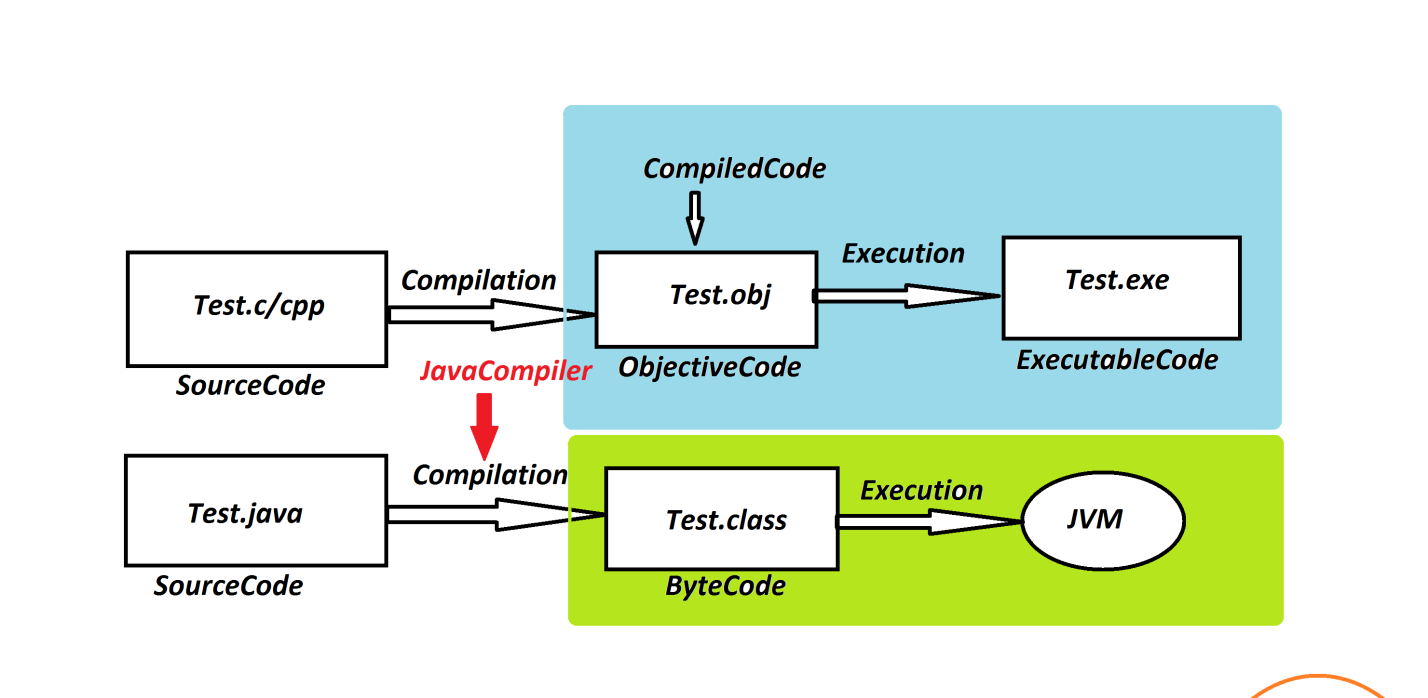
***=>In C and c++,Objective Code is converted into Executable code***

