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

***Java : language***

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

***Process of Learning Language:***

***step-1 : Alphabets***

***step-2 : Grammer***

***step-3 : Construction rules***

***Note:***

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

***Construction rules***

***Java:***

***part-1 : CoreJava***

***part-2 : AdvJava***

***part-1 : CoreJava:***

***(i)Programming Components(Java Alphabets)***

***(ii)Programming Concepts***

***(iii)Object Oriented Programming features.***

***(i)Programming Components(Java Alphabets)***

***1.Variables***

***2.Methods***

***3.Blocks***

***4.Constructors***

***5.Classes***

***6.Interfaces***

***7.AbstractClasses***

***(ii)Programming Concepts***

***1.Object Oriented programming***

***2.Exception Handling process***

***3.Multi-Threading process***

***4.Java Collection Framework(JCF)***

***=>Data Structures in Java***

***5.Files in Java***

***6.Networking in Java***

***(iii)Object Oriented Programming features.***

***1.Class***

***2.Object***

***3.Abstraction***

***4.Encapsulation***

***5.PolyMorphism***

***6.Inheritance***

***Note:***

***=>Using CoreJava Alphabets and Concepts we can develop StandAlone applications.***

***faq:***

***define StandAlone Applications?***

***=>The applications which are installed in one computer performs actions in the same computer are known as StandAlone applications or DeskTop applications or Windows Aplications.***

***Note:***

***=>According to developed StandAlone applications means,***

***No HTML input***

***No Server Environment***

***part-2 : AdvJava***

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

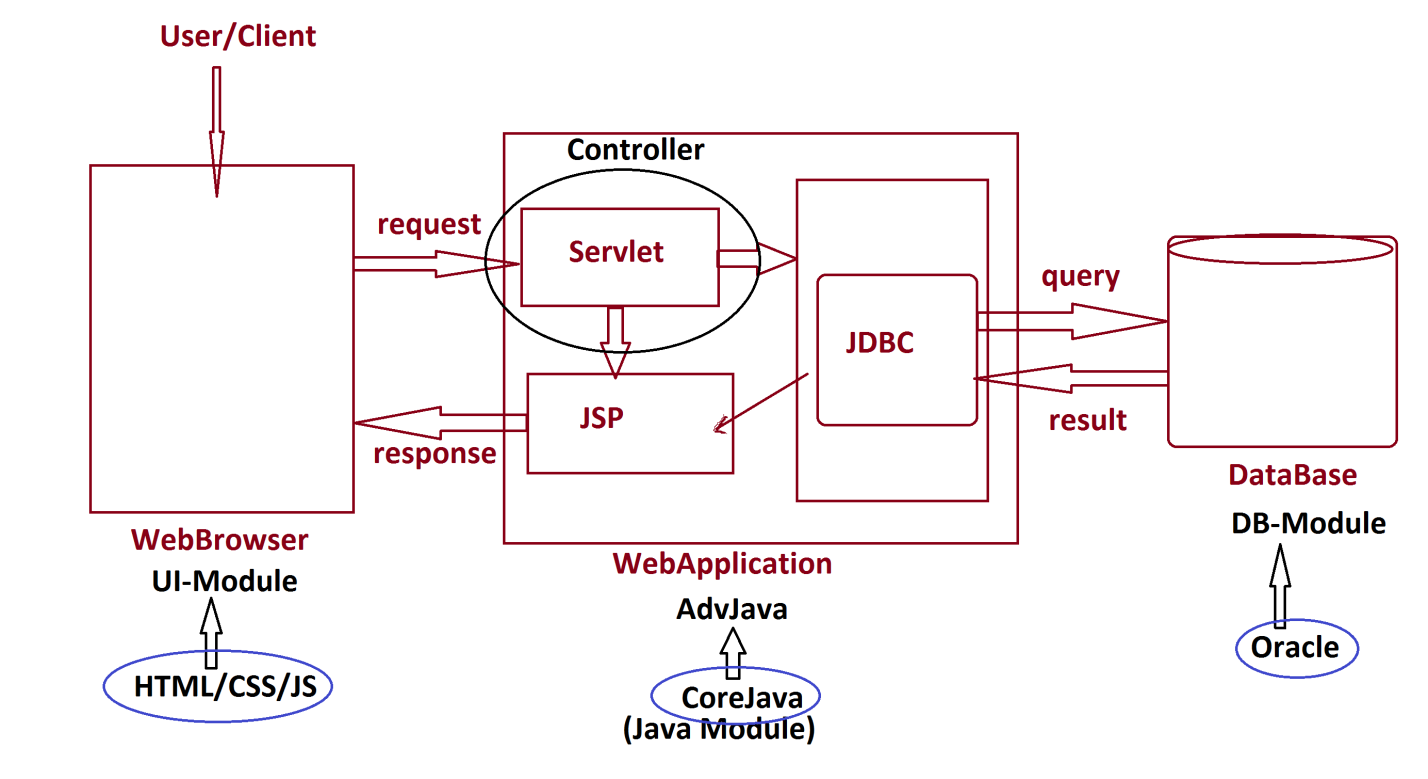
***WebApplications:***

***(a)JDBC***

***(b)Servlet***

***(c)JSP***

***Diagram:***

******

***(a)JDBC:***

***=>JDBC stands for Java DataBase Connectivity and which is used to interact with DataBase.***

***(b)Servlet:***

***=>Servlet means 'Server program' and which is used to accept the request from users through WebBrowser.***

***(c)JSP:***

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

***faq:***

***define Web Applications?***

***=>The applications which are executed in WebEnvironment or Internet Environment are known as WebApplications.***

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

***Programming Module***

***(i)CoreJava - StandAlone application development***

***(ii)AdvJava - WebApplication development***

***(iii)Spring and WebServices - Enterprise Applications***

***(Project)***

***faq:***

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

***(i)Language***

***(ii)Technology***

***(iii)Framework***

***(i)Language:***

***=>Language will provide Programming components and Concepts used in constructing applications.***

***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***

***WebService***

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

***faq:***

***define Enterprise Applications?***

***=>The application which is executing in distributed environment and depending on the features like 'Security','Load balancing' and 'Clustering' is known as Enterprise application or Enterprise distributed***

***application.***

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

***faq:***

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

***(i)JDK***

***(ii)JRE***

***(i)JDK:***

***=>JDK stands for 'Java development Kit' and which provide the following:***

***(a)Java Compiler***

***(b)Java Library***

***(c)JVM***

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

***=>Java Compiler will compile the source code and generate***

***Byte code.***

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

***=>Java Library will hold Pre-defined programming components which can be used in application development.***

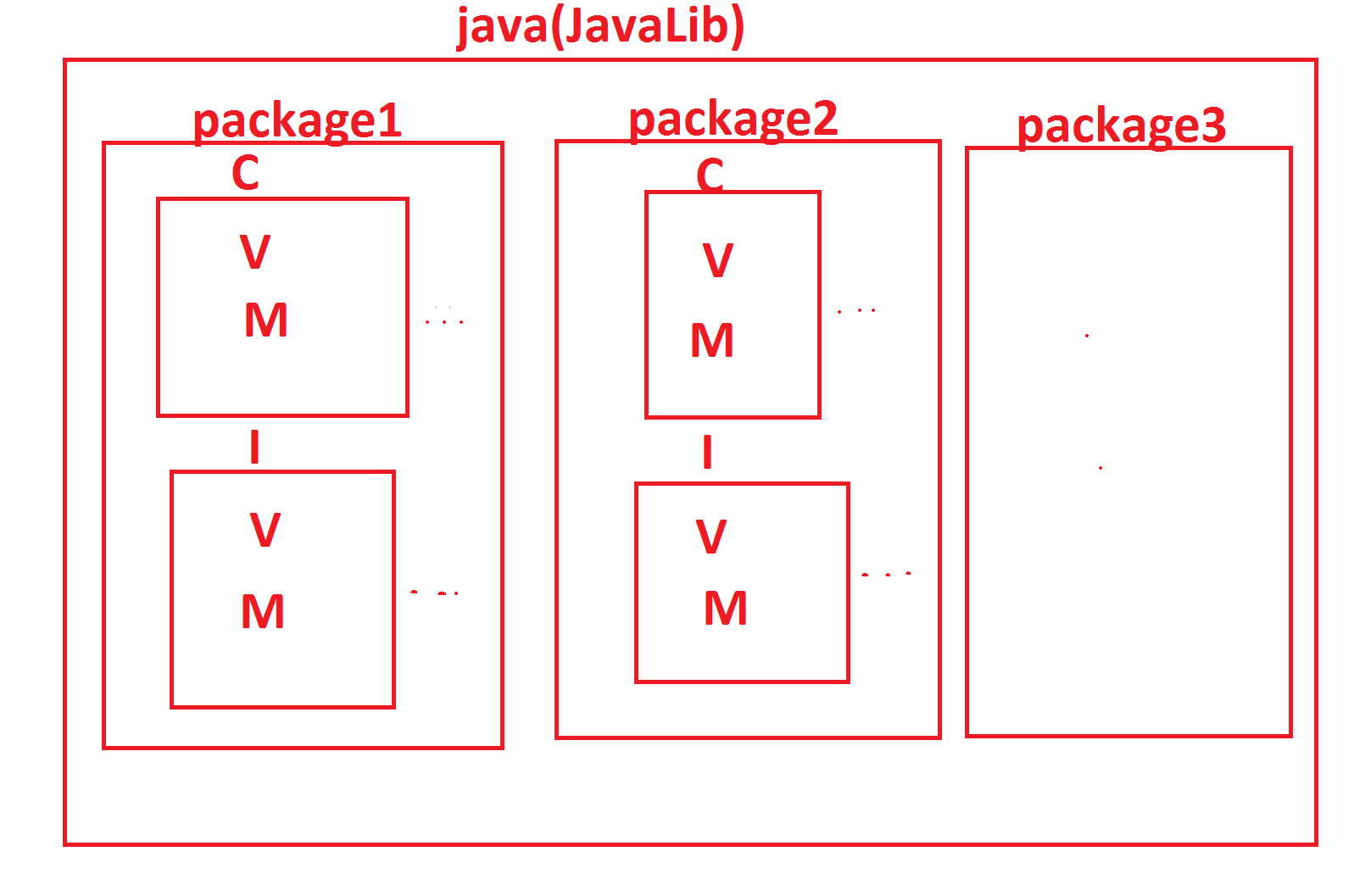
***=>we use 'java' word to represent JavaLibrary.***

***=>JavaLib will hold 'packages'.***

***=>packages will hold 'Classes and Interfaces'.***

***=>Classes and Interfaces will hold 'Variables and Methods'***

***Diagram:***

******

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

***CoreJava packages:***

***(i)java.lang - Language package***

***(ii)java.util - Utility package***

***(iii)java.io - Streams and Files package***

***(iv)java.net - Networking package***

***AdvJava packages:***

***(i)java.sql - DataBase connection package***

***(ii)javax.servlet - Servlet Programming package***

***(iii)javax.servlet.jsp - JSP programming package***

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

***\*imp***

***(c)JVM:***

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

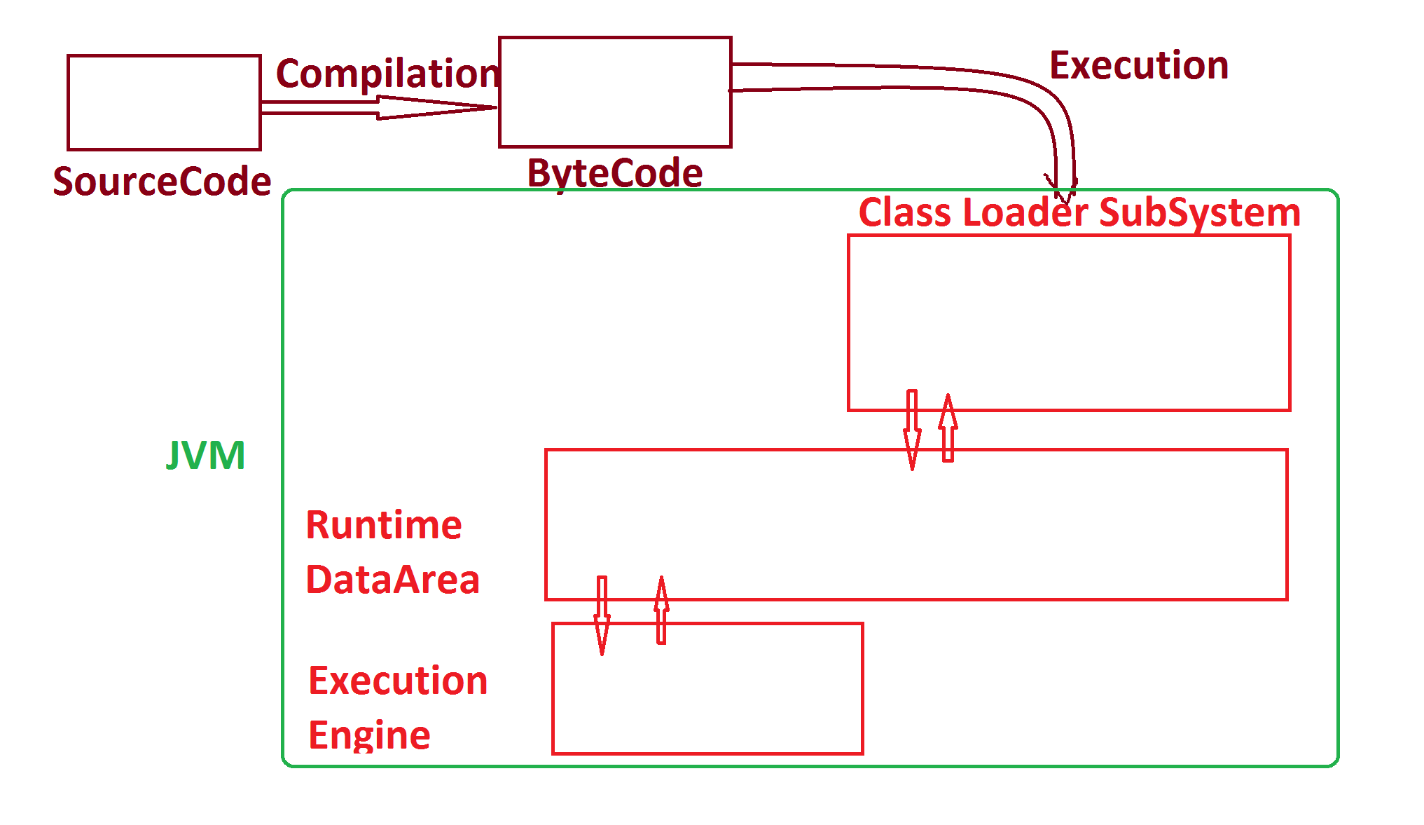
***=>JVM internally having the following three partitions:***

