***Dt : 8/8/2022(Day-1)***

***(i)C-Lang -> Logical Programming***

***(ii)JavaLang -> Object oriented Programming***

***Note:***

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

***Construction rules.***

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

***JavaLanguage:***

***part-1 : CoreJava***

***part-2 : AdvJava***

***part-1 : CoreJava***

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

***(2)Java Programming Concepts***

***(3)Object Oriented Programming features***

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

***(a)Variables***

***1.Primitive datatype variables(Values)***

***2.Non-Primitive datatype variables(object references)***

***(b)Methods***

***(c)Blocks***

***(d)Constructors***

***(e)Classes***

***1.Pre-defined classes***

***2.User defined clases***

***(f)Interfaces***

***(g)AbstractClasses***

***(2)Java Programming Concepts:***

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

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

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

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

***(data Structures in Java)***

***(e)File Storage***

***(f)Networking in Java(Communication with TCP/IP protocol)***

***(3)Object Oriented Programming features:***

***(a)Class***

***(b)Object***

***(c)Abstraction***

***(d)Encapsulation***

***(e)PolyMorphism***

***(f)Inheritance***

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

***Note:***

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

***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 applications means,***

***No HTML input***

***No Server Environment***

***No DataBase Storage***

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

***part-2 : AdvJava***

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

***Web Aplications:***

***1.JDBC***

***2.Servlet***

***3.JSP***

***1.JDBC :***

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

***used to interact with DataBase product.***

***2.Servlet:***

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

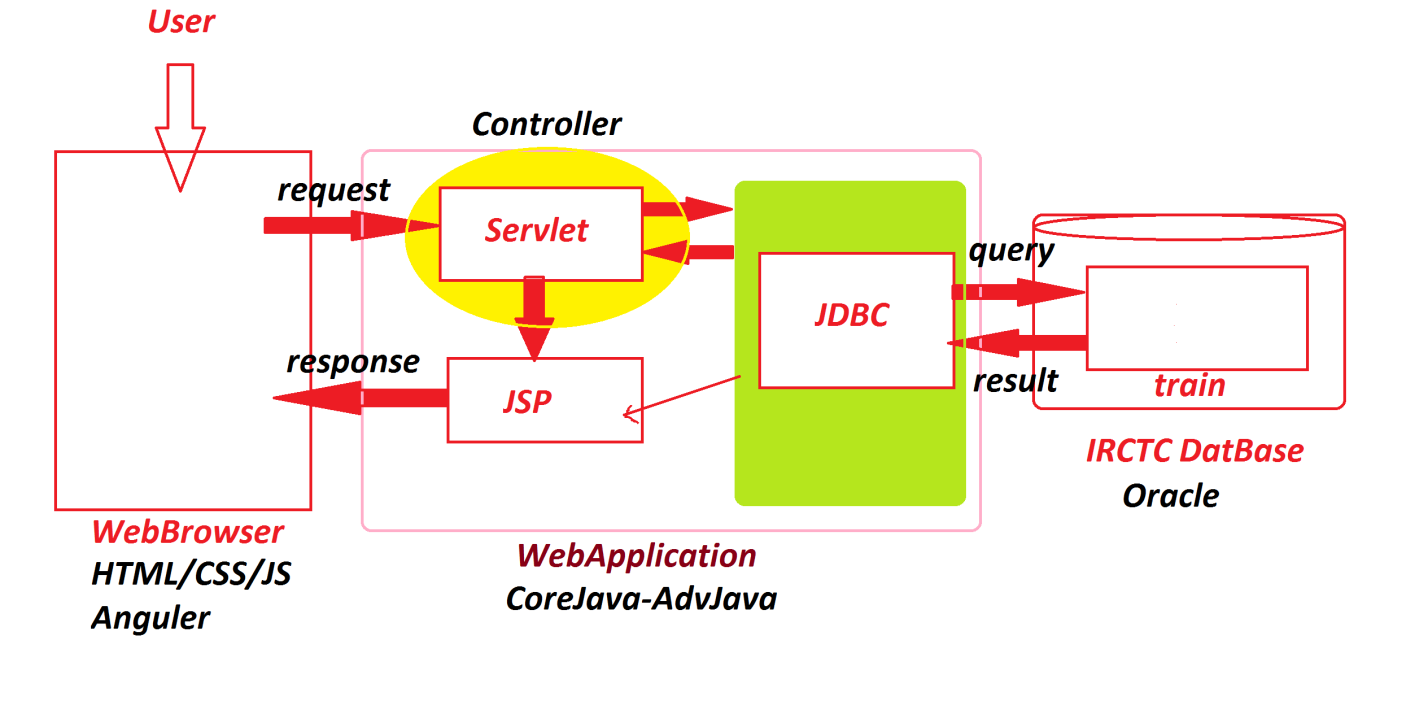
***from User through WebBrowser.***

***3.JSP:***

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

***Web Applications.***

***Diagram:***

******

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

***faq:***

***define Web Applications?***

***=>The applications which are executed in Web Environment or***

***Internet Environment are known as Web Applications.***

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

***faq:***

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

***(i)Language***

***(ii)Technology***

***(iii)Framework***

***(i)Language:***

***=>Language provide Programming components,Concepts and***

***Features used in application development.***

***Ex:***

***CoreJava***

***(ii)Technology:***

***=>The process of converting the knowledge into 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***

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

***Dt : 9/8/2022(Day-2)(Syllabus)***

***\*imp***

***define Program?***

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

***define Programming?***

***=>The process of constructing programs is known as Programming,***

***which means Converting 'Analysis and Design' into CodeForm.***

***define Programmer?***

***=>The person who writes the programs is known as Programmer.***

***Project:***

***SDLC(Software Development Life Cycle)***

***stage-1 : Analysis***

***stage-2 : Design***

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

***stage-4 : Testing***

***stage-5 : Deployment & Maintanance***

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

***Note:***

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

***extention***

***Ex:***

***Test.c***

***Test.cpp***

***Test.java***

***=>After saving the program,the program will have the following***

***two stages:***

***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 and Java***

***Programs generate ByteCode.***

***2.Execution process:***

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

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

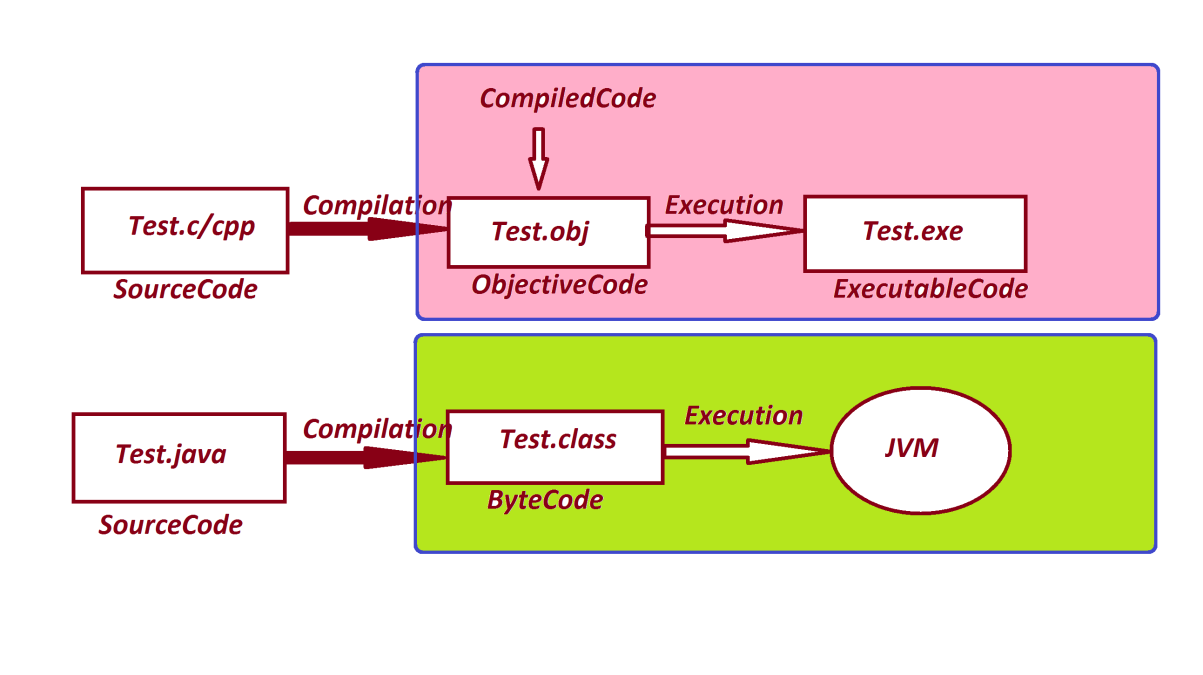
***=>In c and c++ languages,the Objective code is converted into***

***Executable code and generate result.***

***=>In Java language 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 compiled code generated from c and c++ programs is known***