***and generate result.***

***=>In JavaLang the ByteCode is executed on JVM(Java Virtual***

***Machine) and generate result.***

***Diagram:***

******

***==================================================***

***faq:***

***wt is the diff b/w***

***(i)Objective Code***

***(ii)Byte Code***

***(i)Objective Code:***

***=>The compiledCode generated from c and c++ programs is known***

***as Objective Code.***

***=>while Objective Code generation,operatingSystem is***

***participated beacause of this reason Objective Code is PlatForm***

***dependent code.***

***DisAdvantage:***

***=>The ObjectiveCode which is generated from One PlatForm cannot***

***be executed on other PlatForms.***

***Note:***

***=>C and c++ languages which are generating Objective Code are***

***Platform dependent languages.***

***-------------------------------------------------***

***(ii)Byte Code:***

***=>The Compiled Code generated from JavaPrograms is known as***

***Byte Code.***

***=>while Byte Code generation Operating System is not***

***participated,because of this reason Byte Code is Platform***

***independent code.***

***Advantage:***

***=>The Byte code generated from one platform can be executed on***

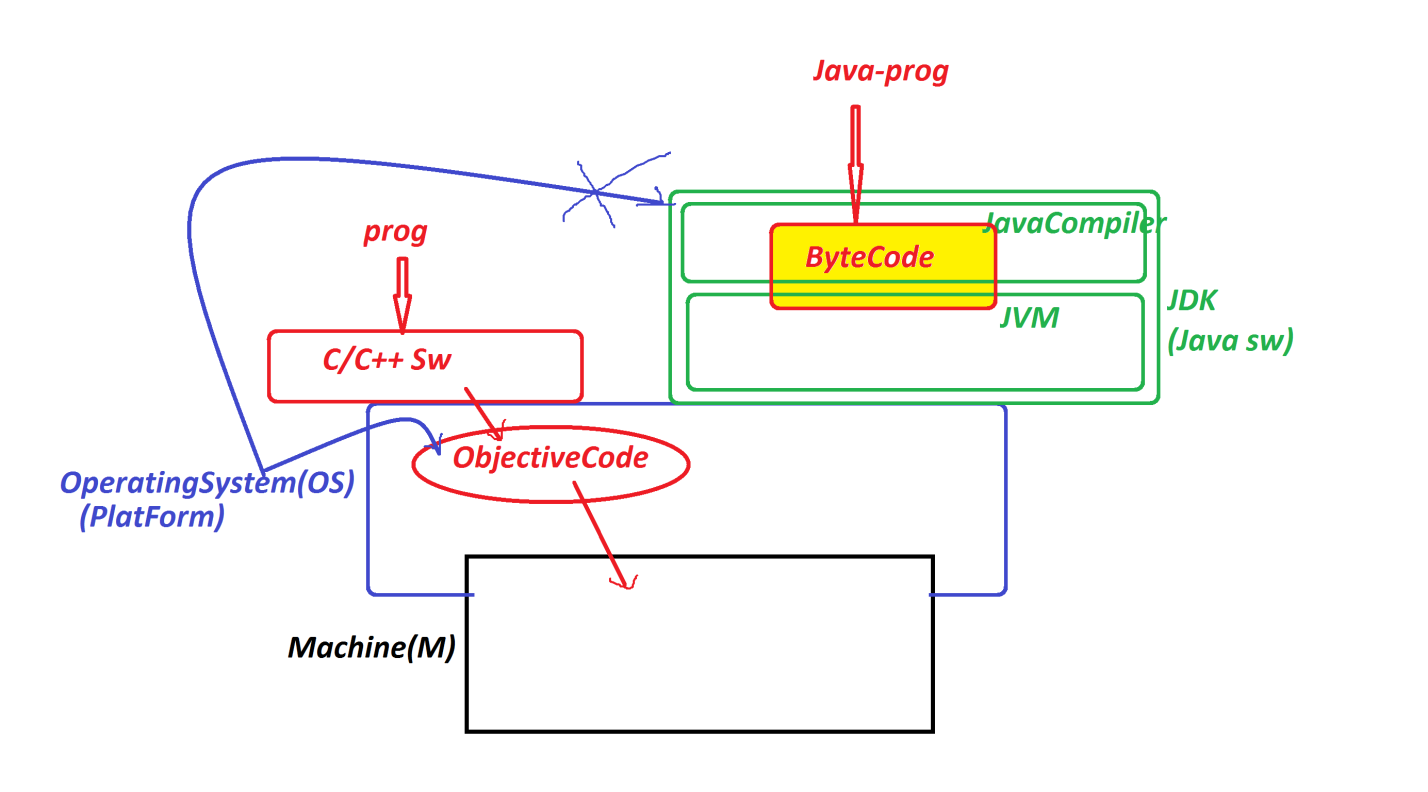
***others platforms where JVM is available.***