***(i)Class Loader SubSystem***

***(ii)Runtime Data Area***

***(iii)Execution Engine***

***Diagram:***

******

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

***faq:***

***define Virtual Machine?***

***=>The s/w component which internally having the behaviour like machine is known Virtual Machine.***

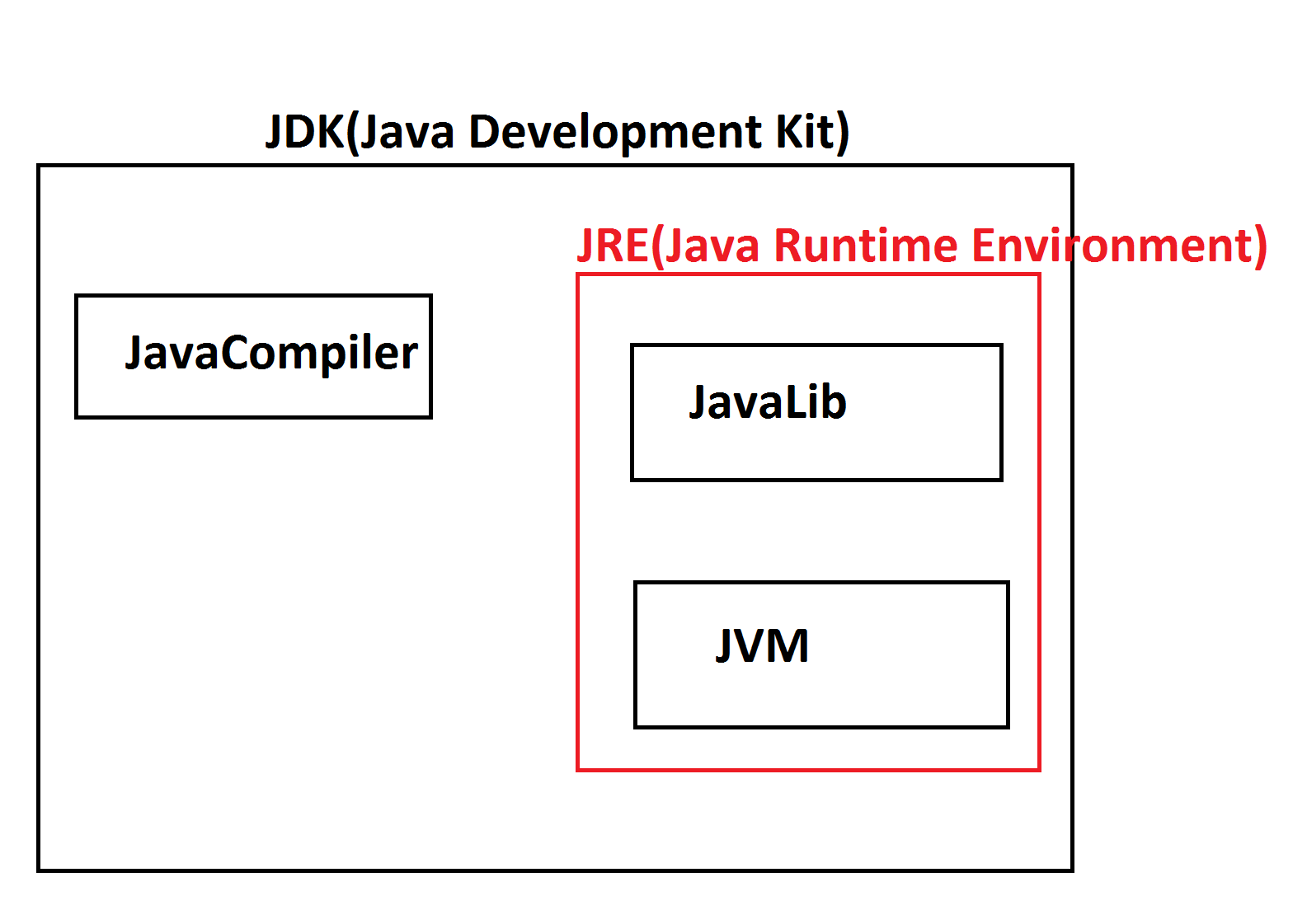
***(ii)JRE:***

***=>JRE stands for 'Java Runtime Environment' and which provide 'JavaLib and JVM'.***

***=>JRE is inbuilt in JDK,which means part of JDK.***

***=>On demand we can download JRE separatly and which can be used part of WebServers like Tomcat.***

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

******

***Writing,Saving,Compiling and Execution Java Program:***

***define Program?***

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

***define Programming?***

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

***Note:***

***=>The process of generating Source Code from 'project Analysis and project design' is known as Programming or Coding.***

***define Programmer?***

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

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

***\*imp***

***Stages of Program:***

***=>After writing the program,we 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***

***2.Execution***

***1.Compilation:***

***=>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***

***Note:***

***=>C,C++ and Java languages will use Compiler in Compilation stage.***

***2.Execution:***

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

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

***Note:***

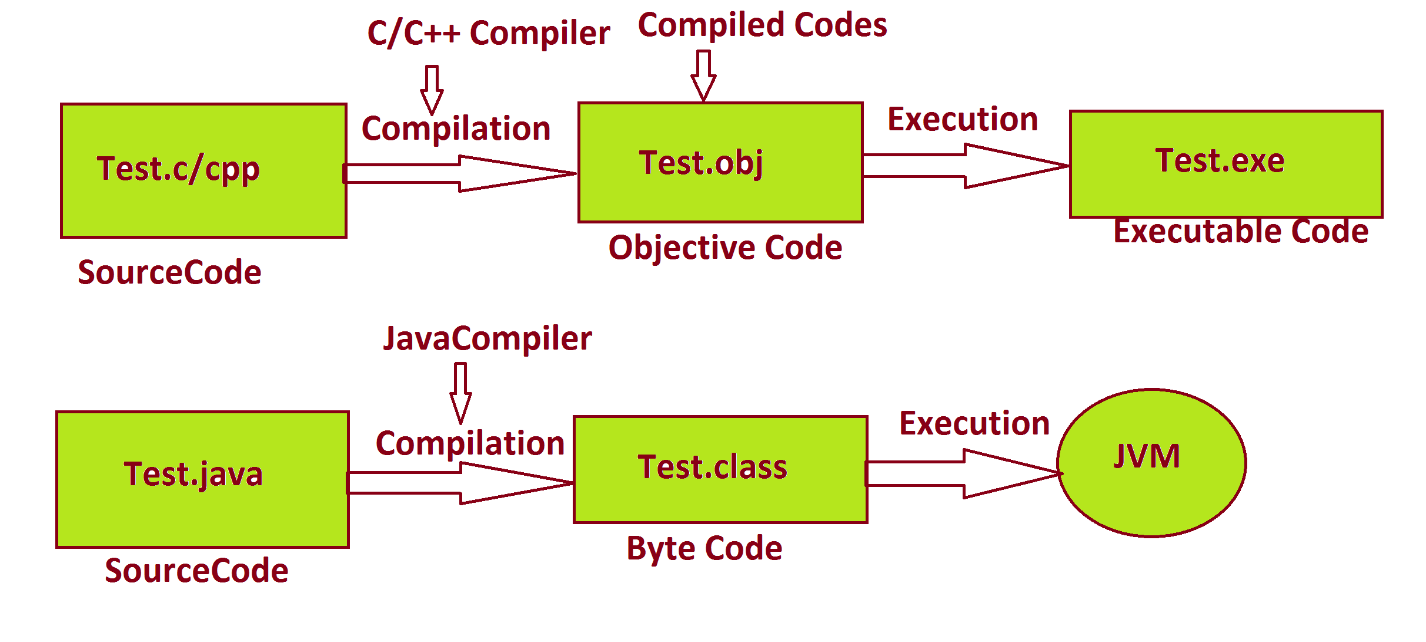
***=>In c and c++ languages the Objective Code is converted into***

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

***=>In Java Language the Byte Code is executed on JVM(Java Virtual***

***Machine).***

***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 Operating System is participated, because of this reason Objective Code is PlatForm dependent code.***

***DisAdvantage:***

***=>The Objective Code 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 ByteCode generation Operating System is not participated because of this reason, Byte Code is PlatForm independent code.***

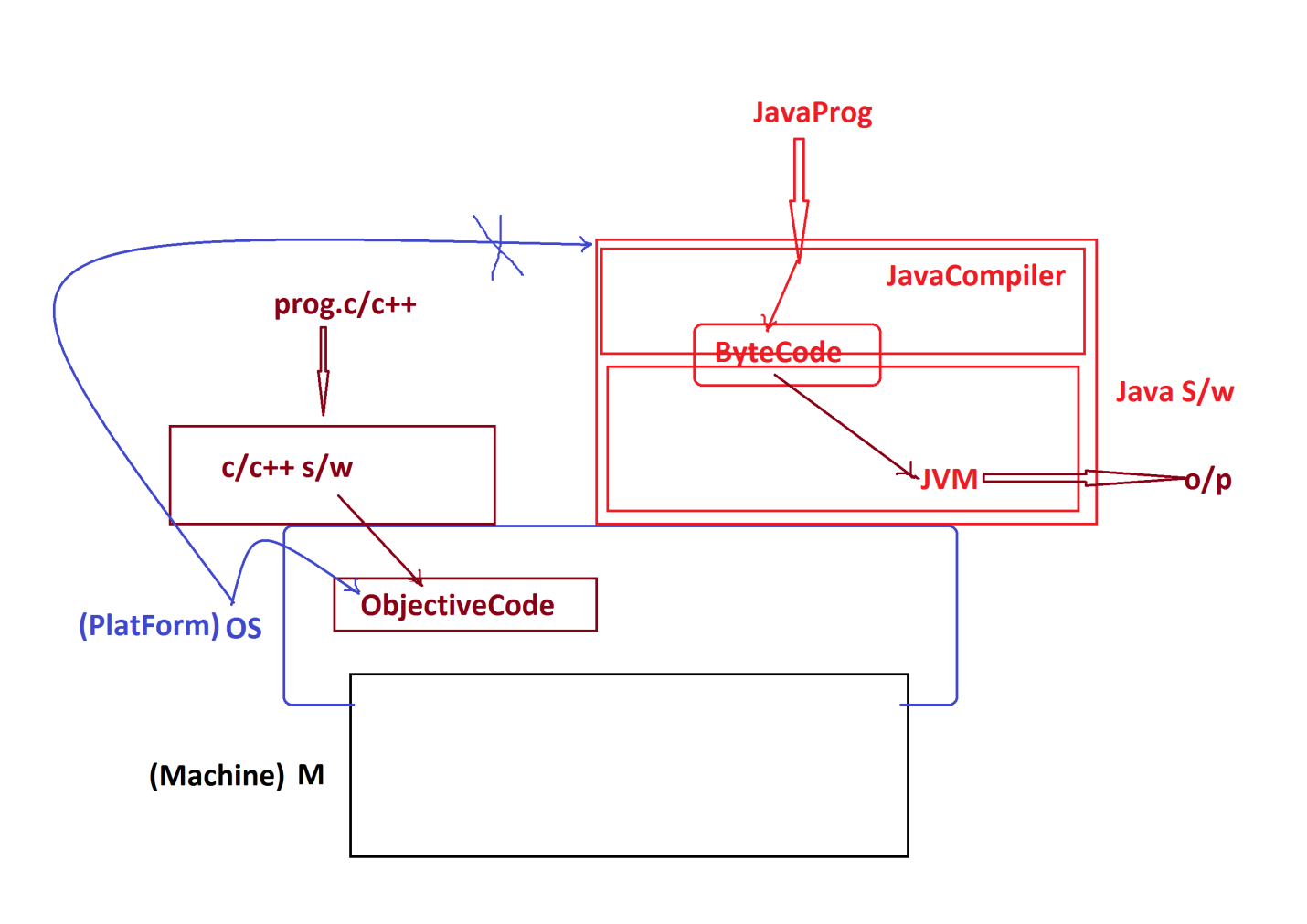
***Advantage:***

***=>The ByteCode generated from One platform can be executed on all platform where JVM is available.***

***Note:***

***=>JavaLanguage which is generating ByteCode is PlatForm independent Language.***

***Diagram:***

******

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

***faq:***

***define High Level language?***

***=> The Language programs which are constructed from User understandable formats, is known as High Level Language.***

***faq:***

***define Low Level Language ?***

***=>The Language programs which are constructed from the formats which are not understandable by the users, is known as Low Level Language.***

***faq:***

***define Translators?***

***=>Translators are used to translate 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:***

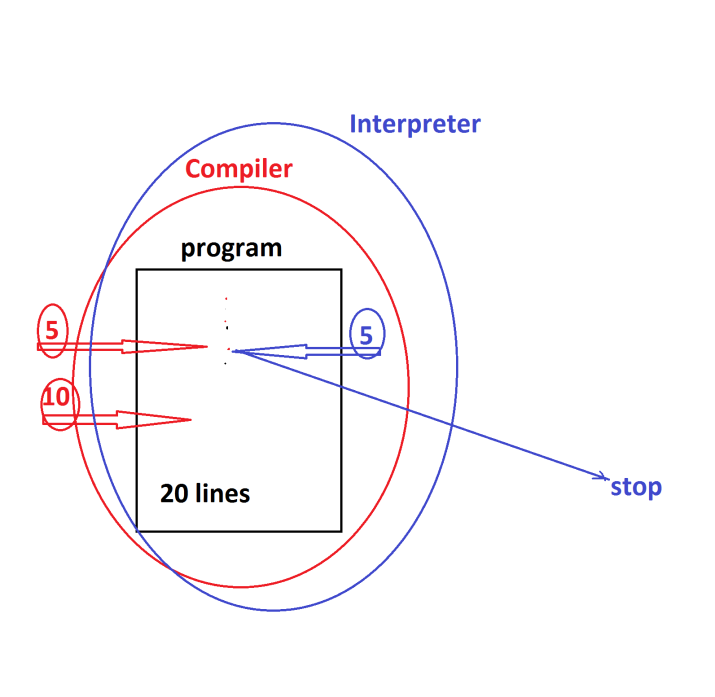
***(i)Compiler***

***=>Compiler translates the total program at-a-time.***

***(ii)Interpreter***

***=>Interpreter translates the program line-by-line.***

***Diagram:***

******

***Note:***

***=>C,C++ and Java Languages will use compiler in Compilation process.***

***=>C and C++ languages will use Compiler in execution process,but Java language will use "Interpreter and Compiler" in execution process.***