***as Objective Code.***

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

***participated,because of this reason Objective Code is PlatForm***

***dependent code.***

***DisAdvantage:***

***=>The Objective Code 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***

***ByteCode.***

***=>while ByteCode generation OperatingSystem 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***

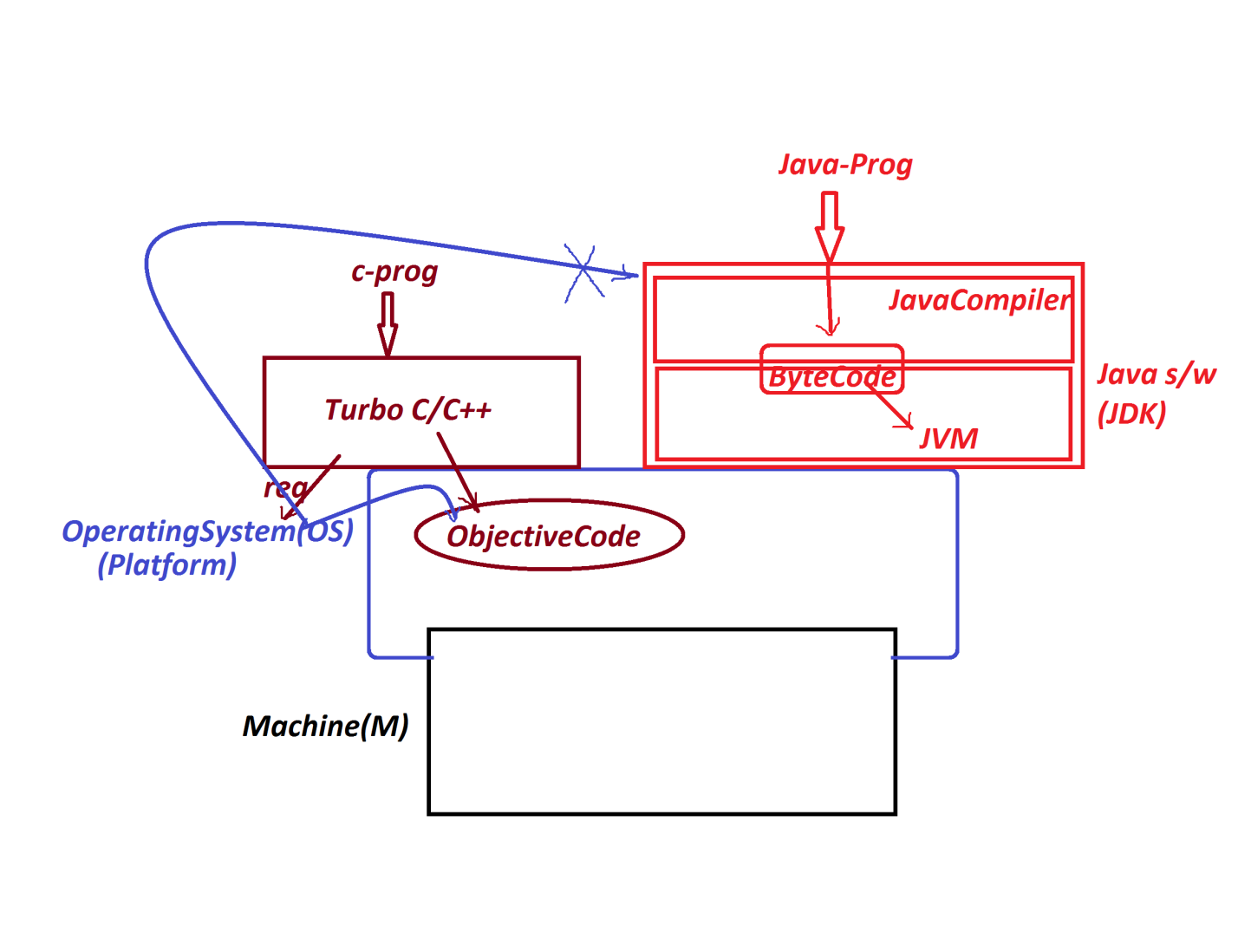
***other platforms based on JVM.***

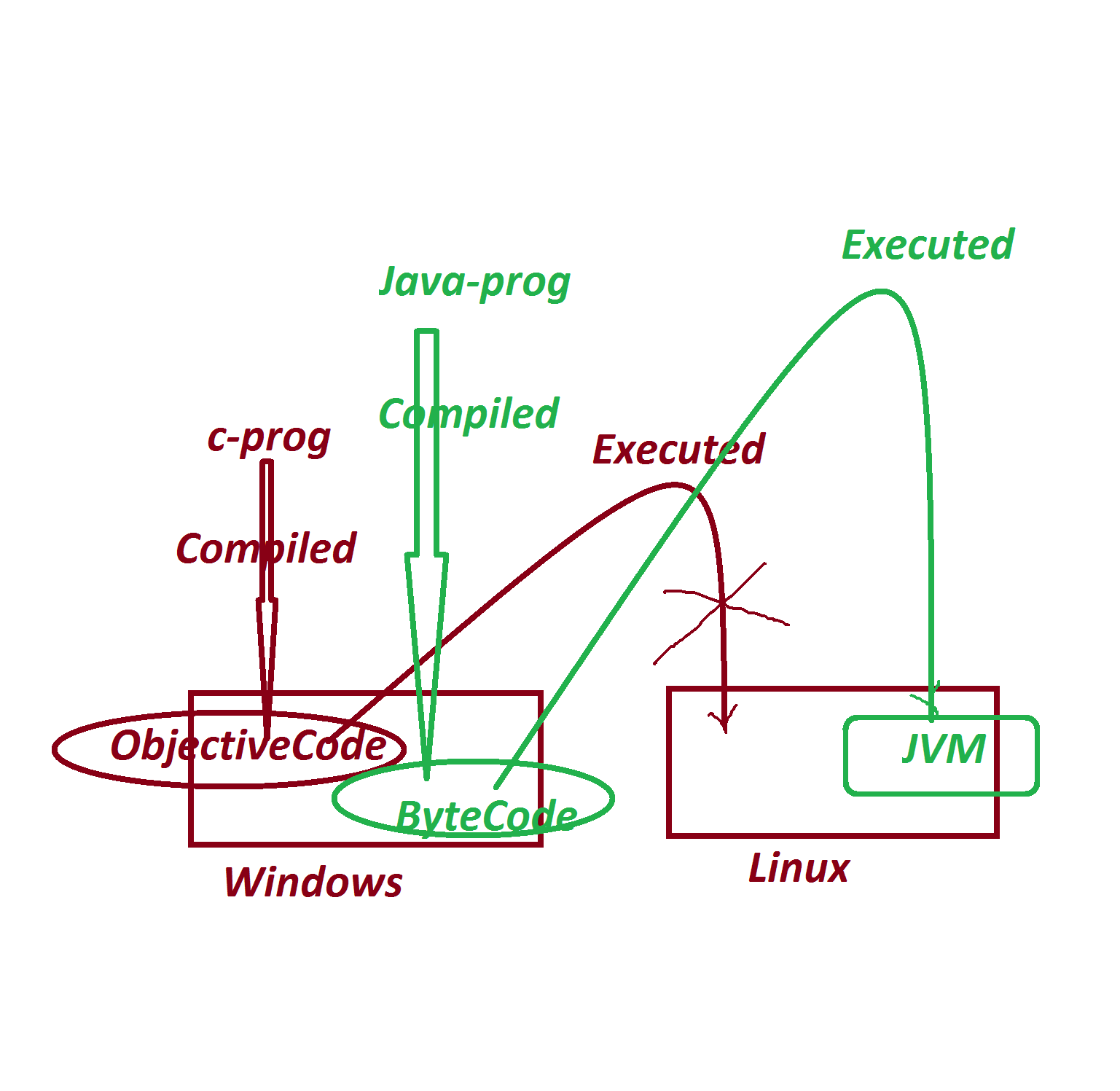
***Note:***

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

***independent language.***

***Diagram:***

******

******

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

***"James Gosling" - SunMicroSystems - 1991 - CodeWriter(Programmer)***

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

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

***OAK Language***

***SILK DNA JAVA***

***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***

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