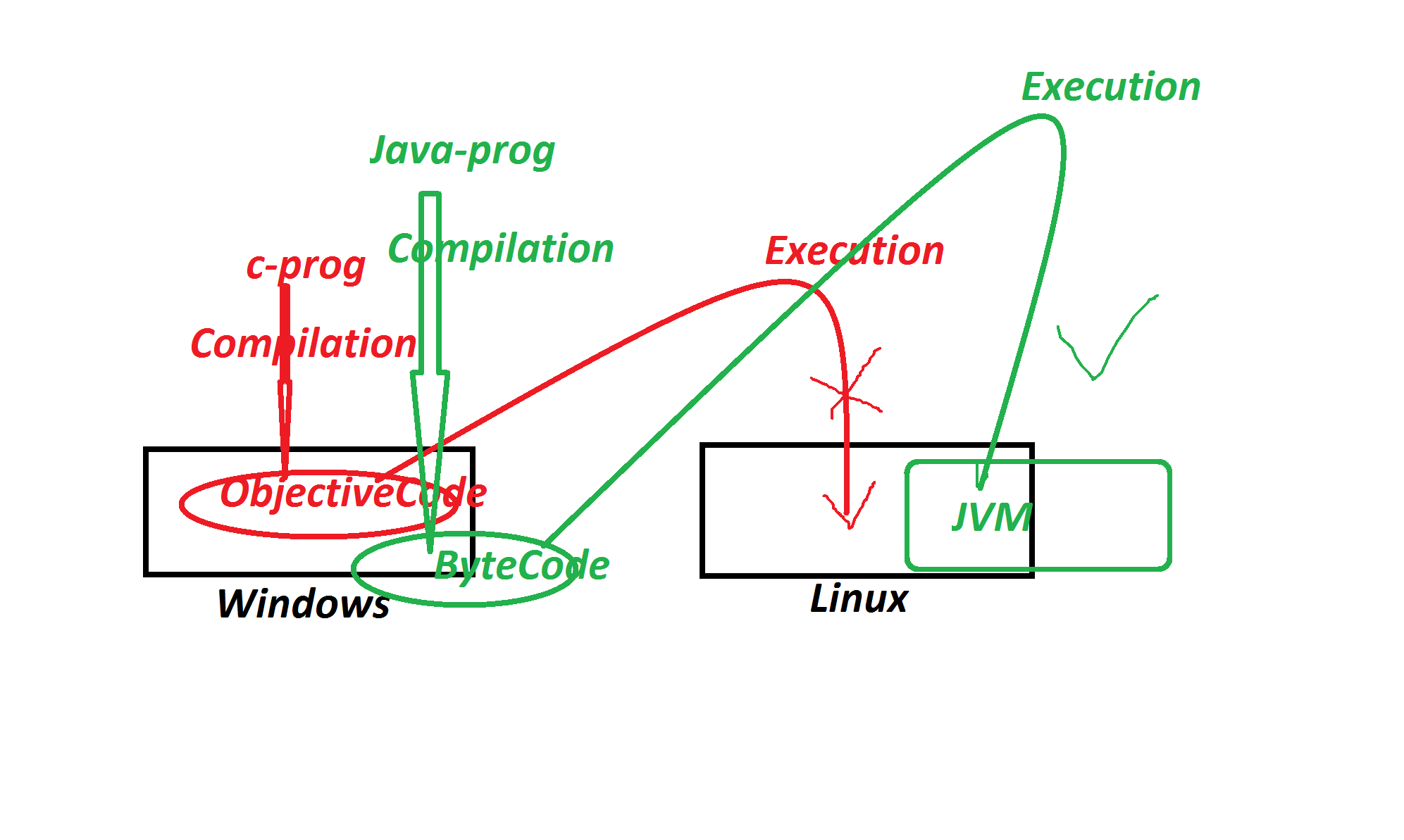
***Note:***

***=>The JavaLanguage which is generating ByteCode is PlatForm***

***independent language.***

***Diagram:***

******

******

***========================================================***

***Dt : 3/8/2022***

***faq:***

***define High Level Languages?***

***=>The languages which are understandable by the users are***

***known as High Level Languages.***

***Ex:***

***c,c++,java***

***faq:***

***define Low Level Languages?***

***=>The languages which are not Understandable by the user are***

***known as Low Level Languages.***

***Ex:***

***Machine Languages***

***faq:***

***define Translators?***

***=>Translators are used to convert High Level Language formats***

***into Low Level Language formats and Low Level Language formats***

***into High Level Language formats***

***=>These Translators are categorized into two types:***

***(i)Compilers***

***(ii)Interpreters***

***(i)Compilers - Translate the total program at-a-time.***

***(ii)Interpreters - Translate the program line-by-line***

***Note:***

***=>c and c++ languages use Compiler in Compilation and Execution***

***processes.***

***=>Java Lang use compiler in Compilation process and uses***

***'Compiler and Interpreter' in Execution process.