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

***"James Gosling"***

***Sun Micro Systems - 1991***

***=>Code Writer(Programmer)***

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

***Test.gt(green talk)***

***OAK Language***

***Java - 1995***

***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***

***=>JDK 1.5***

***=>JRE 1.5***

***2006 - Java6***

***=>JDK 1.6***

***=>JRE 1.6***

***2011 - Java7***

***=>JDK 1.7***

***=>JRE 1.7***

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

***2014 - Java8***

***=>JDK 1.8***

***=>JRE 1.8***

***2017 - Java9***

***=>JDK 1.9***

***=>JRE 1.9***

***2018 - Java10 and Java11***

***2019 - Java12 and Java13***

***2020 - Java14 and Java15***

***2021 - Java16 and Java17***

***2022 - Java18***

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

***Installing Java S/w(JDK) and Setting path in Environment***

***Variables:***

***step-1 : Identifying the System Environment where the JDK is Installed***

***(JDK is PlatForm dependent while installation process)***

***step-2 : DownLoad JDK from Oracle WebSite.***

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

***step-3 : Install JDK***

***Note:***

***=>After installation process is Successfull,we can find the folder with name "java" in ProgramFiles***

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

***step-4 : Set Java-path 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-17.0.1\bin;***

***step-5 : Click 'ok' for three times,then path is setted.***

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

***Note:***

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

***Working or not:***

***javac - Compilation Command***

***java - Execution Command***

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

***\*imp***

***define 'class'?***

***=>class is a 'structured layout' in java generating objects.***

***=>class in java can hold Variables,methods(functions) and***

***main()***

***Variables - are data holders***

***Methods - are the actions***

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

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

***Structure of class in Java:***

***class Class\_name***

***{***

***//Variables***

***//methods***

***//main()***

***}***

***Note:***

***(i)In classes the starting letter must be Capital according to real time coding standards***

***(ii)In Java main() is having the following Pre-defined***

***Standard format:***

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

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

***Ex\_program-1 :***

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

***class Display***

***{***

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

***{***

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

***}***

***}***

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

***step-1 : Create folder in any drive***

***E:\Demo132***

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

***step-3 : Save the program with Language extention in Folder***

***Click on File->Save->Browse and select folder->name the file as "Display.java" and click 'save'.***

***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***

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

***Note:***

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

***process.***

***=>To open CommandPrompt,GoTo folder->***

***type 'cmd' in Address Bar and press 'enter'.***

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

***Ex\_program-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("The value of a="+a);***

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

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

***}***

***}***

***o/p:***

***The value of a=12***

***The value of b=13***

***The sum=25***

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

***Note:***

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

***with result.***

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

***faq:***

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

***(i)print()***

***(ii)println()***

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

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

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

***Assignment-1:***

***wap to evaluate the following expression?***

***d=a\*b-(x+y)\*z;***

***Sop(a)***

***Sop(b)***

***Sop(x)***

***Sop(y)***

***Sop(z)***

***Sop(d)***

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

***Assignment-2:***

***wap to calculte the percentage of Student?***

***s1=***

***s2=***

***s3=***

***s4=***

***s5=***

***s6=***

***totMarks = s1+s2....;***

***per =***

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

***faq:***

***define Environment Variables?***

***=>The variables part of Operating System holding the information about the resources installed in Computer System are known as Environment Variables.***

***=>These Environment Variables are categorized into two types:***

***(i)User Variables***

***(ii)System Variables***

***(i)User Variables:***

***=>The information in User variables can be used by only***

***individual user.***

***(ii)System Variables:***

***=>The information in System Variables can be used by all***

***the multiple users of Computer System.***

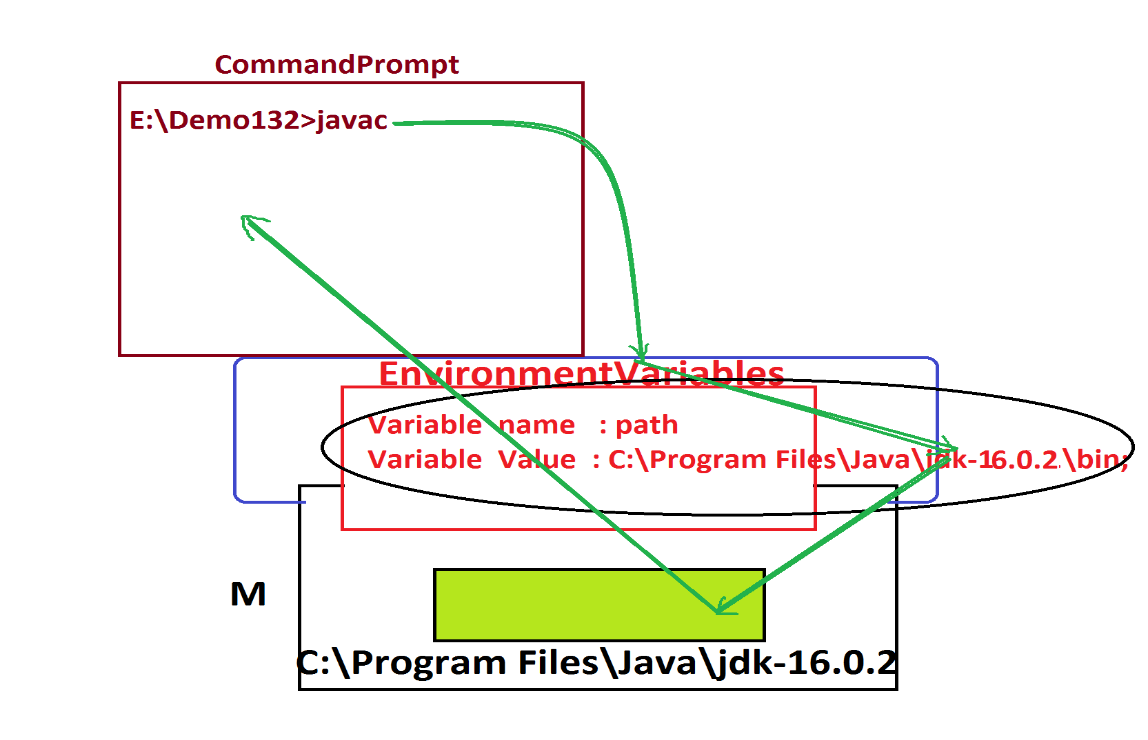
***faq:***

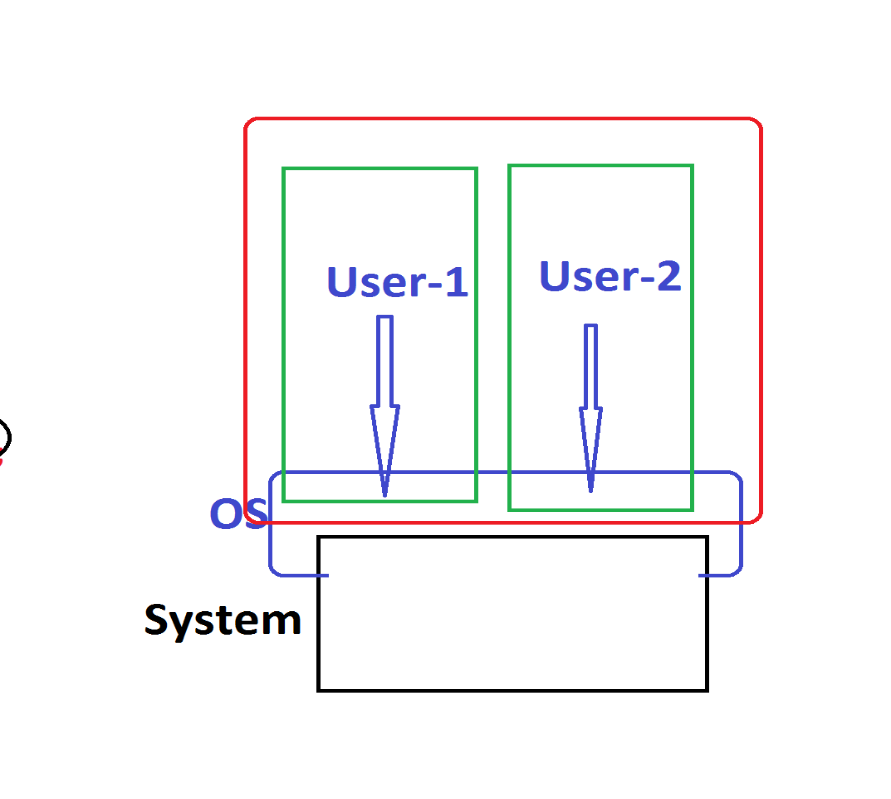
***wt is the advantage of having JavaPath in Environment***

***Variables?***

***=>when we have JavaPath in EnvironmentVariables,then we can compile and execute JavaPrograms from any location of computer System.***

***Diagram:***

******

******

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

***\*imp***

***DataTypes,Variables and Methods:***

***DataTypes in Java:***

***=>Types of data which we are expecting as input to JavaProgram 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 or Fundamental DataTypes or Basic 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 are known as Integer***

***datatypes.***

***=>Types:***

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

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

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

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

***Note:***

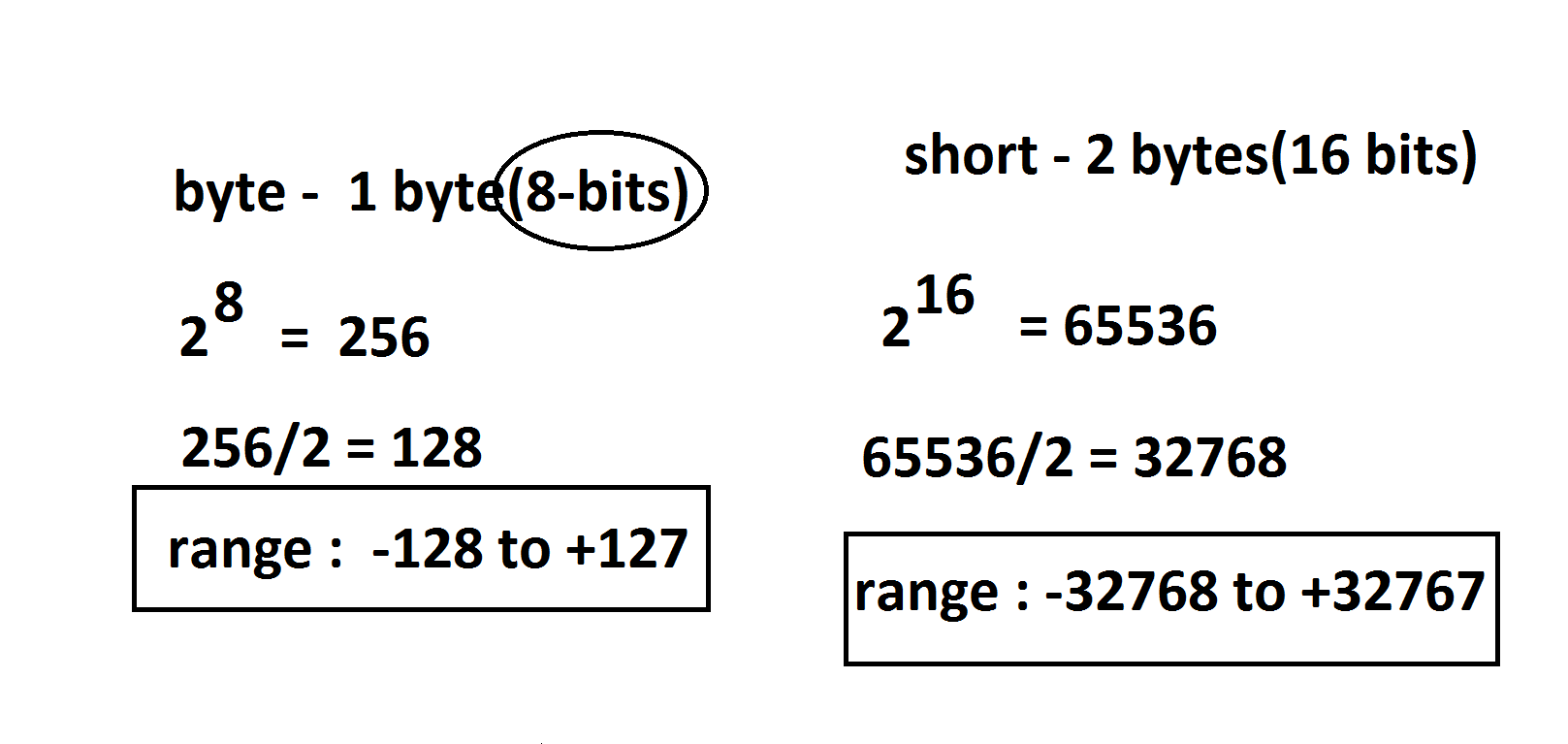
***=>byte and short datatypes are used for Stream data.***

***(MultiMedia data)***

***=>int datatype is used in normal programming process.***

***=>long datatype is to hold big integer values like PhoneNo, AccNo, CardNo,...***

***=>In Java,when we want to assign long value we must use "L" or "l" in the RHS of declaration.***

******

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

***=>The numeric data with decimal point are known as Float datatypes.***

***=>Types:***

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

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

***Note:***

***=>float datatype is used in normal programming process.***

***=>double datatype is used to hold biggest float values.***

***=>In Java,when we want to assign float value we must use "F" or "f" in the RHS of declaration.***

***Summary:***

***=>According to compilation process,***

***=>The numeric value without decimal point automatically considered as 'int' datatype.***

***=>The numeric value with decimal point automatically considered as 'double' value.***

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

***=>The single valued character which is represented in single quotes is known as Character datatype.***

***Ex:***

***'k','n',...***

***=>Types:***

***char - 2 bytes***

***(d)boolean datatype:***

***=>The datatype which is available in the form of true or false is known as boolean datatype.***

***=>Types:***

***boolean - 1 bit***

***Ex\_Program : DataTypes.java***

***class DataTypes***

***{***

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

***{***

***byte b = 127;***

***short s = 32767;***

***int i = 641234;***

***long l = 9898981234L;***

***float f = 12.34F;***

***double d = 2345.67;***

***char ch = 'A';***

***boolean bl = true;***

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

***System.out.println("short value = "+s);***

***System.out.println("int value = "+i);***

***System.out.println("long value = "+l);***

***System.out.println("float value = "+f);***

***System.out.println("double value = "+d);***

***System.out.println("char value = "+ch);***

***System.out.println("boolean value = "+bl);***

***}***

***}***

***o/p:***

***byte value = 127***

***short value = 32767***

***int value = 641234***

***long value = 9898981234***

***float value = 12.34***

***double value = 2345.67***

***char value = A***

***boolean value = true***

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

***faq:***

***define TypeCasting?***

***=>The process of converting one data type value into another datatype value is known as Type Casting process.***

***=>TypeCasting process can be done in two ways:***

***(a)Widening process***

***(b)Narrowing process***

***(a)Widening process:***

***=>The process of converting Lower datatypes into Higher data types is known as Widening process.***

***char->byte->short->int->long->float->double***

***=>This Widening process is also known as UpCasting process or Implicit TypeCasting process***

***(b)Narrowing process:***

***=>The process of converting Higher datatypes into Lower data types is known as Narrowing process.***

***double->float->long->int->short->byte->char***

***=>This Narrowing process is also known as DownCasting process or Explicit TypeCasting process.***

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

***Assignment-2:(Solution)***

***wap to calculte the percentage of Student?***

***Program : StudentResult.java***

***class StudentResult***

***{***

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

***{***

***int s1=77,s2=67,s3=57,s4=73,s5=87,s6=72;***

***int totMarks = s1+s2+s3+s4+s5+s6;***

***float per=(float)totMarks/6;***

***System.out.println("s1="+s1);***

***System.out.println("s2="+s2);***

***System.out.println("s3="+s3);***

***System.out.println("s4="+s4);***

***System.out.println("s5="+s5);***

***System.out.println("s6="+s6);***

***System.out.println("TotMarks="+totMarks);***

***System.out.println("Percentage="+per);***

***}***

***}***

***o/p:***

***s1=77***

***s2=67***

***s3=57***

***s4=73***

***s5=87***

***s6=72***

***TotMarks=433***

***Percentage=72.166664***

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

***Assignment:***

***wap to calculate and display totSal of an emmployee based onbSal?***

***bSal = 12000***

***totSal = bSal+HRA+DA;***

***HRA = 93% of bSal***

***DA = 63% of bSal***

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

***\*imp***

***2.Non-Primitive DataTypes:***

***=>The 'Group Valued data formats' are known as Non-Primitive datatypes or Referential DataTypes.***

***=>These Non-Primitive DataTypes are categorized into following:***

***(a)Class***

***(b)Interface***

***(c)Array***

***(d)Enum***

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

***\*imp***

***Object Oriented Programming:***

***=>The process of Constructing programs using Class-Object concept is known as Object Oriented Programming.***

***=>In Object Oriented Programming we control and manage NonPrimitive DataTypes.(we work with NonPrimitive DataTypes)***