***Note:***

***=>Java8,Java11 and Java17 are LTS(Long Term Support) products.***

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

***Dt : 10/8/2022(Day-3)***

***faq:***

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

***(i)JDK***

***(ii)JRE***

***(i)JDK:***

***=>JDK stands for 'Java Development Kit' and which is collection***

***of following:***

***(a)Java Compiler***

***(b)Java Library***

***(c)JVM***

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

***=>Java Compiler will compile the program and translates the***

***JavaSourceCode into JavaByteCode.***

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

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

***which are used in constructing Java Applications.***

***=>we use 'java' word to represent JavaLib.***

***=>JavaLib is collection of 'packages'.***

***=>packages are collection of 'Classes and Interfaces'.***

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

***Methods'.***

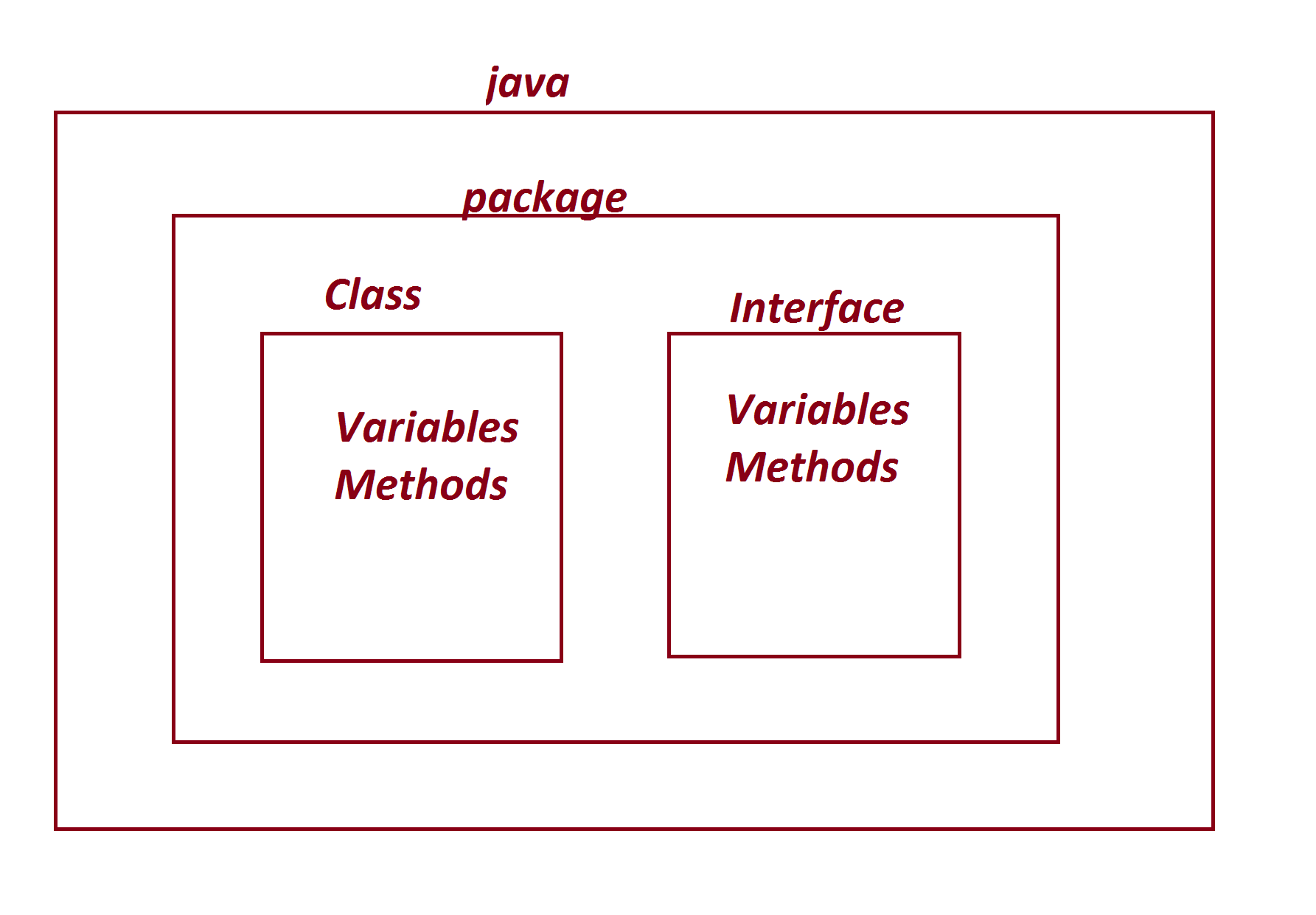
***=>The following are some important packages:***