***

***=======================================================***

***James Gosling - Sun micro Systems - 1991 - CodeWriter(Programmer)***

***WORA - Write Once and Run Anywhere***

***Test.gt(Green Talk)***

***OAK Language***

***SILK DNA JAVA***

***\*imp***

***Java Versions:***

***1995 - Java Alpha & Beta***

***1996 - JDK 1.0***

***1997 - JDK 1.1***

***1998 - JDK 1.2***

***2000 - JDK 1.3***

***2002 - JDK 1.4***

***------------------------***

***2004 - Java5 (Tiger)***

***2006 - Java6***

***2011 - Java7***

***--------------------------***

***2014 - Java8***

***2017 - Java9***

***2018 - Java10,Java11***

***2019 - Java12,Java13***

***2020 - Java14,Java15***

***2021 - Java16,Java17***

***2022 - Java18***

***============================================***

***LTS(Long Term Support)-Products : Java8,Java11,Java17***

***============================================***

***Dt : 4/8/2022***

***faq:***

***wt is the diff b/w***

***(i)JDK***

***(ii)JRE***

***(i)JDK:***

***=>JDK stands for 'Java Development Kit' and which provide the***

***following,used for Java Application development.***

***(a)Java Compiler***

***(b)Java Library***

***(c)JVM***

***(a)Java Compiler:***

***=>Java Compiler is used to translate 'Java Source Code' into***

***'Java Byte Code'.***

***(b)Java Library:***

***=>Java Library will provide Pre-defined programming components***

***which are used in application development.***

***=>Java Library is represented using the word 'java'.***

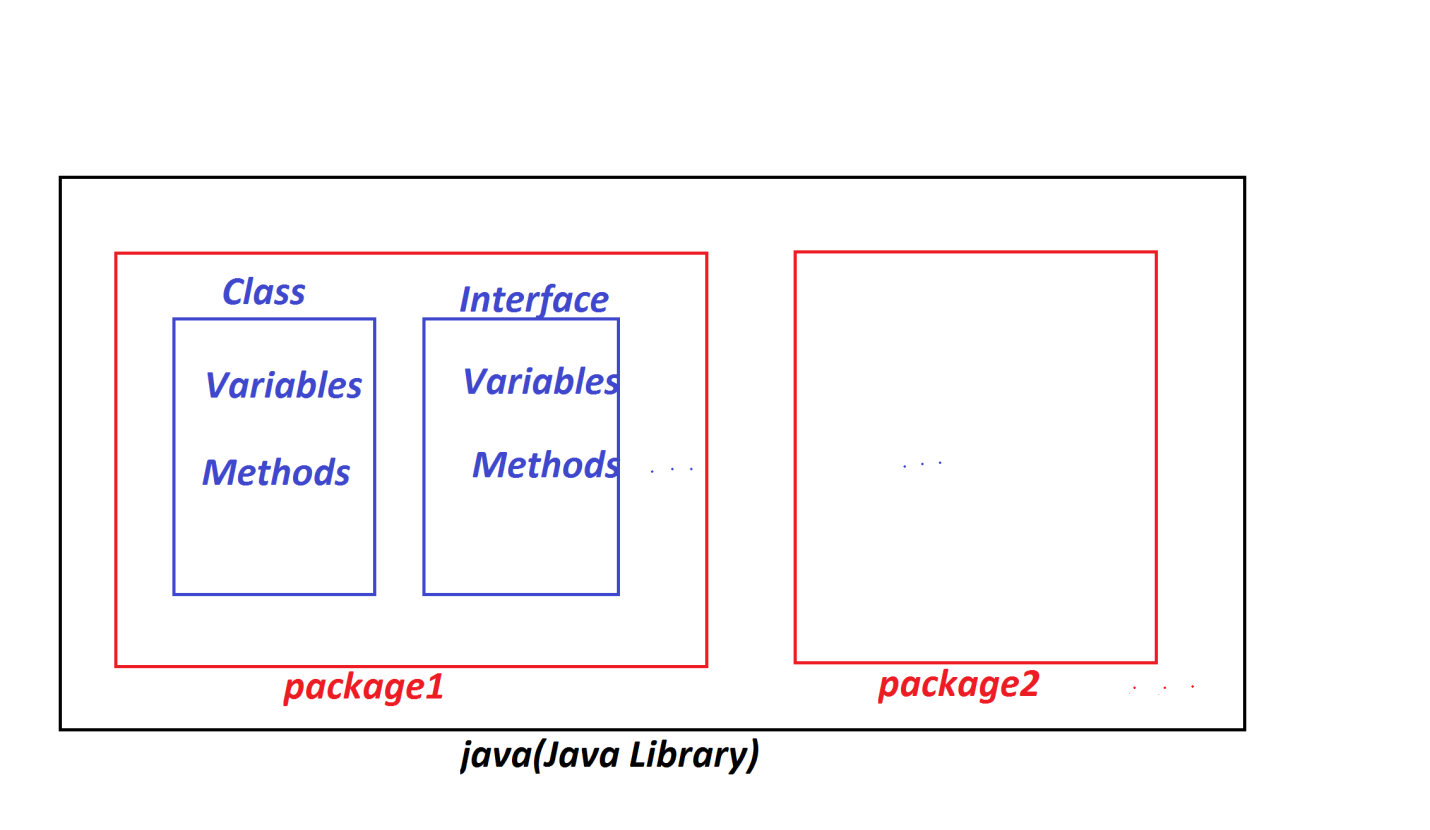
***=>Java Library is collection of 'packages'***