***(a)Class***

***(b)Interface***

***(c)Array***

***(d)Enum***

***(a)Class:***

***=>Class in Java is a 'Structured Layout' generating objects.***

***=>Class in Java can hold Variables,methods and main()***

***(Class in Java can hold 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 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***

***Ex:***

***Display***

***Addition***

***DataTypes***

***StudentResult***

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

***\*imp***

***define 'Object'?***

***=>Object is a Storage(memory) related to a class holding the***

***members of class.***

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

***syntax of Object creation:***

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

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

***\*imp***

***Variables in Java:***

***=>variables are the data holders which hold data in the programs.***

***=>Based on DataTypes the variables are categorized into two types:***

***(i)Primitive DataType variables***

***(ii)NonPrimitive DataType variables***

***(i)Primitive DataType variables:***

***=>The variables which are declared with Primitive DataTypes like byte,short,int,long,float,double,char and boolean are known as Primitive DataType Variables.***

***=>These Primitive DataType variables will hold Values.***

***(ii)NonPrimitive DataType variables:***

***=>The variables which are declared with NonPrimitive DataTypes like Class, Interface, Array and Enum are known as NonPrimitive DataType variables.***

***=>These NonPrimitive DataType variables will hold Object references.(Object Addresses)***

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

***=>Based on 'static' keyword the variables are categorized into two types:***

***(a)Static variables***

***(b)NonStatic Variables***

***(a)Static variables:***

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

***(b)NonStatic Variables:***

***=>The variables which are declared without 'static' are known NonStatic Variables.***

***=>These NonStatic variables are categorized into two types:***

***(i)Instance variables***

***(ii)Local Variables***

***(i)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.***

***(ii)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.***

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

***Program : DemoVariables.java***

***class DemoVariables***

***{***

***static int a=12;***

***int b=13;***

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

***{***

***int c=14;***

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

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

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

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

***}***

***}***

***o/p:***

***The value a=12***

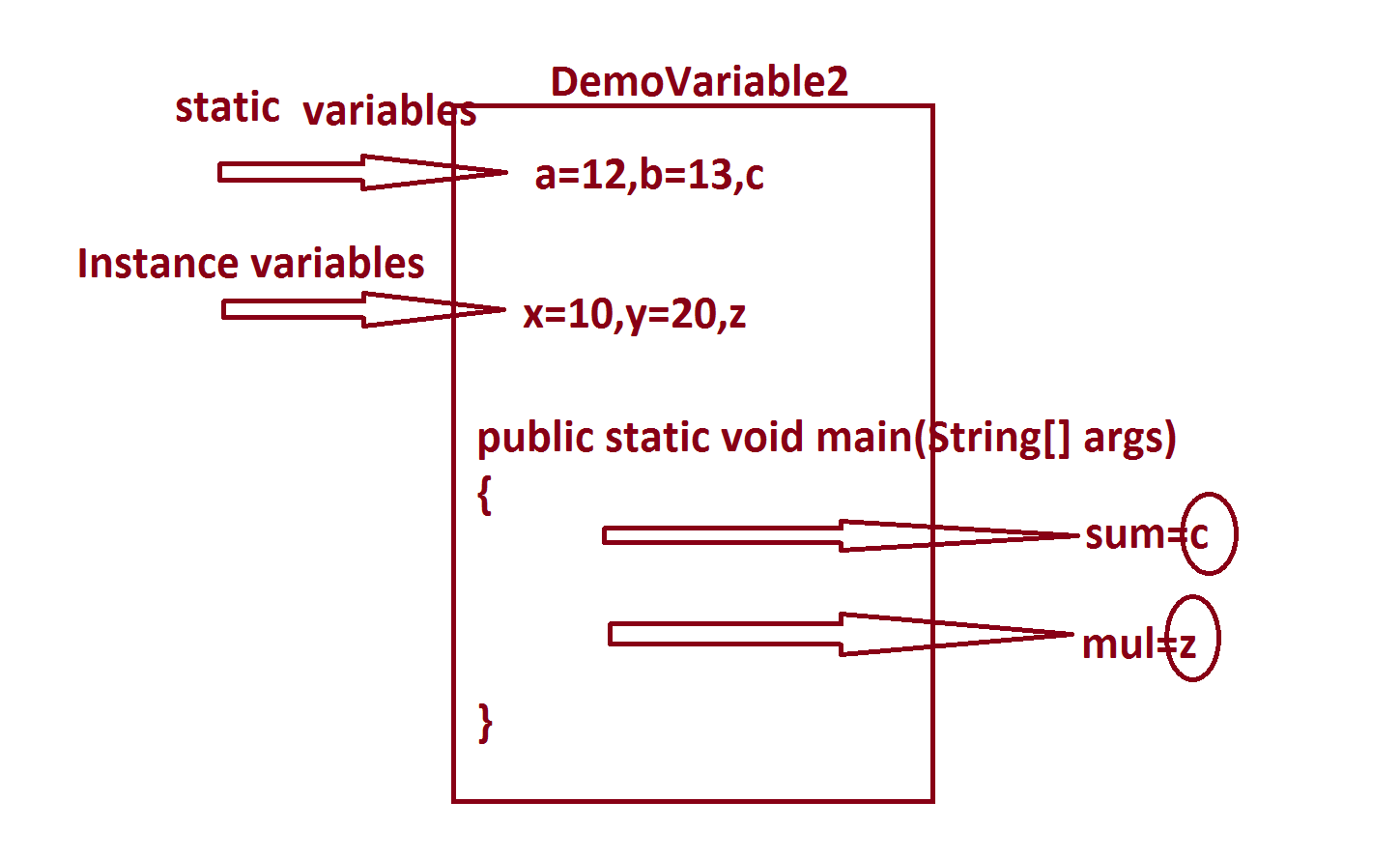
***The value b=13***

***The value c=14***

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

***Assignment:***

***Layout:***

******

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

***Assignment:(Solution)***

***Program : DemoVariables2.java***

***class DemoVariables2***

***{***

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

***int x=10,y=20,z;***

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

***{***

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

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

***DemoVariables2 ob = new DemoVariables2();***

***ob.z = ob.x\*ob.y;***

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

***}***

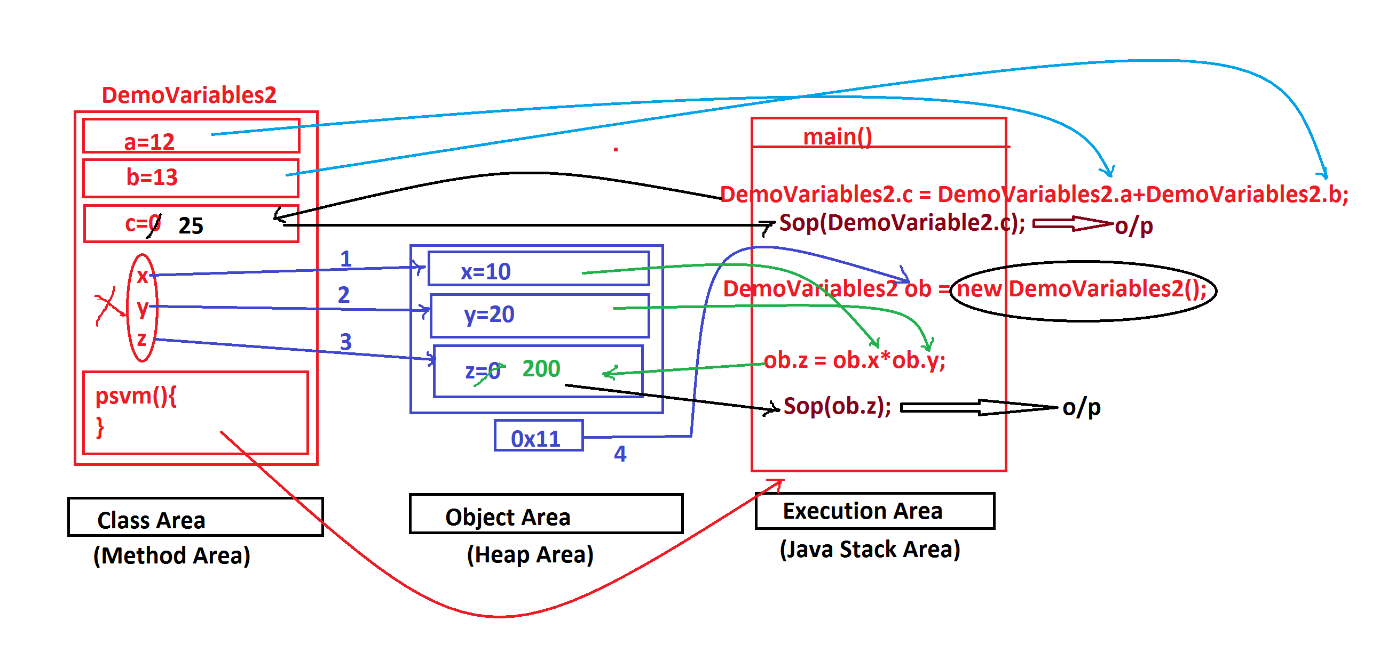
***}***

***o/p:***

***Sum=25***

***Mul=200***

***Diagram:(Demonstrating the Memory Location of Variables)***

******

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

***\*imp***

***Methods in Java:***

***=>Methods are the actions which are performed on data to generate***

***results.***

***=>Based on static keyword the methods are categorized into two***

***types:***

***1.Static methods***

***2.NonStatic methods(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***

***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, cannot access Instance variables directly.***

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

***=>Static methods are categorized into two types:***

***(i)Pre-defined static methods***

***(ii)User defined static methods***

***(i)Pre-defined static methods:***

***=>The static methods which are available from JavaLib are known as Pre-Defined static methods or Built-in static methods***

***(ii)User defined methods:***

***=>The static methods which are defined by the user are known as User defined static methods.***

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

***2.NonStatic methods(Instance methods):***

***=>The methods which are declared without static keyword are known as NonStatic methods or Instance methods or Object 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:***

***=>Instance methods can access both static and Instance variables directly,because Instance method belongs to Object and Object belongs to class.***

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

***=>These Instance methods are categorized into two types:***

***(i)Pre-defined Instance methods***

***(ii)User defined Instance methods***

***(i)Pre-defined Instance methods:***

***=>The Instance methods which are available from JavaLib are known as Pre-defined Instance methods or Built-In instance methods***

***(ii)User defined Instance methods:***

***=>The Instance methods which are defined by the programmer(user) are known as User defined Instance methods.***

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

***Ex\_Program : (Demonstrating methods)***

***class DemoMethods1***

***{***

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

***int b=20;***

***void m1()***

***{***

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

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

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

***}***

***static void m2()***

***{***

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

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

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

***}***

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

***{***

***DemoMethods1 ob = new DemoMethods1();***

***ob.m1();***

***DemoMethods1.m2();***

***}***

***}***

***o/p:***

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

***The value a=10***

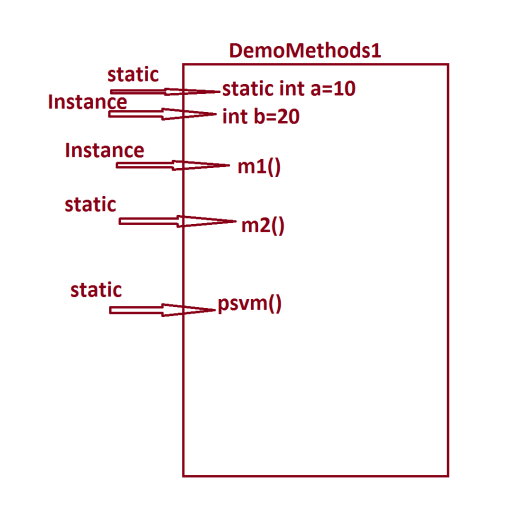
***The value b=20***

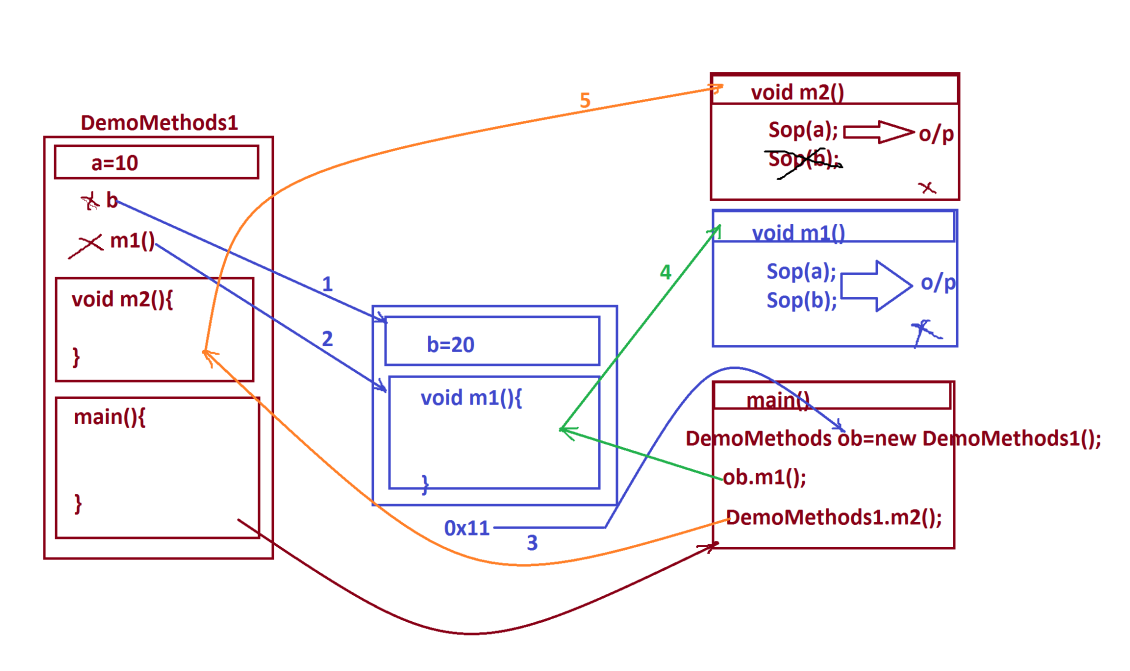
***====m2()====***

***The value a=10***

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

***Diagram:(Demonstrating the Location of methods);***

******

******

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

***define return\_type?***

***=>return\_type specify the methods will return the value after method execution or not***

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

***(i)Non-return\_type methods***

***(ii)return\_type methods***

***(i)Non-return\_type methods:***

***=>The methods which will not return any value after method execution are known as Non-return\_type methods.***

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

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

***Ex:***

***m1(),m2() methods are Non-return\_type methods***

***(ii)return\_type methods:***

***=>The methods which return the value after method execution are known as return\_type methods.***

***=>The methods which are declared without 'void' are know as return\_type methods.('void' is replaced with datatype)***