***java.lang - language package(default package)***

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

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

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

******

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

***(c)JVM:***

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

***execute JavaByteCode and generate result.***

***=>The JVM internally divided into the following three***

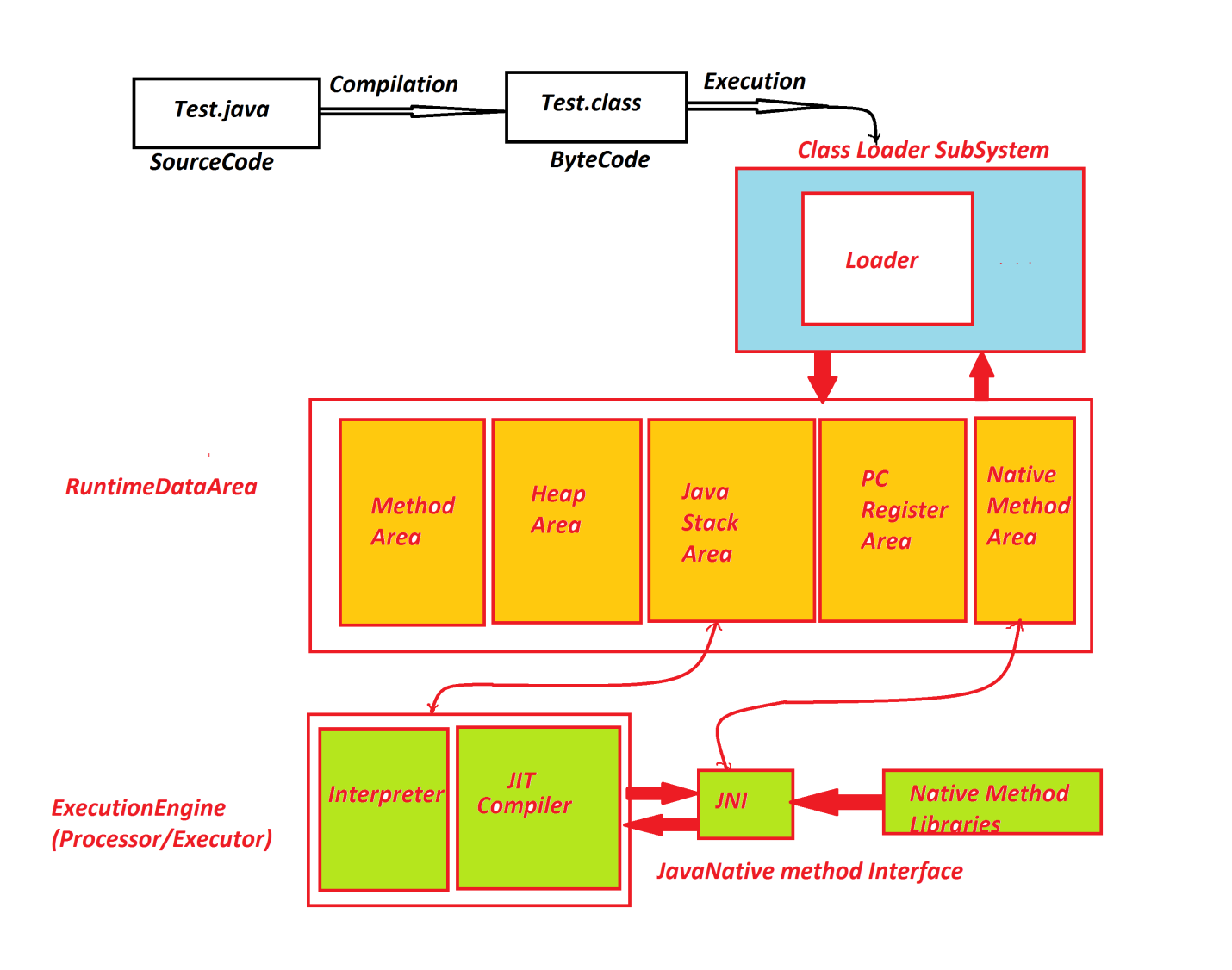
***partitions:***

***1.Class Loader SubSystem***

***2.Runtime Data Area***

***3.Execution Engine***

***=>JVM Architecture:***

******

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

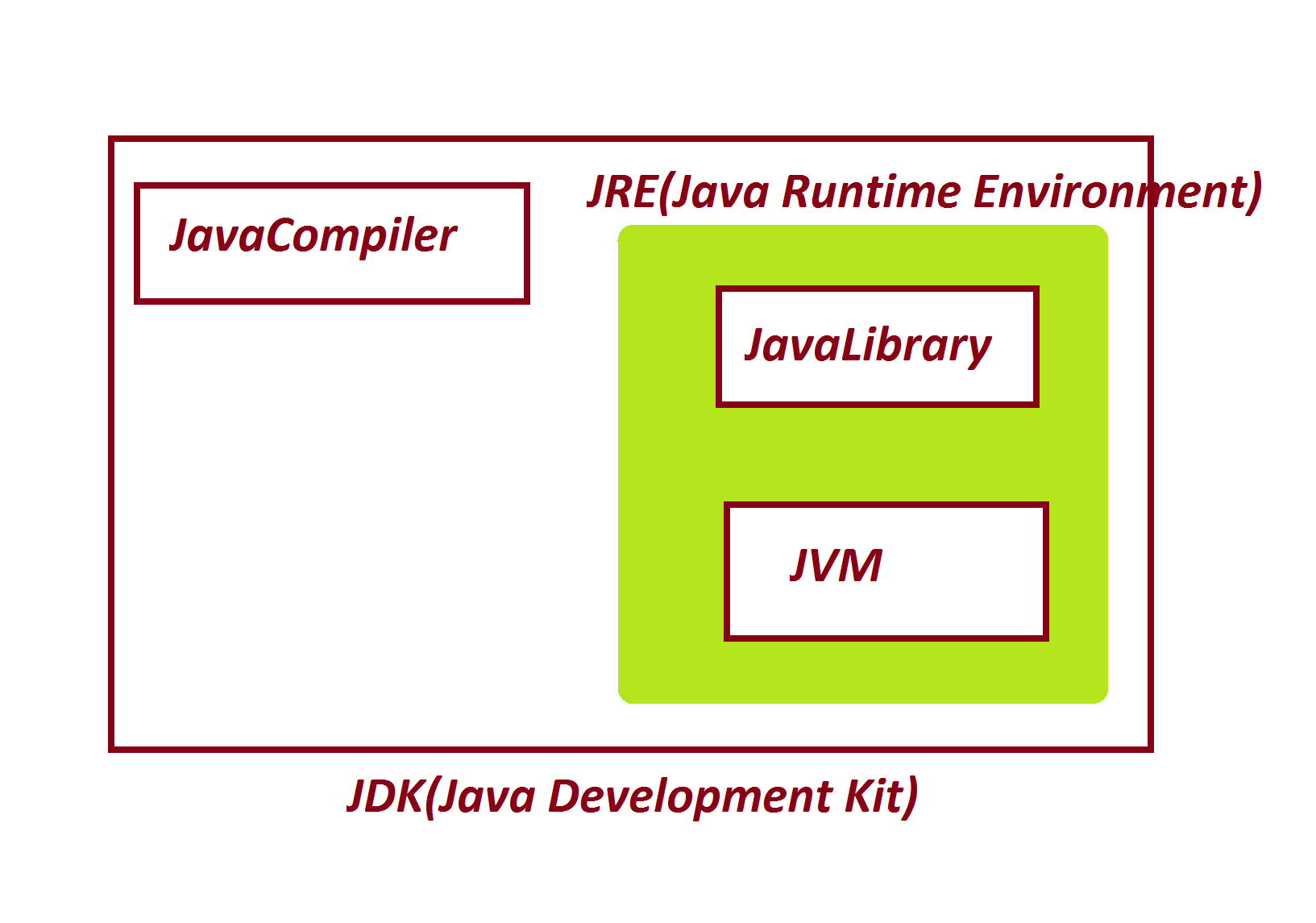
***(ii)JRE:***

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

***Collection of JavaLib and JVM.***

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

***Diagram:***

******

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

***\*imp***

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

***step-1 : Downlaod JDK17 from Oracle WebSite***

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

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

***Note:***

***=>After installation process is successfull,we can find one***

***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:***

***javac - Compilation Command***

***java - Execution Command***

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

***Note:***

***=>Use the following syntax to know the Java Version:***

***java -version***

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

***Dt : 11/8/2022***

***\*imp***

***define 'class'?***

***=>class is a 'structured layout' in java and which generate***

***objects.***

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

***Variables - are data holders***

***Methods - are actions***

***main() - is starting point of execution process.***

***=>In Java,main() is having the following pre-defined format:***

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

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

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

***class Class\_name***

***{***

***Variables***

***methods***

***main()***

***}***

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

***Ex\_Program:***

***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:\Demo137***

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

***step-3 : Save the program in folder with language extention***

***click on file->Save->browse the folder->name the file as***

***'Display.java' and select 'save as type' as 'All Files' ->***

***click 'save'.***

***Note:***

***=>Open CommnadPrompt to perform Compilation and execution***

***process.***

***=>To open CommandPrompt,Goto folder->type 'cmd' in address bar***

***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***

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