***=>Packages are Collection of 'Classes and Interfaces'***

***=>'Classes and Interfaces' are collection of 'Variables and***

***Methods'***

***Diagram:***

******

***=>The following are some important packages from JavaLib:***

***java.lang - Language package***

***java.io - Input Output package***

***java.net - Networking package***

***java.util - Utility package***

***(c)JVM:***

***=>JVM stands for 'Java Virtual Machine' and which is used to***

***execute 'Java Byte Code' and generate result.***

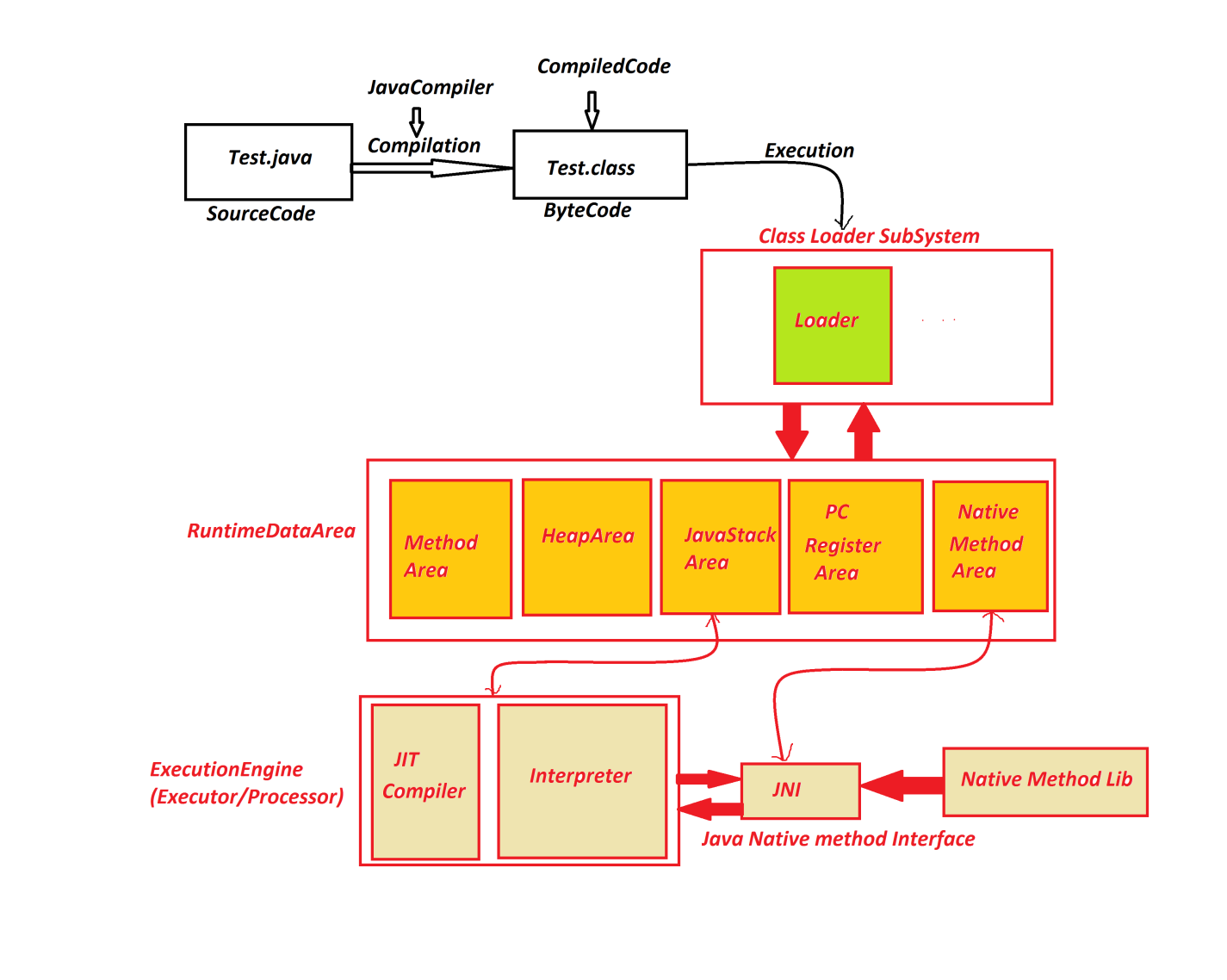
***=>JVM internally having the following patitions:***