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

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

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

***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:***

***(i)Methods without parameters***

***(ii)Methods with parameters***

***(i)Methods without parameters:***

***=>The methods which are declared without parameters are known as Zero Parameter methods or Methods without parameters.***

***Ex:***

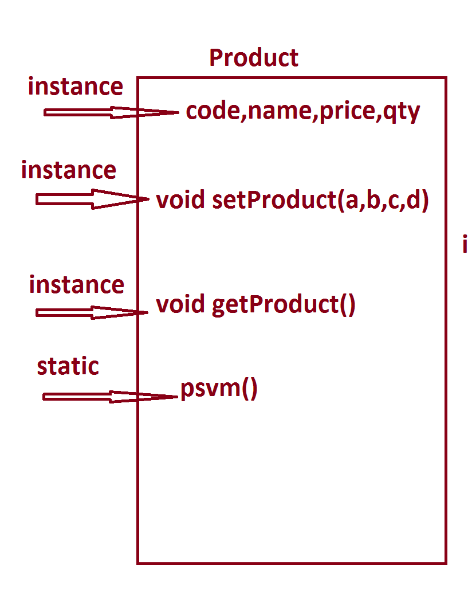
***m1() and m2() are Zero parameter methods.***

***(ii)Methods with parameters:***

***=>The methods which are declared with parameters are known as Parameterized methods or Methods with parameters.***

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

***Ex\_Program:(Demonstrating methods with parameters)***

******

***Program : Product.java***

***class Product***

***{***

***int code;***

***String name;***

***float price;***

***int qty;***

***void setProduct(int a,String b,float c,int d)***

***{***

***code=a;***

***name=b;***

***price=c;***

***qty=d;***

***}***

***void getProduct()***

***{***

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

***System.out.println("ProdCode="+code);***

***System.out.println("ProdName="+name);***

***System.out.println("ProdPrice="+price);***

***System.out.println("ProdQty="+qty);***

***}***

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

***{***

***int pC = 123;***

***String pN = "Mouse";***

***float pP = 123.45F;***

***int pQ = 12;***

***Product p = new Product();***

***p.setProduct(pC,pN,pP,pQ);***

***p.getProduct();***

***}***

***}***

***o/p:***

***====ProductDetails===***

***ProdCode=123***

***ProdName=Mouse***

***ProdPrice=123.45***

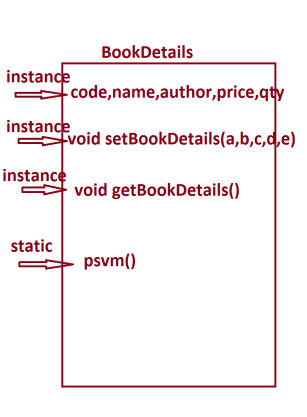
***ProdQty=12***

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

***Assignment:***

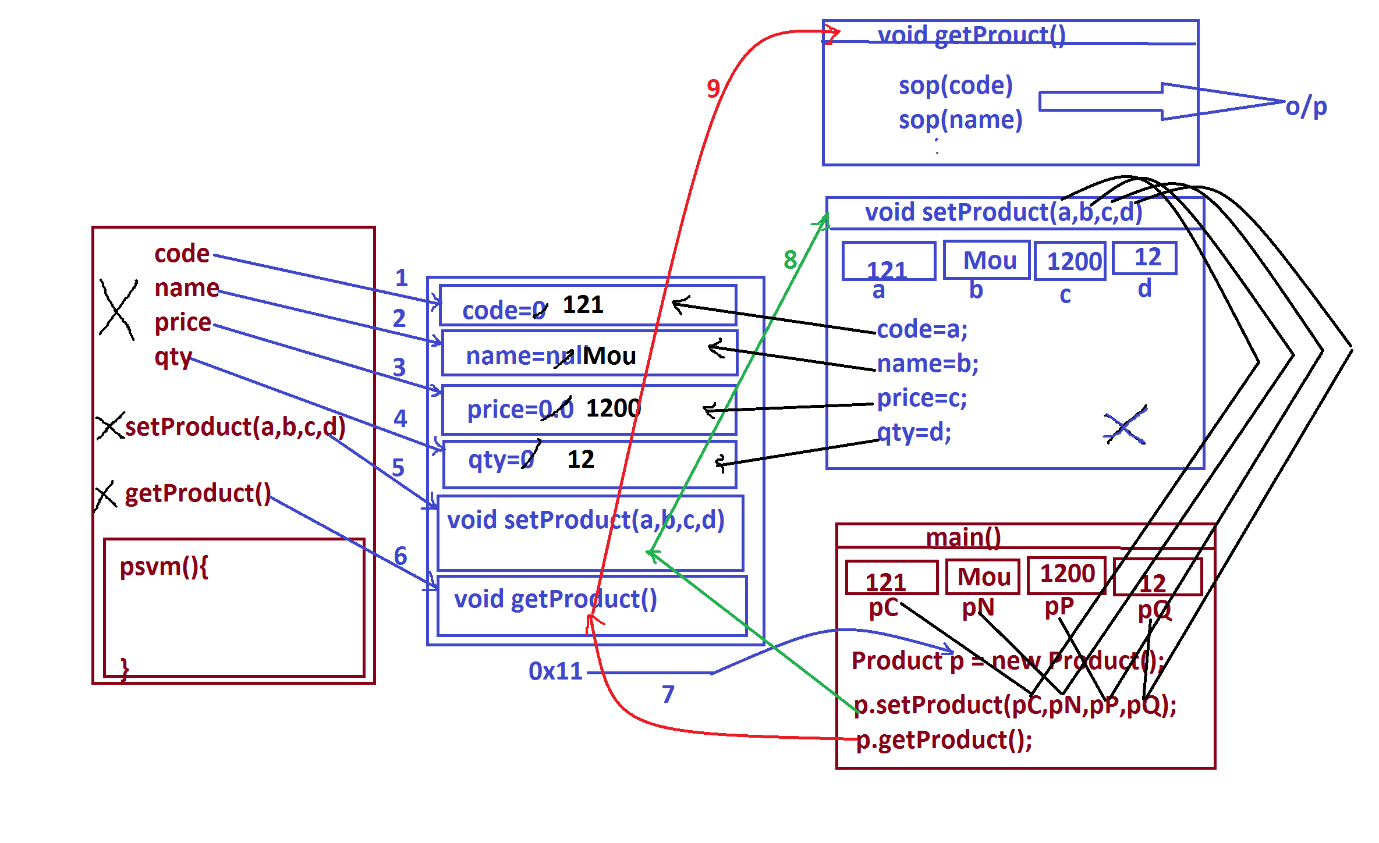
***wap to display book details?***

***Layout:***

******

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

***Diagram of Product.java***

******

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

***faq:***

***Types of parameters:***

***=>Parameters are categorized into two types:***

***(i)Actual Parameters***

***(ii)Formal Parameters***

***(i)Actual Parameters:***

***=>The parameters which hold original input data and used while***

***method calls are known as Actual Parameters.***

***Ex:***

***pC,pN,pP,pQ***

***(ii)Formal Parameters:***

***=>The parameters which hold intermediate data and used in***

***method signature are known as formal parameters.***

***Ex:***

***a,b,c,d***

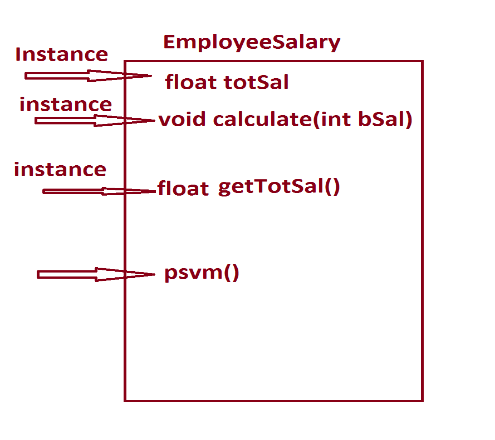
***Note:***

***=>Actual parameters and Formal parameters can be declared with***

***same names.***

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

***Ex\_Program:(Demonstrating return type method)***

******

***program : EmployeeSalary.java***

***class EmployeeSalary***

***{***

***float totSal;***

***void calculate(int bSal)***

***{***

***totSal=bSal+(0.93F\*bSal)+(0.63F\*bSal);***

***}***

***float getTotSal()***

***{***

***return totSal;***

***}***

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

***{***

***int bS = 12000;***

***EmployeeSalary es = new EmployeeSalary();***

***es.calculate(bS);***

***float tS = es.getTotSal();***

***System.out.println("TotSal of an Employee="+tS);***

***}***

***}***

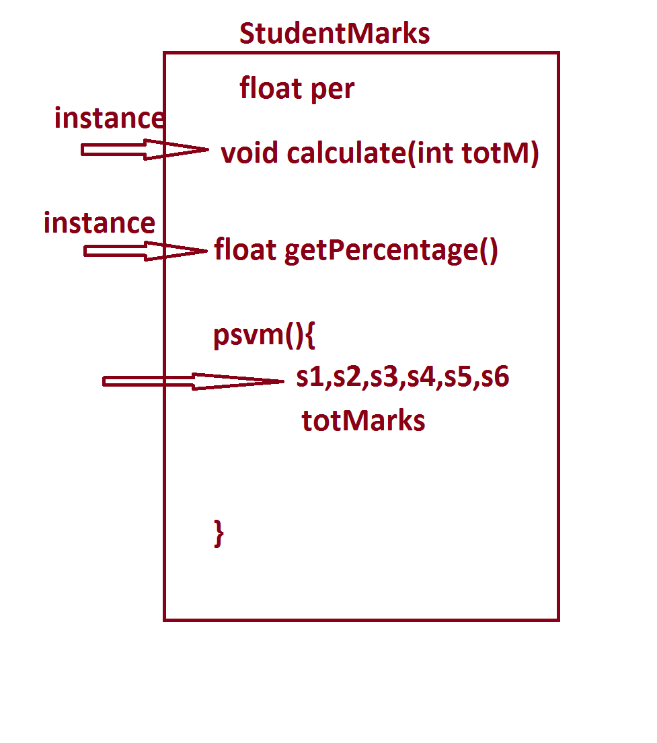
***o/p:***

***TotSal of an Employee=30720.0***

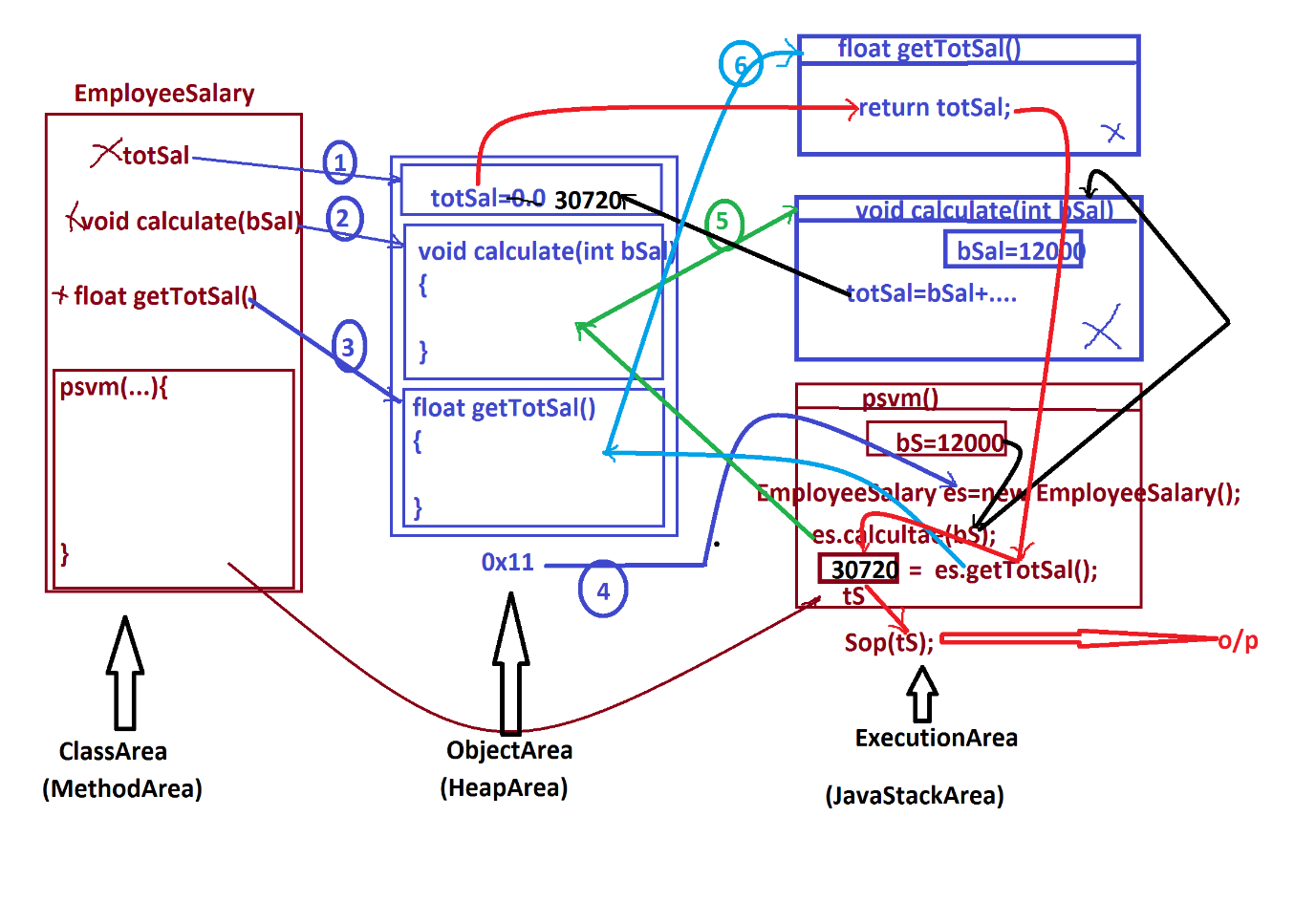
***==========================================================***