***Ex-2:***

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

***class Addition***

***{***

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

***{***

***int a=10,b=20,c;***

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

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

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

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

***}***

***}***

***o/p:***

***a Value=10***

***b Value=20***

***Sum=30***

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

***Assignment-1:***

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

***Assignment-2:***

***wap to calculate torMarks and percentage based on 6 subject***

***marks?***

***s1=***

***s2=***

***s3=***

***s4=***

***s5=***

***s6=***

***totMarks = ...***

***per =***

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

***Note:***

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

***result.***

***=>print() method is used to display in same line.***

***=>println() method is used to display in different lines.***

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

***Dt : 12/8/2022***

***\*imp***

***Naming Conventions in Java:***

***=>The Coding rules followed by the programmer in writing Java***

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

***packages :***

***def : packages are collection of 'classes and Interfaces'***

***rule : package must be 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 UpperCase or capital.***

***Ex:***

***EmployeeSalary***

***InputStreamReader***

***Variables and Methods:***

***def : Variables are data holders and methods are actions.***

***rule : In Variables and methods the first word must be in***

***LowerCase and from second word onwards the starting***

***letter must be capital.***

***Ex:***

***panCardNo***

***rollNo***

***basicSal***

***calculateTotSal()***

***getRollNo()***

***readLine()***

***Keywords:***

***def : The pre-defined words which are already available are***

***known as Keywords or Built-in words***

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

***Ex:***

***void,static,public,int,...***

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

***faq:***

***define Identifiers?***

***=>The names of Programming components are known as Identifiers***

***and which are used to identify the components in programs.***

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

***\*imp***

***DataTypes in Java:***

***=>The 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.NonPrimitive 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 without decimal point representation are***