***1.Class Loader SubSystem***

***2.Runtime Data Area***

***3.Execution Engine(Processor/Executor)***

***Diagram:***

******

***========================================================***

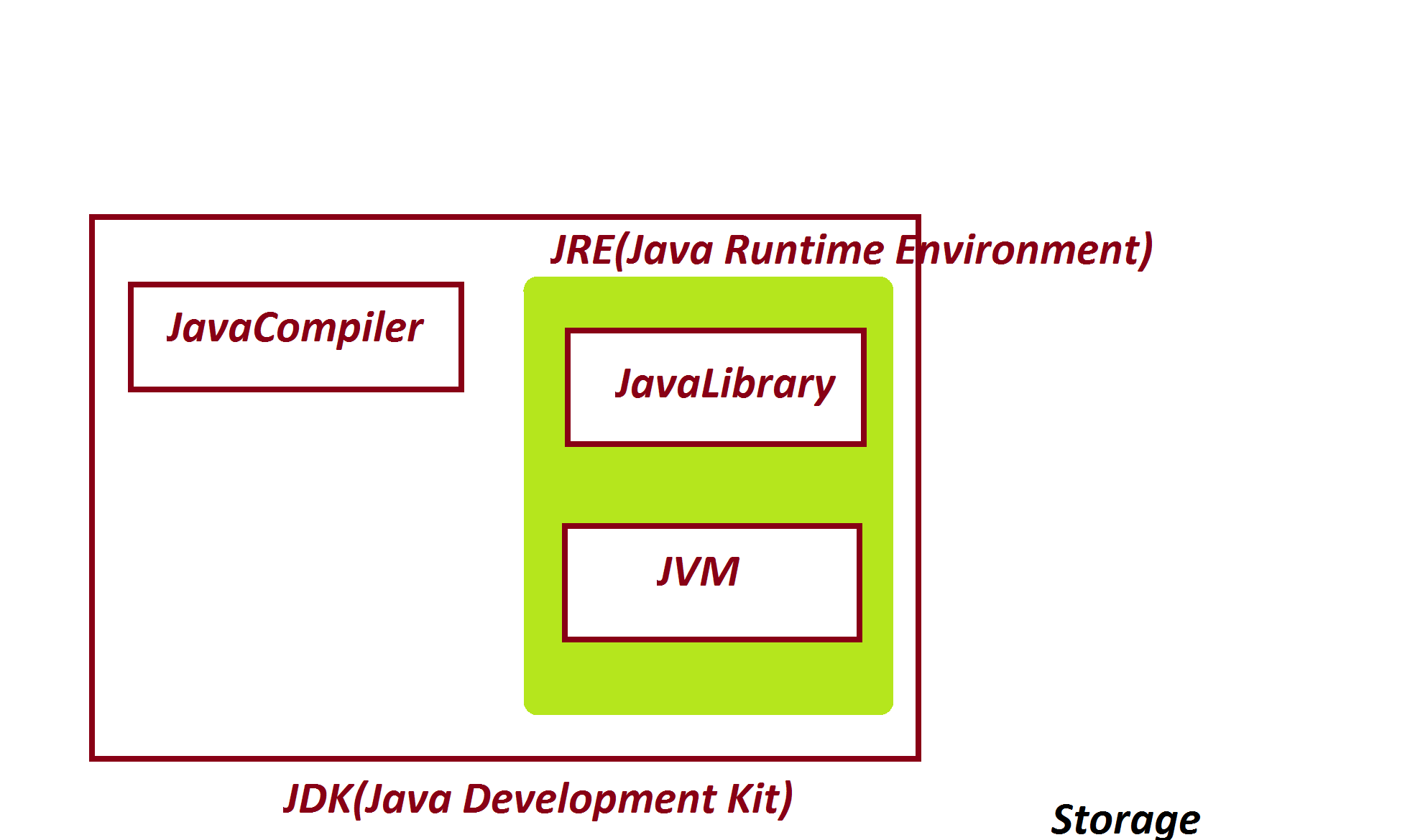
***(ii)JRE:***

***=>JRE stands for 'Java Runtime Environment' and which is***

***Collection of only 'Java Library' and 'JVM'.***

***=>JRE is internal part of JDK.***

***Diagram:***

******

***==========================================================***

***Dt : 5/8/2022***

***\*imp***

***Installing Java S/w and Setting path:***

***step-1 : Download Java S/w(JDK) from Oracle WebSite***

***https://www.oracle.com/java/technologies/downloads/***

***step-2 : Install JDK***

***Note:***

***=>After Installation process is Successfull,we can find the***

***folder with name 'java' in 'ProgramFiles'***

***C:\Program Files\Java***

***step-3 : Set JavaPath in 'Environment Variables'***

***RightClick on MyComputer->Properties->Advanced System Settings->***

***Environment Variables->click 'new' from System Variables***

***Variable name : path***

***Variable value : C:\Program Files\Java\jdk-17.0.3.1\bin;***

***step-4 : Click 'ok' for three times***

***=========================================================***

***Note:***

***=>Open CommandPrompt and check the following commands are***

***Working or not:***

***javac - Compilation command***

***java - Execution command***

***========================================================***

***Note:***

***=>use the following syntax to know the Java version:***

***java -version***

***=========================================================***

***\*imp***

***define 'class' in Java?***

***=>class is a 'Structured Layout' in Java and which generate***

***Objects.***

***=>class is collection of Variables,Methods and main()***

***Variables - will hold data***

***Methods - will perform actions***

***main() - will start the program execution***

***=>In java,main() will have the following pre-defined format***

***public static void main(String[] args)***

***=>we use 'class' keyword to construct classes in Java***

***structure of class in Java:***

***class Class\_name***

***{***

***//variables***

***//methods***

***//main()***

***}***

***===========================================================***

***Wap to display the msg as "Welcome to Java"?***

***class Display***

***{***

***public static void main(String args[])***

***{***

***System.out.print("Welcome to Java");***

***}***

***}***

***=======================================================***

***\*imp***

***writing,Saving,Compiling and Executing Java program:***

***step-1 : create one folder in any drive***

***E:\Demo136***

***step-2 : Open notepad and type the program***

***step-3 : Save the program in folder(Demo136) with language***

***extention***

***Click on file->save->Browse the folder->name the file as***

***"Display.java"->click 'save'***

***E:\Demo136\Display.java***

***Note:***

***=>Open CommandPrompt and perform 'Compilation and execution***

***process'.***

***To Open CommandPrompt->goto folder->type 'cmd' in AddressBar***

***and press 'enter'***

***step-4 : Compile the program as follows***

***syntax:***

***javac Class\_name.java***

***Ex:***

***javac Display.java***

***step-5 : Execute the program as follows***

***syntax:***

***java Class\_name***

***Ex:***

***java Display***

***========================================================***

***Dt : 6/8/2022***

***Ex-2 :***

***wap to display the sum of two numbers?***

***class Addition***

***{***

***public static void main(String args[])***

***{***

***int a=12,b=13,c;***

***c = a+b;***

***System.out.println("Value of a="+a);***

***System.out.println("Value of b="+b);***

***System.out.println("The sum="+c);***

***}***

***}***

***o/p:***

***Value of a=12***

***Value of b=13***

***The sum=25***

***=======================================================***

***Note:***

***=>"+" symbol in print() method specify combining message with***

***result.