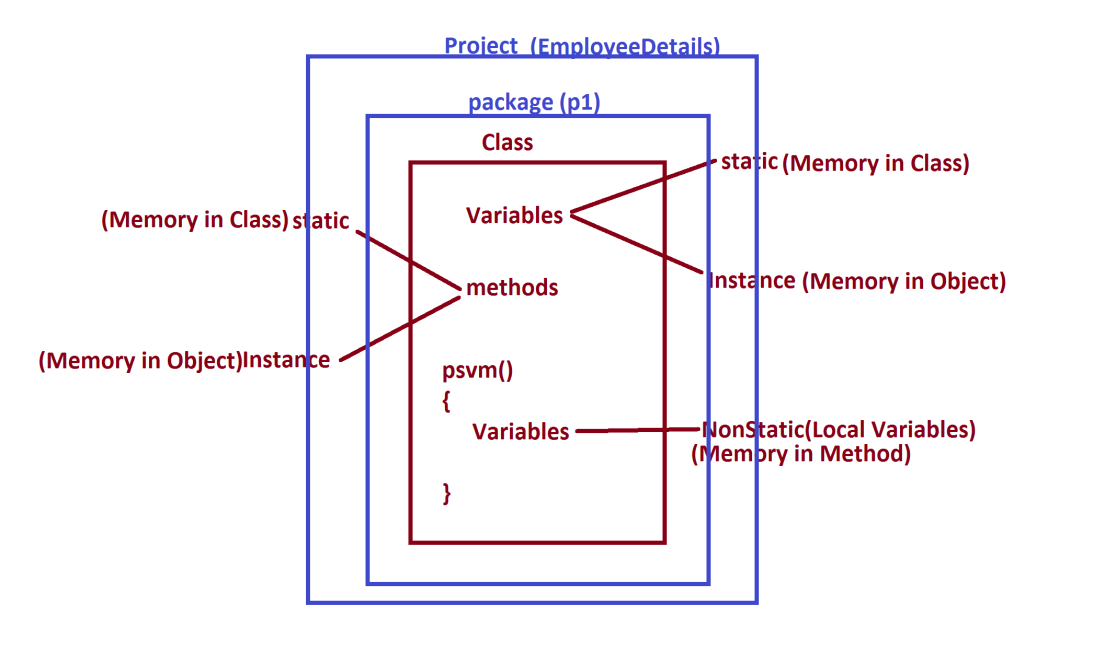
***Assignment:***

***Layout:***

******

***Execution flow of EmployeeSalary.java***

******

******

***\*imp***

***packages in Java:***

***=>package is a collection of Classes and Interfaces.***

***=>packages are categorized into two types:***

***1.User defined packages***

***2.Pre-Defined packages***

***1.User defined packages:***

***=>The packages which are defined by the programmer are known as User defined packages or Custom packages.***

***=>we use 'package' keyword to declare user defined packages.***

***syntax:***

***package package\_name;***

***2.Pre-Defined packages:***

***=>The packages which are available from JavaLib are known as Pre-defined packages or Built-in packages***

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

***CoreJava packages:***

***(i)java.lang - Language package(default package)***

***(ii)java.util - Utility package***

***(iii)java.io - Streams and Files package***

***(iv)java.net - Networking package***

***AdvJava packages:***

***(i)java.sql - DataBase connection package***

***(ii)javax.servlet - Servlet Programming package***

***(iii)javax.servlet.jsp - JSP programming package***

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

***\*imp***

***Creating JavaProject using IDE Eclipse:***

***(IDE - Integrated Development Environment)***

***step-1 : DownLoad IDE Eclipse from www.eclipse.org***

***https://www.eclipse.org/downloads/***

***Select : Eclipse IDE for Enterprise Java and Web Developers***

***step-2 : Open IDE Eclipse,while opening name the WorkSpace and click 'Launch'.***

***step-3 : Create Java Project***

***Click on File->new->Project->Java->Select 'Java project' and click 'next'->name the project and click 'finish'.***

***step-4 : Create package in 'src'.***

***RightClick on 'src'->new->package,name the package and click 'finish'***

***step-5 : create class under package***

***RightClick on package->new->class,name the class and click finish.***

***Note:***

***=>To increase Editor Text font,Click Window->Preferences-> General->Appearance->Colors and Fonts->Java-> Java Editor Text Font...***

***step-6 : Construct the Class code***

***step-7 : Open MainClass and execute***

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

***"Scanner" class:***

***=>"Scanner" is a Pre-Defined class from java.util package and***

***which provide pre-defined methods to read data into Java program.***

***=>The following are some important pre-defined methods from***

***'Scanner' class:***

***1.nextByte()***

***2.nextShort()***

***3.nextInt()***

***4.nextLong()***

***5.nextFloat()***

***6.nextDouble()***

***7.nextBoolean()***

***8.nextLine()***

***=>These are NonStatic methods and will get the memory within the object and can be accessed with object\_name.***

***=>The following is the syntax of creating object for 'Scanner'***

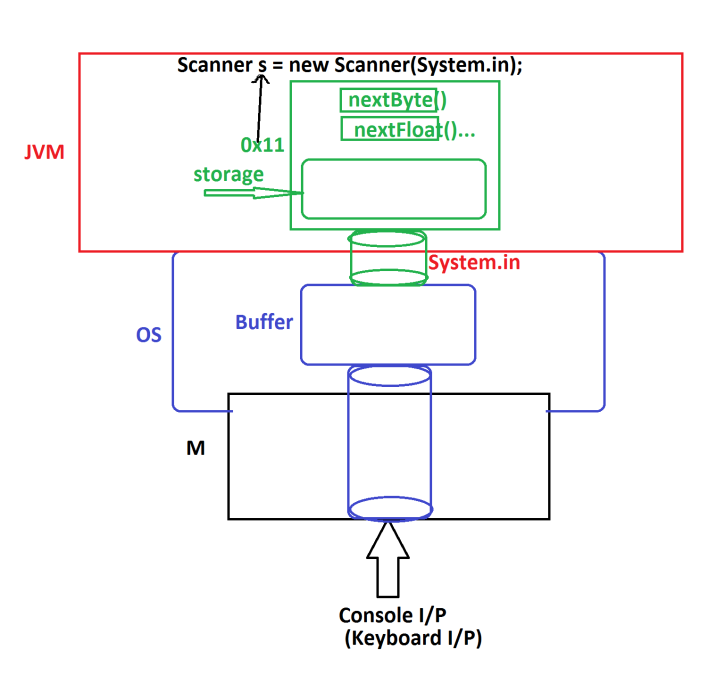
***class:***

***syntax:***

***Scanner s = new Scanner(System.in);***

***=>'System.in' in Java represent connecting OS-Buffer to Scanner***

***Class object.***

******

***1.nextByte():***

***=>This method is used to read byte value.***

***Method Signature:***

***public byte nextByte();***

***syntax:***

***byte b = s.nextByte();***

***2.nextShort():***

***=>This method is used to read short value.***

***Method Signature:***

***public short nextShort();***

***syntax:***

***short sh = s.nextShort();***

***3.nextInt():***

***=>This method is used to read int value.***

***Method Signature:***

***public int nextInt();***

***syntax:***

***int i = s.nextInt();***

***4.nextLong():***

***=>This method is used to read long value.***

***Method Signature:***

***public long nextLong();***

***syntax:***

***long l = s.nextLong();***

***5.nextFloat():***

***=>This method is used to read float value.***

***Method Signature:***

***public float nextFloat();***

***syntax:***

***float f = s.nextFloat();***

***6.nextDouble():***

***=>This method is used to read double value.***

***Method Signature:***

***public double nextDouble();***

***syntax:***

***double d = s.nextDouble();***

***7.nextBoolean():***

***=>This method is used to read boolean value.***

***Method Signature:***

***public boolean nextBoolean();***

***syntax:***

***boolean bl = s.nextBoolean();***

***8.nextLine():***

***=>This method is used to read string value.***

***Method Signature:***

***public String nextLine();***

***syntax:***

***String str = s.nextLine();***

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

***Ex\_Program:***

***wap to read and display User details like name,mailid and phno?***

***Program : DemoUser.java***

***package p1;***

***import java.util.Scanner;***

***public class DemoUser***

***{***

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

***{***

***Scanner s = new Scanner(System.in);***

***System.out.println("Enter the username:");***

***String name = s.nextLine();***

***System.out.println("Enter the MailId:");***

***String mId = s.nextLine();***

***System.out.println("Enter the phoneNo:");***

***long phNo = s.nextLong();***

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

***System.out.println("UserName:"+name);***

***System.out.println("MailId:"+mId);***

***System.out.println("PhoneNo:"+phNo);***

***s.close();***

***}***

***}***

***o/p:***

***Enter the username:***

***JAVA***

***Enter the MailId:***

***v@gmail.com***

***Enter the phoneNo:***

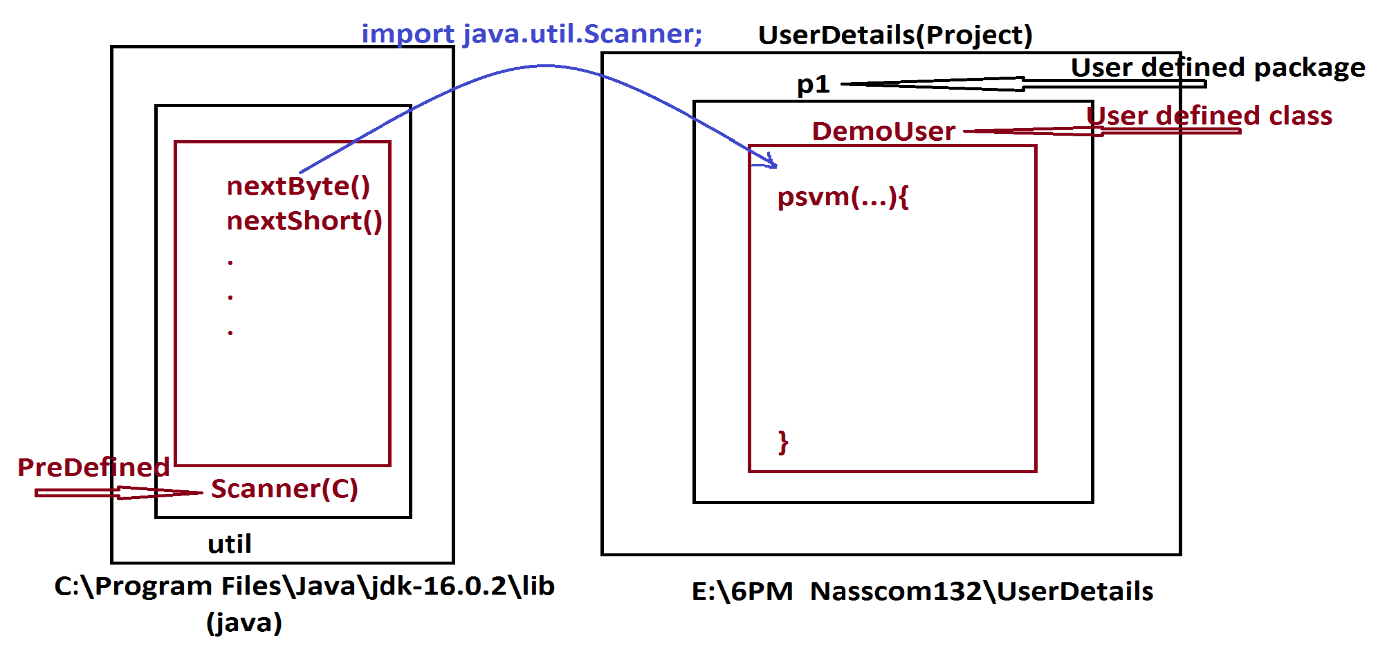
***9898981234***

***====UserDetails====***

***UserName: JAVA***

***MailId:v@gmail.com***

***PhoneNo:9898981234***

******

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

***Assignment:***

***Update StudentResult program by reading six subject marks from Console(Keyboard)?***

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

***Note:***

***=>In Java,when we read String data after numeric data then the data will be skipped in reading.***

***=>This can be overcomed using the following parse methods:***

***byte b = Byte.parseByte(s.nextLine());***

***short sh = Short.parseShort(s.nextLine());***

***int i = Integer.parseInt(s.nextLine());***

***long l = Long.parseLong(s.nextLine());***

***float f = Float.parseFloat(s.nextLine());***

***double d = Double.parseDouble(s.nextLine());***

***Program : Demo.java***

***package p1;***

***import java.util.Scanner;***

***public class Demo {***

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

***Scanner s = new Scanner(System.in);***

***System.out.println("Enter the name:");***

***String name = s.nextLine();***

***System.out.println("Enter the PhoneNO:");***

***long phNo = Long.parseLong(s.nextLine());***

***System.out.println("Enter the MailId:");***

***String mId = s.nextLine();***

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

***System.out.println("Name:"+name);***

***System.out.println("PhNo:"+phNo);***

***System.out.println("mId:"+mId);***

***s.close();***

***} }***

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

***Note:***

***=>In the process of constructing JavaApplications we must use SubClasses and MainClass.***

***=>The application can have any number of SubClasses and only one***

***MainClass.***

***SubClasses => The classes which are declared in the application other than MainClass are known as SubClasses***

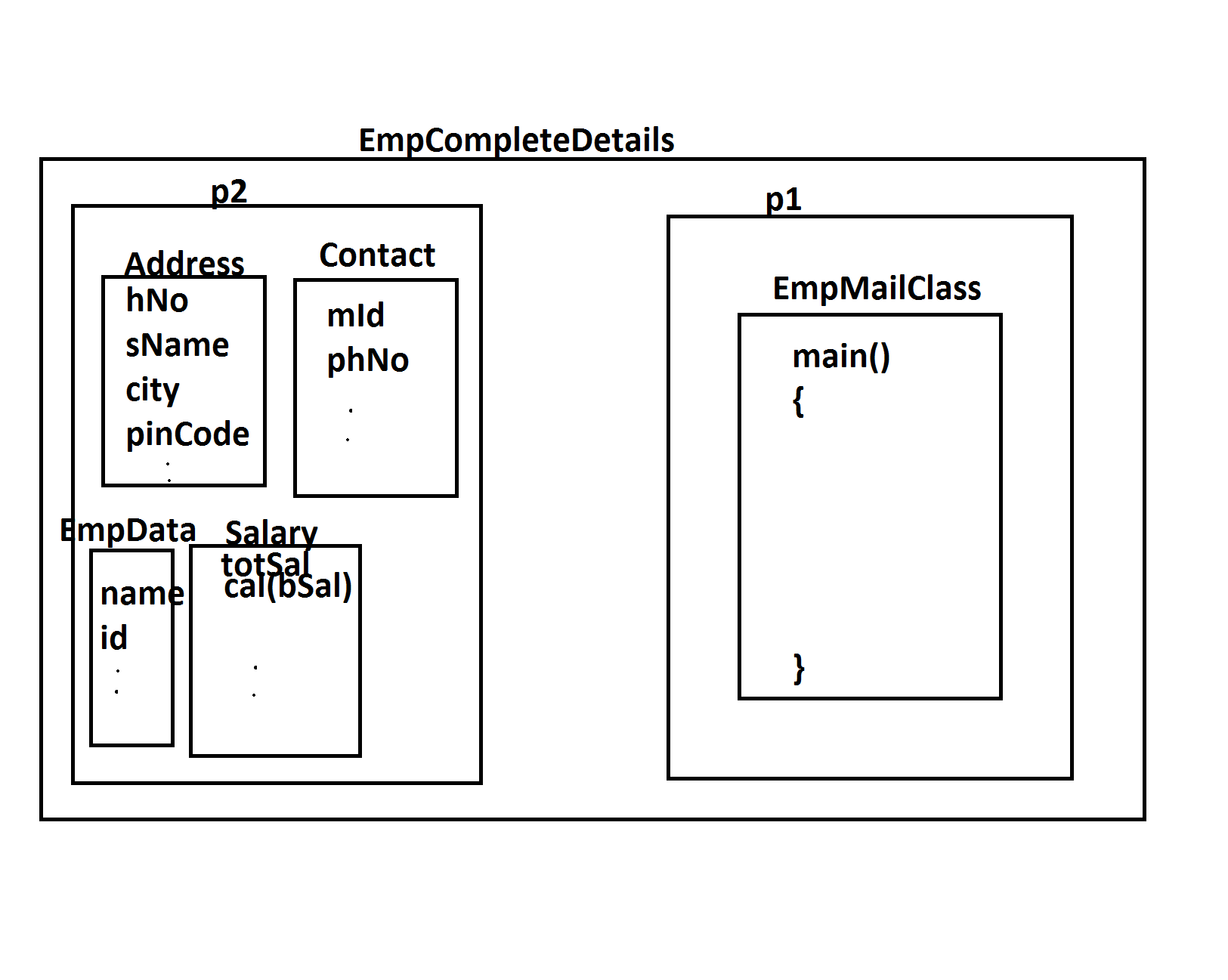
***MainClass => The Classes which is declared with main() is known as MainClass.***