***known as Integer DataTypes.***

***=>Types:***

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

***short - 2bytes***

***int - 4bytes***

***long - 8bytes***

***(b)Float DataTypes:***

***=>The numeric data with decimal point representation are***

***known as Float DataTypes.***

***=>Types:***

***float - 4bytes***

***double - 8bytes***

***(c)Character DataType:***

***=>The 'single valued character' which is represented in***

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

***Ex:***

***'j','g','n',...***

***=>Types:***

***char - 2bytes***

***(d)Boolean DataType:***

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

***false is known as Boolean datatype.***

***=>Types:***

***boolean - 1bit***

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

***2.NonPrimitive DataTypes:***

***=>The 'group valued data formats' are known as NonPrimitive***

***DataTypes or Referential DataTypes.***

***=>These NonPrimitive DataTypes are categorized into four types:***

***(a)Class***

***(b)Interface***

***(c)Array***

***(d)Enum***

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

***\*imp***

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

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

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

***=>In Object Oriented programming we control NonPrimitive***

***datatypes or referential datatypes.***

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

***\*imp***

***define 'Object'?***

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

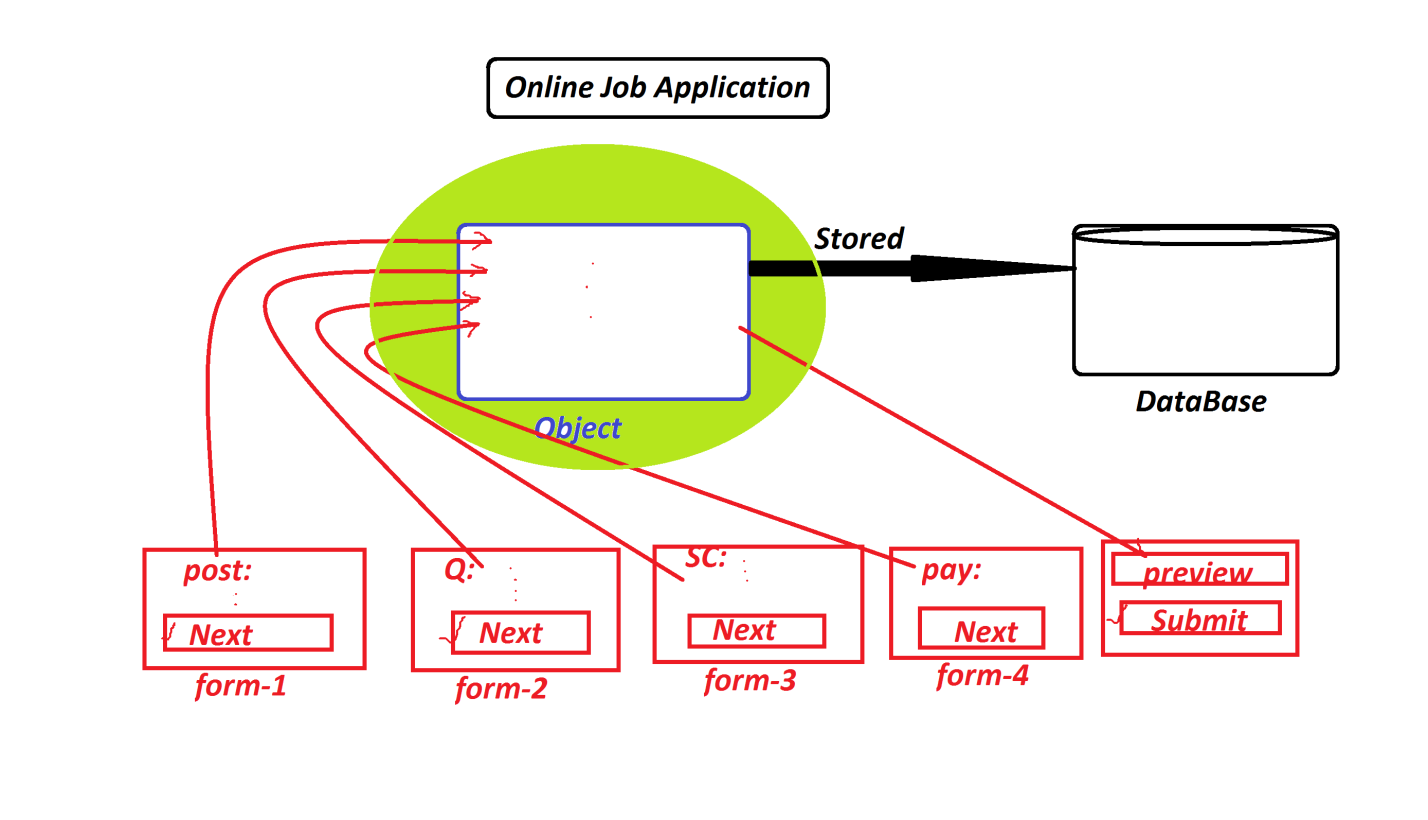
***of class.***

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

***syntax:***

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

***Diagram:***

******

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

***Dt : 18/8/2022***

***(a)Class:***

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

***Objects.***

***=>Class is a 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 defined 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 defined by the programmer are known***

***as User defined classes or Custom classes.***

***Ex:***

***Display***

***Addition***

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

***\*imp***

***Variables in Java:***

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

***=>based on datatype,the 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.***

***=>These 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 DataType variables or Referential datatype***

***variables.***

***=>These 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 and 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 NonStatic 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 and can be accessed with object\_name.***