***

***=>print() method will display data in same line and println()***

***method will display the data in different lines.***

***==========================================================***

***Assignment-1:***

***wap to display the average of three numbers?***

***Variables:***

***int a,b,c;***

***float d;***

***o/p:***

***Value of a=***

***Value of b=***

***Value of c=***

***Average =***

***========================================================***

***Assignment-2:***

***wap to display Student Six submarks,totMarks and percentage?***

***o/p:***

***s1=***

***s2=***

***s3=***

***s4=***

***s5=***

***s6=***

***TotMarks=***

***Per=***

***=======================================================***

***Dt : 8/8/2022***

***\*imp***

***Naming Conventions in Java:***

***=>The coding rules followed by the programmer in constructing***

***programs are known as Naming Conventions in Java.***

***packages :***

***def : packages are collection of 'Classes & Interfaces'***

***rule : packages must be in LowerCase.***

***Classes and Interfaces:***

***def : 'Classes and Interfaces' are collection of 'Variables***

***and methods'.***

***rule : In 'Classes and Interfaces' the starting letter of every***

***word must be capital.***

***Ex:***

***Addition***

***EmployeeSalary***

***InputStreamReader***

***Variables and Methods:***

***def :***

***=>Variables are the data holders and Methods are the actions.***

***rule : In Variables and methods the first word in LowerCase and***

***from Second word onwrads the starting letter capital.***

***Ex variables:***

***panCardNo***

***basicSalary***

***rollNo***

***Ex methods:***

***calculateTotSal()***

***getEmployeeSalary()***

***Keywords:***

***def : The pre-defined words or Built-in words are known as***

***Keywords.***

***rule : The keywords must be in LowerCase.***

***Ex:***

***void***

***public***

***static***

***...***

***============================================================***

***faq:***

***define Identifiers?***

***=>The names given for Programming components like Variables,***

***methods,Classes and Interfaces are known as Identifiers.***

***=>Using Identifiers we can identify the programming components.***

***=============================================================***

***\*imp***

***DataTypes in Java:***

***=>Types of data which we are expecting as input to Java programs***

***are known as "DataTypes in Java".***

***=>DataTypes in Java are categorized into two types:***

***1.Primitive Datatypes***

***2.Non-Primitive Datatypes***

***1.Primitive Datatypes:***

***=>The 'single valued data formats' are known as Primitive***

***Datatypes.***

***=>These Primitive Datatypes are categorized into four types:***

***(a)Integer datatypes***

***(b)Float datatypes***

***(c)Character datatype***

***(d)Boolean datatype***

***(a)Integer datatypes:***

***=>The numeric data which is represented without decimal point***

***are known as Integer datatypes.***

***=>Types based on size:***

***(i)byte - 1 byte(8-bits)***

***(ii)short - 2 bytes***

***(iii)int - 4 bytes***

***(iv)long - 8 bytes***

***(b)Float datatypes:***

***=>The numeric data which is represented with decimal point are***

***known as Float datatypes.***

***=>Types based on size:***

***(i)float - 4 bytes***

***(ii)double - 8 bytes***

***(c)Character datatype:***

***=>The 'Single valued charcter' which is represented in single***

***quotes is known as Character datatype.***

***Ex:***

***'n','y',...***

***=>Types:***

***char - 2 bytes***

***(d)Boolean datatype:***

***=>The datatype which is represented in the form of true or***

***false is known as boolean datatype***

***=>Types:***

***boolean - 1 bit***

***--------------------------------------------------------***

***\*imp***

***2.Non-Primitive Datatypes:***

***=>The 'group valued data formats' are known as Non-Primitive***

***datatypes or Referential datatype.***

***=>The following are four types of Non-Primitive datatypes:***

***(a)Class***

***(b)Interface***

***(c)Array***

***(d)Enum***

***==========================================================***

***\*imp***

***Object Oriented Programming in Java:***

***=>The process of constructing programs using Class-Object***

***Concept is known as Object Oriented Programming.***

***=>In Object Oriented programming we control Non-Primitive***

***Datatypes.***

***==========================================================***

***Dt : 9/8/2022***

***\*imp***

***define 'Object'?***

***=>Object is a Storage related to a class holding the members***

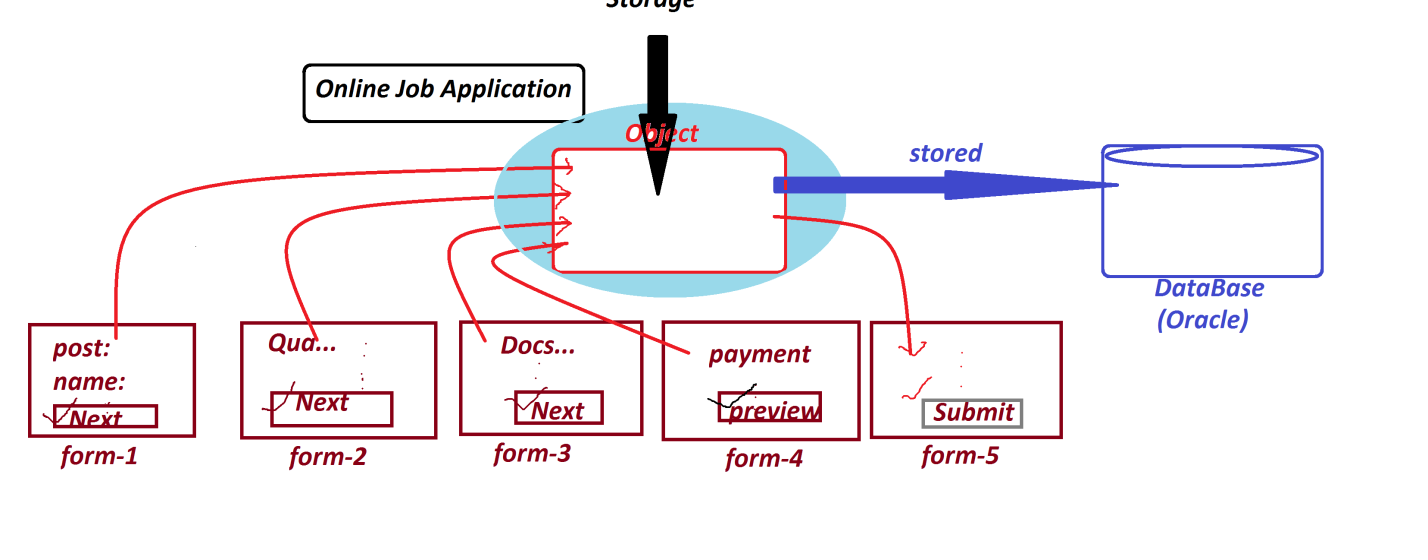