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

***Ex:***

***ProjectName : EmpCompleteDetails***

***Diagram:***

******

***p1 : EmpMainClass***

***p2 : EmpData,Address,Contact,Salary***

***EmpMainClass.java***

***package p1;***

***import java.util.Scanner;***

***import p2.EmpData;***

***import p2.Address;***

***import p2.Contact;***

***import p2.Salary;***

***public class EmpMainClass {***

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

***Scanner s = new Scanner(System.in);***

***EmpData ed = new EmpData();***

***Address ad = new Address();***

***Contact cn = new Contact();***

***Salary sl = new Salary();***

***System.out.println("Enter the EmpId:");***

***String eId = s.nextLine();***

***System.out.println("Enter the EmpName:");***

***String eName = s.nextLine();***

***ed.setEmpData(eId, eName);//Setting the data to object***

***System.out.println("Enter the hNo:");***

***String hNo = s.nextLine();***

***System.out.println("Enter the SName:");***

***String sName = s.nextLine();***

***System.out.println("Enter the City:");***

***String city = s.nextLine();***

***System.out.println("Enter the PinNo:");***

***int pinNo = Integer.parseInt(s.nextLine());***

***ad.setAddress(hNo, sName, city, pinNo);***

***//Setting the data to Object***

***System.out.println("Enter the MailId:");***

***String mId = s.nextLine();***

***System.out.println("Enter the PhNo:");***

***long phNo = s.nextLong();***

***cn.setContact(mId, phNo);//Setting the data to Object***

***System.out.println("Enter the bSal:");***

***int bSal = s.nextInt();***

***sl.calculate(bSal);***

***float tS = sl.getTotSal();***

***ed.getEmpData();***

***ad.getAddress();***

***cn.getContact();***

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

***System.out.println("BSal:"+bSal);***

***System.out.println("TotSal:"+tS);***

***s.close();***

***}***

***}***

***EmpData.java***

***package p2;***

***public class EmpData {***

***public String id,name;***

***public void setEmpData(String a,String b)***

***{***

***id=a;***

***name=b;***

***}***

***public void getEmpData()***

***{***

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

***System.out.println("Id:"+id);***

***System.out.println("name:"+name);***

***}***

***}***

***Address.java***

***package p2;***

***public class Address {***

***public String hNo,sName,city;***

***public int pinCode;***

***public void setAddress(String a,String b,String c,int d)***

***{***

***hNo=a;***

***sName=b;***

***city=c;***

***pinCode=d;***

***}***

***public void getAddress()***

***{***

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

***System.out.println("HNO:"+hNo);***

***System.out.println("SName:"+sName);***

***System.out.println("City:"+city);***

***System.out.println("PinCode:"+pinCode);***

***}***

***}***

***Contact.java***

***package p2;***

***public class Contact {***

***public String mId;***

***public long phNo;***

***public void setContact(String a,long b)***

***{***

***mId=a;***

***phNo=b;***

***}***

***public void getContact()***

***{***

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

***System.out.println("MId:"+mId);***

***System.out.println("PhNo:"+phNo);***

***}***

***}***

***Salary.java***

***package p2;***

***public class Salary {***

***public float totSal;***

***public void calculate(int bSal)***

***{***

***totSal = bSal+(0.93F\*bSal)+(0.63F\*bSal);***

***}***

***public float getTotSal()***

***{***

***return totSal;***

***}***

***}***

***o/p:***

***Enter the EmpId:***

***A121***

***Enter the EmpName:***

***Raj***

***Enter the hNo:***

***12-34/h***

***Enter the SName:***

***MainRaod***

***Enter the City:***

***Hyd***

***Enter the PinNo:***

***612345***

***Enter the MailId:***

***raj@gmail.com***

***Enter the PhNo:***

***9898981234***

***Enter the bSal:***

***120000***

***====EmpData====***

***Id:A121***

***name:Raj***

***====Address=====***

***HNO:12-34/h***

***SName: MainRaod***

***City:Hyd***

***PinCode:612345***

***====Contact====***

***MId:raj@gmail.com***

***PhNo:9898981234***

***====Salary====***

***BSal:120000***

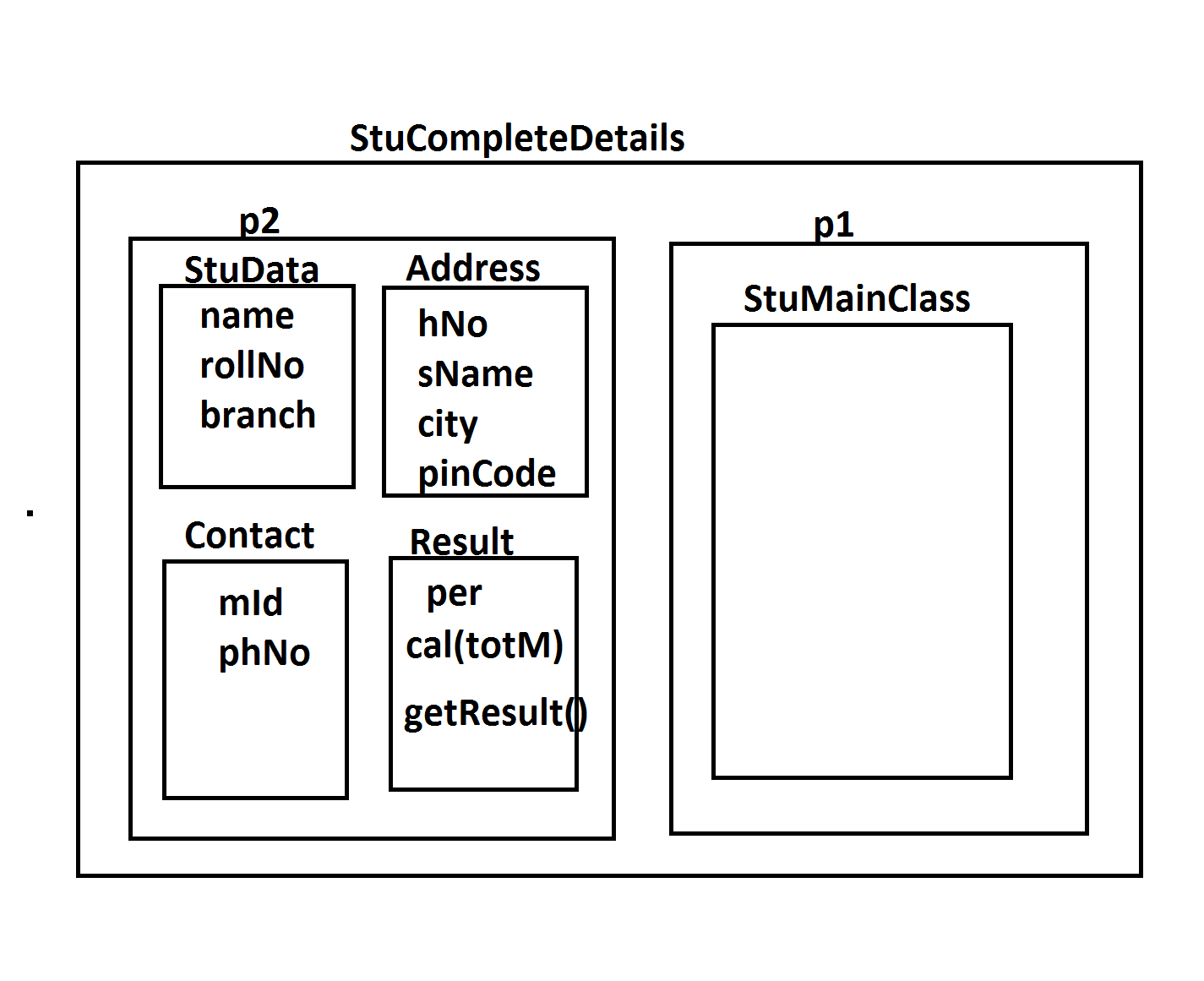
***TotSal:307200.0***

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

***Assinment:***

***Construct Application for StudentCompleteDetails.***

***Layout:***

******

***\*imp***

***Operators in Java:***

***=>Operator is a Special symbol used to perform operations.***

***=>The following are some important operators in Java:***

***1.Arithmetic Operators***

***2.Relational Operators***

***3.Logical Operators***

***4.Increment-Decrement Operators***

***1.Arithmetic Operators:***

***=>Arithmetic Operators are used to perform basic operations***

***or Fundamental operations.***

***Operator Meaning***

***+ Addition***

***- Subtraction***

***\* Multiplication***

***/ Division***

***% ModDivision***

***Ex:***

***a=7 b=2***

***a/b = 7/2 = 3***

***a%b = 7%2 = 1***

***2.Relational Operators:***

***=>Relational Operators are used to compare two values and***

***generate boolean result.***

***=>These operators are also known as Conditional Operators***

***Operator Meaning***

***> Greater Than***

***>= Greater Than or Equal to***

***< Less Than***

***<= Less Than or Equal to***

***== Is equal to***

***!= Not equal to***

***3.Logical Operators:***

***=>Logical Operators are used to compare two comparisions and***

***generate boolean result.***

***Operator Meaning***

***&& Logical AND***

***|| Logical OR***

***! Logical NOT***

***Logical AND(&&):***

***A B A&&B***

***T T T***

***F T F***

***T F F***

***F F F***

***Logical OR(||):***

***A B A||B***

***T T T***

***F T T***

***T F T***

***F F F***

***Logical NOT(!):***

***A !A***

***T F***

***F T***

***4.Increment-Decrement Operators:***

***=>Increment operator will increment the value by 1 and decrement***

***operator will decrement the value by 1.***

***Operator Meaning***

***++ Increment***

***-- Decrement***

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

***\*imp***

***Control Structures in Java:***

***=>The structures which are used to control the part of the***

***program are known as Control Structures.***

***=>These Control Structures are categorized into three types:***

***1.Selection Statements***

***2.Iterative Statements***

***3.Branching Statements***

***1.Selection Statements:***

***=>The Statements which are used to select one part of the***

***program for execution are known as Selection Statements or***

***Conditional Statements.***

***=>Types:***

***(a)Simple if***

***(b)if-else***

***(c)Nested if***

***(d)Ladder if-else***

***(e)switch-case***

***2.Iterative Statements:***

***=>The statements which are used to execute set-of-lines from the***

***program repeatedly on some condition are known as Iterative***

***Statements or Repeatitive Statements or Looping Structures.***

***=>Types:***

***(a)while loop***

***(b)do-while loop***

***(c)for loop***

***3.Branching Statements:***

***=>The statements which are used to transfer the control from***

***one location to another location are known as Branching Statements***

***or Transfer Statements.***

***=>Types:***

***(a)break***

***(b)continue***

***(c)return***

***(d)exit***

***Note:***

***=>'goto' statement is not available in Java.***

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

***Ex\_Program:***

***wap to read three integer values and perform comparision based***

***on User choice.***

***1.Greater***

***2.Smaller***

***Layout:***

***Greater.java***

***package p2;***

***public class Greater {***

***public int compare(int x,int y,int z)***

***{***

***if(x>y && x>z)***

***{***

***return x;***

***}***

***else if(y>x && y>z)***

***{***

***return y;***

***}***

***else***

***{***

***return z;***

***}***

***}***

***}***

***Smaller.java***

***package p2;***

***public class Smaller {***

***public int compare(int x,int y,int z)***

***{***

***if(x<y && x<z)***

***{***

***return x;***

***}***

***else if(y<x && y<z)***

***{***

***return y;***

***}***

***else***

***{***

***return z;***

***}***

***}***

***}***

***DemoComparision.java(MainClass)***

***package p1;***

***import java.util.Scanner;***

***import p2.Greater;***

***import p2.Smaller;***

***public class DemeComparision {***

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

***Scanner s = new Scanner(System.in);***

***System.out.println("Enter the Value1:");***

***int v1 = s.nextInt();***

***System.out.println("Enter the Value2:");***

***int v2 = s.nextInt();***

***System.out.println("Enter the Value3:");***

***int v3 = s.nextInt();***

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

***System.out.println("1.Greater\n2.Smaller");***

***System.out.println("Enter the Choice:");***

***int choice = s.nextInt();***

***switch(choice)***

***{***

***case 1:***

***Greater gt = new Greater();***

***int r1 = gt.compare(v1,v2,v3);***

***System.out.println("Greater Value:"+r1);***

***break;***

***case 2:***

***Smaller sm = new Smaller();***

***int r2 = sm.compare(v1,v2,v3);***

***System.out.println("Smaller Value:"+r2);***

***break;***

***default:***

***System.out.println("Invalid Choice...");***

***}//end of switch***

***s.close();***

***}***

***}***

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

***faq:***

***define switch-case statement?***

***=>swich-case statement is used to select one from multiple***

***available options or cases.***

***syntax:***

***switch(value)***

***{***

***case 1 : statements;***

***break;***

***case 2 : statements;***

***break;***

***.***

***.***

***case n : statements;***

***break;***

***default : statements;***

***}***

***behaviour:***

***=>The switch-value is compared with available options,if the***

***value is matched with any option then the statements under the***

***option are executed.***

***=>After executing the statements the switch-case execution is***

***stopped using 'break'.***

***=>If the switch-value is not matched with any available options***

***then default is executed.***

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

***Assignment:***

***wap to read two int values and perform arithmetic operation based***

***on User choice:***

***1.add***

***2.sub***

***3.mul***

***4.div***

***5.modDiv***

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

***Assignment:(Solution)***

***wap to read two int values and perform arithmetic operation based***

***on User choice:***

***1.add***

***2.sub***

***3.mul***

***4.div***

***5.modDiv***

***Addition.java***

***package p2;***

***public class Addition {***

***public int add(int x,int y)***

***{***

***return x+y;***

***}***

***}***

***Subtraction.java***

***package p2;***

***public class Subtraction {***

***public int sub(int x,int y)***

***{***

***return x-y;***

***}***

***}***

***Multiplication.java***

***package p2;***

***public class Multiplication {***

***public int mul(int x,int y)***

***{***

***return x\*y;***

***}***

***}***

***Division.java***

***package p2;***

***public class Division {***

***public float div(int x,int y)***

***{***

***return (float)x/y;***

***}***

***}***

***ModDivision.java***

***package p2;***

***public class ModDivision {***

***public int modDiv(int x,int y)***

***{***

***return x%y;***

***}***

***}***

***DemoArithmetic.java(MainClass)***

***package p1;***

***import java.util.Scanner;***

***import p2.Addition;***

***import p2.Subtraction;***

***import p2.Multiplication;***

***import p2.Division;***

***import p2.ModDivision;***

***public class DemoArithmetic {***

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

***Scanner s = new Scanner(System.in);***

***System.out.println("Enter int value1:");***

***int v1 = s.nextInt();***

***System.out.println("Enter int value2:");***

***int v2 = s.nextInt();***

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

***System.out.println("1.add\n2.sub\n3.mul\n4.div\n5.modDiv");***

***System.out.println("Enter the Choice:");***

***int choice = s.nextInt();***

***switch(choice)***

***{***

***case 1:***

***Addition ad = new Addition();***

***int r1 = ad.add(v1,v2);***

***System.out.println("Sum:"+r1);***

***break;***

***case 2:***

***Subtraction sb = new Subtraction();***

***int r2 = sb.sub(v1,v2);***

***System.out.println("Sub:"+r2);***

***break;***

***case 3:***

***Multiplication ml = new Multiplication();***

***int r3=ml.mul(v1,v2);***

***System.out.println("Mul:"+r3);***

***break;***

***case 4:***

***Division dv = new Division();***

***float r4=dv.div(v1,v2);***

***System.out.println("Div:"+r4);***

***break;***

***case 5:***

***ModDivision md = new ModDivision();***

***int r5=md.modDiv(v1,v2);***

***System.out.println("ModDiv:"+r5);***

***break;***

***default:***

***System.out.println("Invalid Choice...");***

***}//end of switch***

***s.close();***

***}***

***}***

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

***Assignment:***

***wap to perform bank transaction process?***

***a.read pinNo***

***=>The pinNo must be in 1111 or 2222 or 3333,else Invalid pinNo***

***b.If the pinNo is verified,then show the following choice:***

***1.WithDraw***

***2.Deposit***

***1.WithDraw:***

***=>Enter the amt***

***=>The amt must be greater than zero and multiples of 100, else Invalid Amt***

***=>If the amt is validated, then create object for 'WithDraw' class and pass amt as parameter to the method of WithDraw***