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

***=>The NonStatic variables which are declared inside the***

***methods are known as Local Variables or Method Variables.***

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

***while method execution.***

***faq:***

***define static local variables?***

***=>There is no concept of static local variables in Java,***

***which we cannot declare local variables with 'static' keyword.***

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

***\*imp***

***Methods in Java:***

***=>Methods are the actions which are executed to generate some***

***results.***

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

***1.Static methods***

***2.Non-Static methods or Instance methods***

***1.Static methods:***

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

***known as static methods or Class methods.***

***=>These static methods will get the memory within the class***

***while class loading and can be accessed with Class\_name.***

***Structure of static methods:***

***static return\_type method\_name(para\_list)***

***{***

***//method\_body***

***}***

***Coding Rule:***

***=>static methods can access static variables directly,but***

***cannot access Instance Variables directly.***

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

***2.Non-Static methods or Instance methods :***

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

***are known as Non-Static methods or Instance methods.***

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

***object while object creation and can be accessed with object name.***

***Structure of Instance methods:***

***return\_type method\_name(Para\_list)***

***{***

***//method\_body***

***}***

***Coding rule:***

***=>These Instance methods will can access both variables,***

***static and Instance variables.***

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

***Ex-Program : DemoVariables.java***

***class DemoVariables***

***{***

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

***int b=20;***

***static void m1()***

***{***

***System.out.println("=====static m1()======");***

***System.out.println("The value of a:"+a);***

***//System.out.println("The value of b:"+b);***

***}***

***void m2()***

***{***

***System.out.println("=====Instance m2()=====");***

***System.out.println("The value of a:"+a);***

***System.out.println("The value of b:"+b);***

***}***

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

***{***

***int c=30;***

***DemoVariables.m1();***

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

***ob.m2();***

***System.out.println("====main()=====");***

***System.out.println("The value of c:"+c);***

***}***

***}***

***o/p:***

***=====static m1()======***

***The value of a:10***

***=====Instance m2()=====***

***The value of a:10***

***The value of b:20***

***====main()=====***

***The value of c:30***

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

***Dt : 19/8/2022***

***\*imp***

***JVM Internals with execution flow of program:***

***=>JVM Internally divided into the following Partitions:***

***1.Class Loader SubSystem***

***2.Runtime Data Area***

***3.Execution Engine***

***1.Class Loader SubSystem:***

***=>Class Loader SubSystem will load the class file onto***

***Runtime Data Area using Loader.***

***2.Runtime Data Area:***

***=>This Runtime Data Area internally divided into the following***

***partitions:***

***(a)Method Area***

***(b)Heap Area***

***(c)Java Stack Area***

***(d)PC Register Area***

***(e)Native Method Area***

***(a)Method Area:***

***=>The location where the class is loaded is known as Method***

***Area.***

***=>while class loading static members of class will get the***

***memory within the class.***

***=>Once main() got the memory within the class,then it is***

***automatically copied on JavaStackArea to start execution***

***process.***

***(b)Heap Area:***

***=>The location where the objects are created is known as***

***HeapArea.***

***(c)Java Stack Area:***

***=>The location where the methods are executed is known as***

***Java Stack Area.***

***=>main() is the first method copied onto JavaStackArea to***

***start the execution process.***

***=>main() will call other methods for execution.***

***faq:***

***define Method frame?