***of class.***

***=>we use 'new' keyword in Java to create Objects.***

***syntax of Object creation:***

***Class\_name obj\_name = new Class\_name();***

***Diagram:***

******

***==========================================================***

***\*imp***

***(a)Class:***

***=>Class is a 'Structured layout' in Java and which generate***

***Objects.***

***=>Class is collection of Variables and Methods.***

***=>Classes in Java are categorized into two types:***

***(i)Pre-defined classes***

***(ii)User defined classes***

***(i)Pre-defined classes:***

***=>The classes which are already constructed and available***

***from JavaLib are known as Pre-defined classes or Built-in classes.***

***Ex:***

***String***

***System***

***(ii)User defined classes:***

***=>The classes which are constructed by the programmer are***

***known as User defined classes***

***Ex:***

***Display***

***Addition***

***===========================================================***

***\*imp***

***Variables in Java:***

***=>Variables are the data holders in programs.***

***=>Variables in Java are categorized into two types:***

***1.Primitive DataType variables***

***2.Non-Primitive DataType variables***

***1.Primitive DataType variables:***

***=>The variables which are declared with primitive datatypes***

***like byte,short,int,long,float,double,char,boolean are known***

***as Primitive DataType variables.***

***Note:***

***=>Primitive DataType Variables will hold values.***

***2.Non-Primitive DataType variables:***

***=>The variables which are declared with Non-Primitive datatypes***

***like Class,Interface,Array,Enum are known as Non-Primitive***

***DataTypes Variables or Referential DataTypes Variable***

***Note:***

***=>Non-Primitive DataType variables will hold Object references.***

***============================================================***

***\*imp***

***=>Based on 'static' keyword,the variables in Java are categorized***

***into two types:***

***1.Static Variables***

***2.Non-Static Variables***

***1.Static Variables:***

***=>The variables which are declared with 'static' keyword***

***outside the methods are known as static variables or***

***Class Variables.***

***=>These Static variables will get the memory within the class***

***while class loading.***

***=>These static variables can be accessed with Class\_name.***

***2.Non-Static Variables:***

***=>The variables which are declared without 'static' keyword***

***are known as Non-Static Variables.***

***=>These Non-Static variables are categorized into two types:***

***(a)Instance Variables***

***(b)Local Variables***

***(a)Instance Variables:***

***=>The Non-Static variables which are declared outside the***

***methods are known as Instance Variables or Object Variables.***

***=>These Instance variables will get the memory within the***

***Object while Object creation.***

***=>These Instance Variables are accessed with Object\_name.***

***(b)Local Variables:***

***=>The Non-Static variables which are declared inside the***

***methods are known as Local Variables***

***=>These Local variables will get the memory within the method***

***while method execution.***

***Ex : DemoVariables.java***

***class DemoVariables***

***{***

***static int a=10;***

***int b=20;***

***public static void main(String[] args)***

***{***

***int c = 30;***

***System.out.println("a="+DemoVariables.a);***

***DemoVariables ob = new DemoVariables();***

***System.out.println("b="+ob.b);***

***System.out.println("c="+c);***

***}***

***}***

***===================================================================***