***

***=>The partition of JavaStackArea where the method is copied***

***for execution is known as Method Frame.***

***=>After Method execution completed the method frame will be***

***destroyed automatically.***

***(d)PC Register Area:***

***=>Program Counter(PC) registers will record the status of***

***method executions in Java Stack Area.***

***=>Every method which is executing in Java Stack Area will have***

***its own PC-register.***

***=>All these PC-Registers are opened in a separate location***

***known as PC-Register Area.***

***Dt : 20/8/2022***

***(e)Native Method Area:***

***=>The methods which are declared with 'native' keyword in***

***JavaLib are known as Native methods.***

***=>These Native methods internally having c or c++ code.***

***=>when these methods are used in programs,then they are***

***separated and loaded onto separate location known as Native***

***method Area.***

***=>ExecutionEngine will execute Native methods using using***

***JNI(Java Native method Interface) and this JNI internally uses***

***Native method Libraries.***

***faq:***

***why JavaLib having Native methods?***

***=>The Native methods from JavaLib will support JavaPrograms to***

***interact with resources outside the JVM.***

***(resources means N/W,File,DB,...)***

***3.Execution Engine:***

***=>Execution Engine is an executor or processor of JVM,which***

***starts the execution process with main() method available in***

***Java Stack Area.***

***=>This Execution Engine internally having two translators:***

***(a)Interpreter***

***(b)JIT(Just-In-Time) Compiler***

***(a)Interpreter:***

***=>Interpreter will start the execution process and executes***

***normal instructions.***

***=>when Interpreter finds Stream instructions or MultiMedia***

***instructions then the execution control is transferred to the***

***JIT-Compiler.***

***(b)JIT(Just-In-Time) Compiler:***

***=>JIT Compiler will execute Stream instructions or MultiMedia***

***instructions like Audio,Video,Image and Animation files.***

***faq:***

***why Interpreter in execution process?***

***=>when we have interpreter in execution process then we can***

***accept the request in the middle of execution process and which***

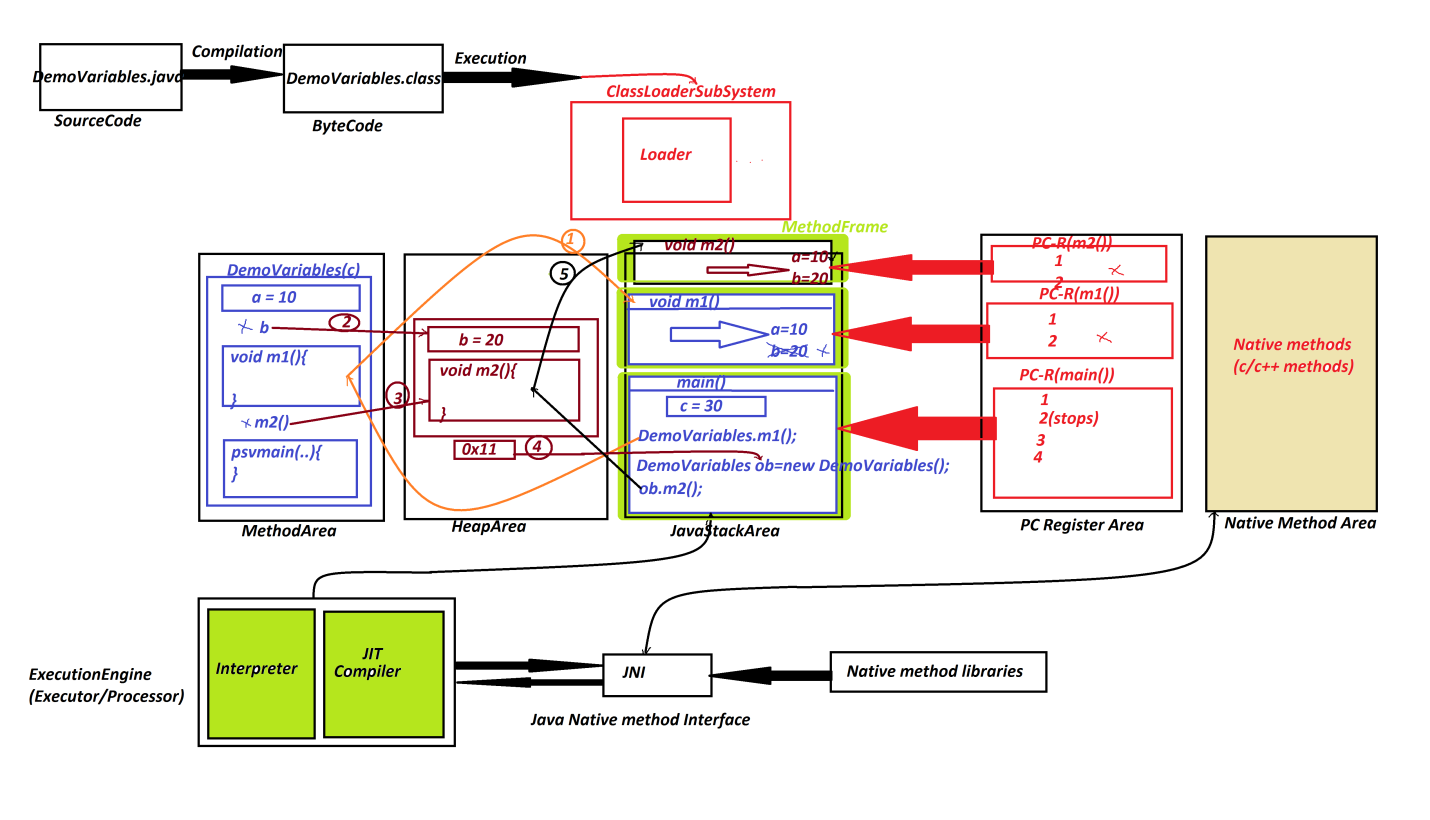
***is preferable in Server application development.***

***Note:***

***=>when we have Interpreter in ExecutionProcess,then JavaLang***

***can also be called as Interpreted Language.***

***Diagram:***

******

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

***Summary:***

***=>MethodArea where classes are loaded.***

***=>HeapArea where Objects are created.***

***=>JavaStackArea where methods are executed.***

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

***faq:***

***define parameters?***

***=>parameters are the variables which are used to transfer the***

***data from one method to another method.***

***=>Based on parameters the methods are categorized into two***

***types:***

***1.Methods without parameters***

***2.Methods with parameters***

***1.Methods without parameters:***

***=>The methods which are declared without parameters are***

***known as 0-parameter methods or Methods without parameters.***

***2.Methods with parameters:***

***=>The methods which are declared with parameters are known as***

***Parameterized methods or Methods with parameters.***

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

***faq:***

***define return\_type?***

***=>return\_type specify the method will return the value or not.***

***=>Based on return\_type methods are categorized into two types:***

***1.Non-return type methods***

***2.Return type methods***

***1.Non-return type methods:***

***=>The methods which will not return any value after execution***

***are known as Non-return type methods.***

***=>The methods which are declared with 'void' are known as***

***Non-return\_type methods.***

***2.Return type methods:***

***=>The methods which return the value after execution are known***

***as Return type methods.***

***=>The methods which are declared without 'void' are known as***

***Return type methods.***

***=>we use 'return' statement to return the value after method***

***execution.***

***=>The returned value will come back to the method call.***

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