***2.Deposit :***

***=>Enter the amt***

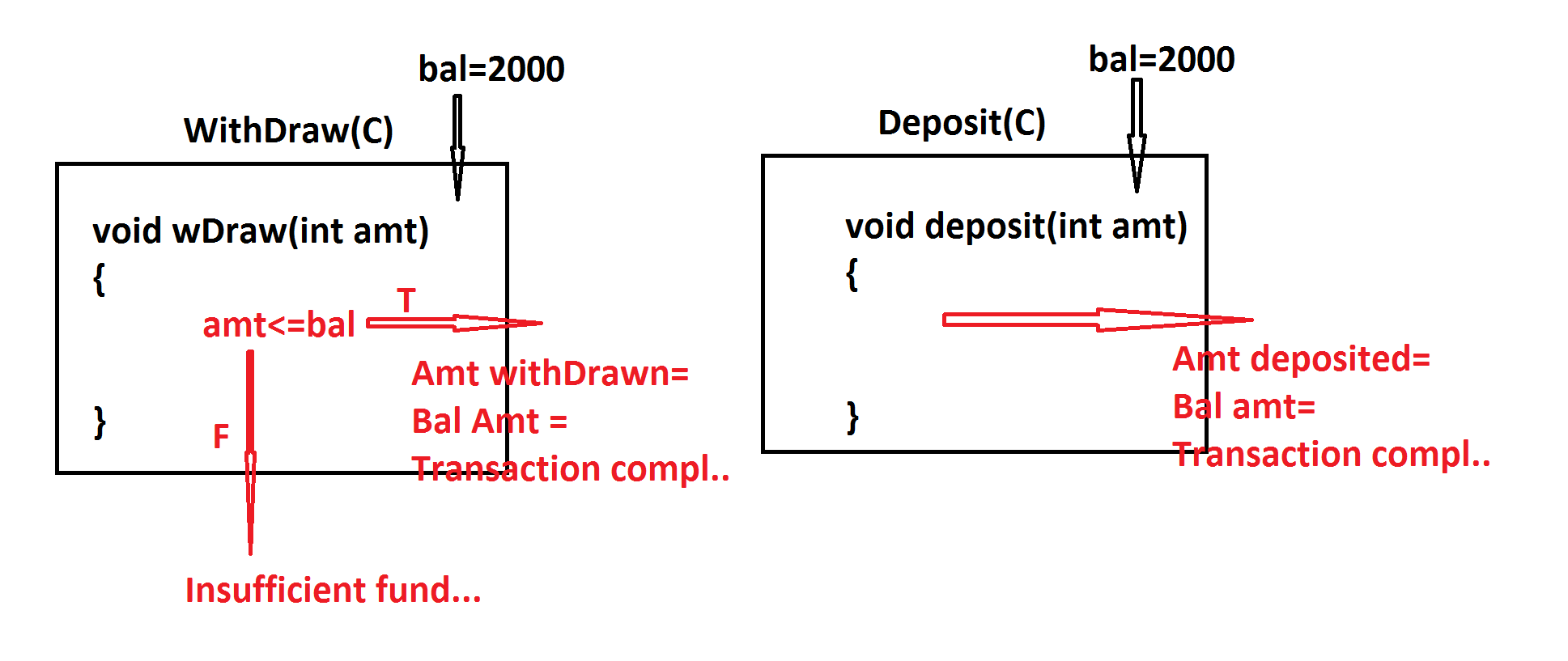
***=>The amt must be greater than zero and multiples of 100,***

***else Invalid Amt***

***=>If the amt is validated, then create object for 'Deposit'***

***class and pass amt as parameter to the method of Deposit***

***Layout:***

******

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

***\*imp***

***define while loop?***

***=>In while looping structure the condition is checked first,if***

***the condition is true then the loop\_body is executed and this***

***process is repeated until the condition is false.***

***syntax:***

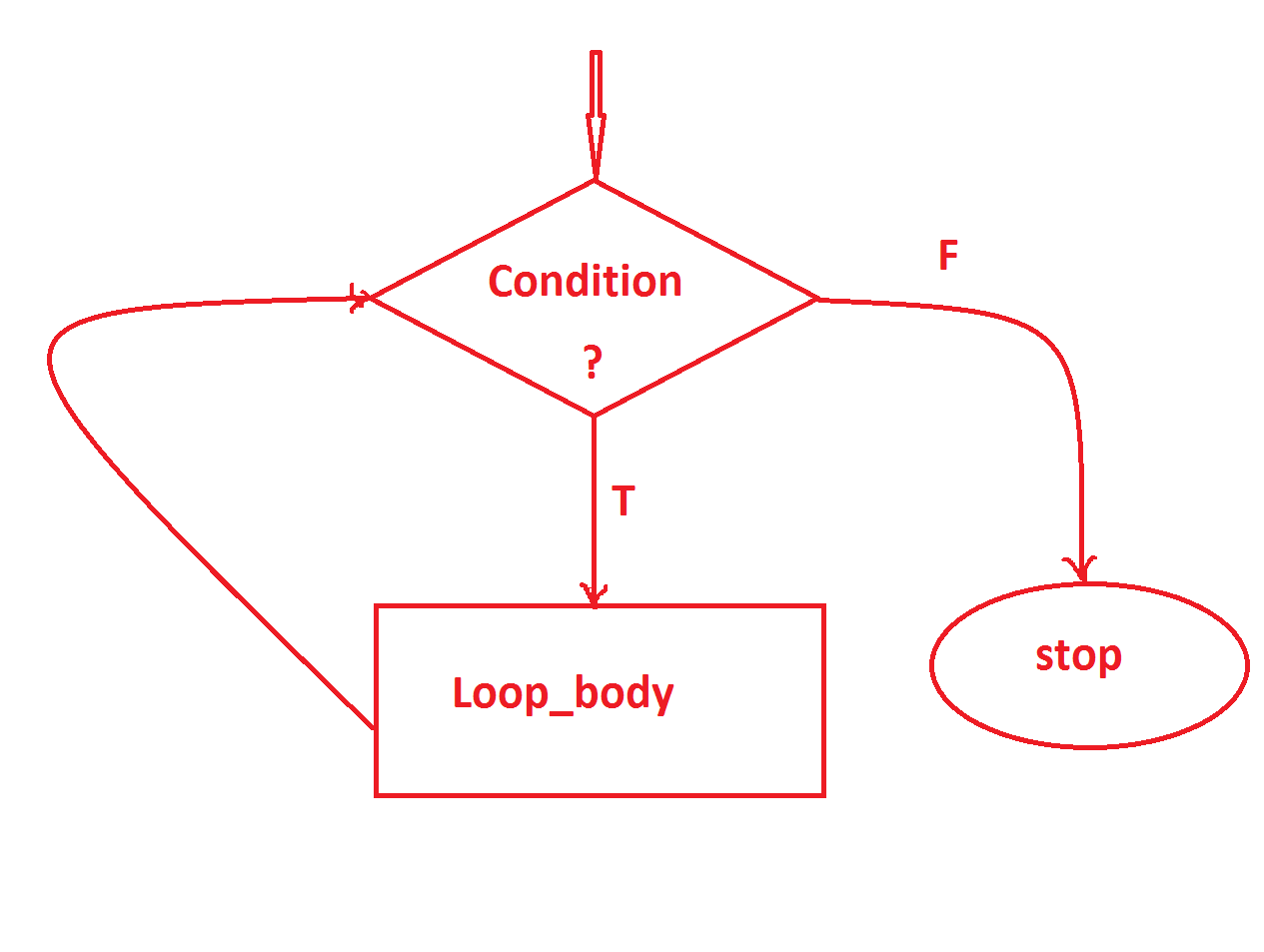
***while(condition)***

***{***

***//loop\_body***

***}***

***FlowChart:***

******

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

***\*imp***

***define do-while loop?***

***=>In do-while loop the loop\_body is executed first and then the condition is checked,this process is repeated until the condition is false.***

***syntax:***

***do***

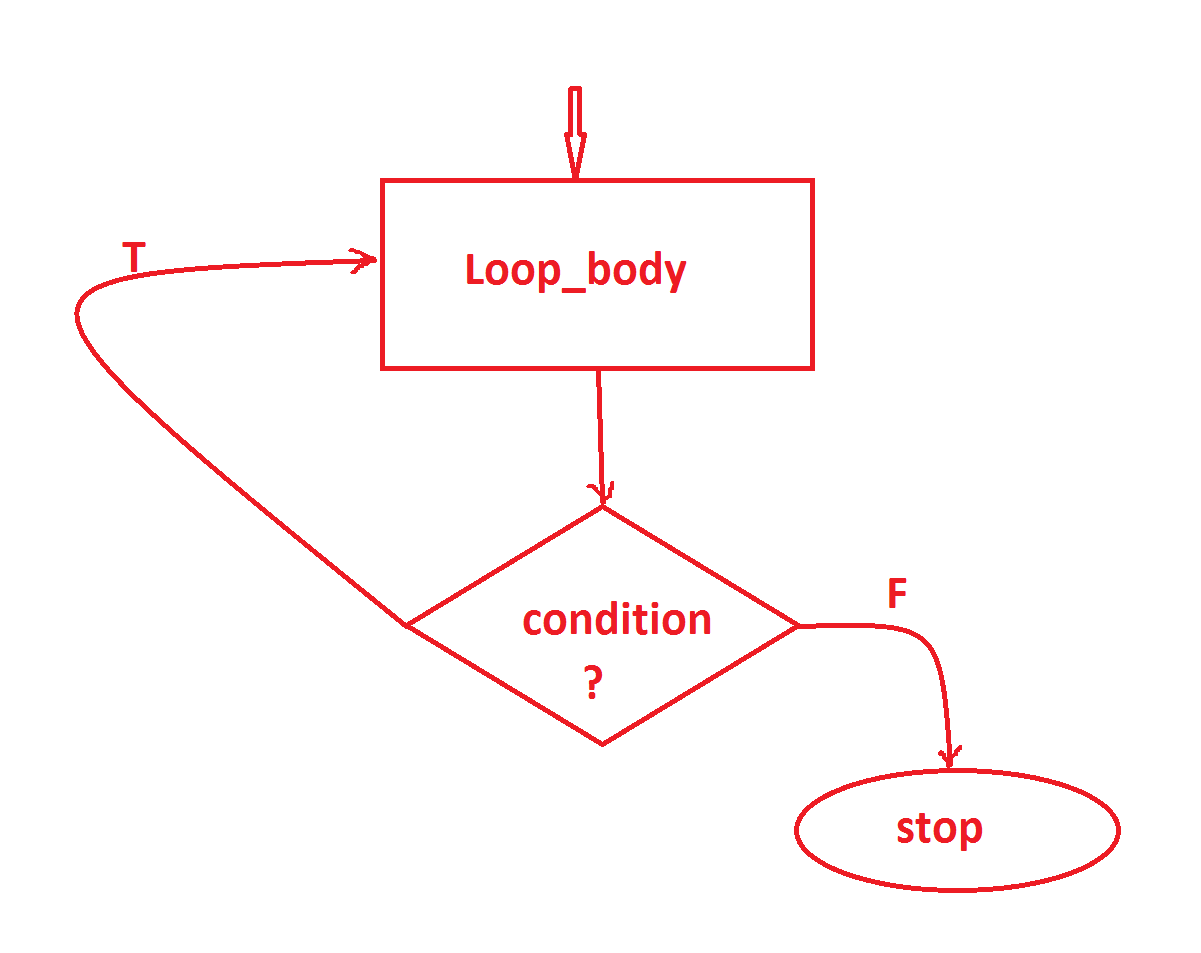
***{***

***//Loop\_body***

***}***

***while(condition);***

***FlowChart:***

******

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

***\*imp***

***define 'import' statement?***

***=>'import' statement is used to make class or interface available one package to another package.***

***=>This importing process in Java can be done in three ways:***

***(i)Using 'import package\_name.CName/IName; '***

***(ii)Using 'import package\_name.\*; '***

***(iii)using 'Fully Qualified Names'***

***(i)Using 'import package\_name.CName/IName; '***

***=>In this importing process we specify the required Class\_name or Interface\_name to be available for current running program.***

***Ex:***

***import java.util.Scanner;***

***import p2.StudentResult;***

***(ii)Using 'import package\_name.\*; ':***

***=>In this importing process all the Classes and Interfaces from the package are available to Current running program.***

***Ex:***

***import java.util.\*;***

***import p2.\*;***

***(iii)using 'Fully Qualified Names':***

***=>The process of declaring Classes and Interfaces with package names part of programming code is known as 'Fully Qualified Names'***

***Ex:***

***java.util.Scanner s = new java.util.Scanner(System.in);***

***p2.StudentResult sr = new p2.StudentResult();***

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

***faq:***

***define 'static' import?***

***=>The process of declaring import statement with 'static' keyword is known as static import and which is introduced by Java5 version.***

***syntax:***

***import static package\_name.CName/IName.\*;***

***Advantage:***

***=>when we use 'static import' then all the static members of Class or Interface available to current running program and can be accessed directly without Class\_name or Interface\_name.***

***define sqrt() method?***

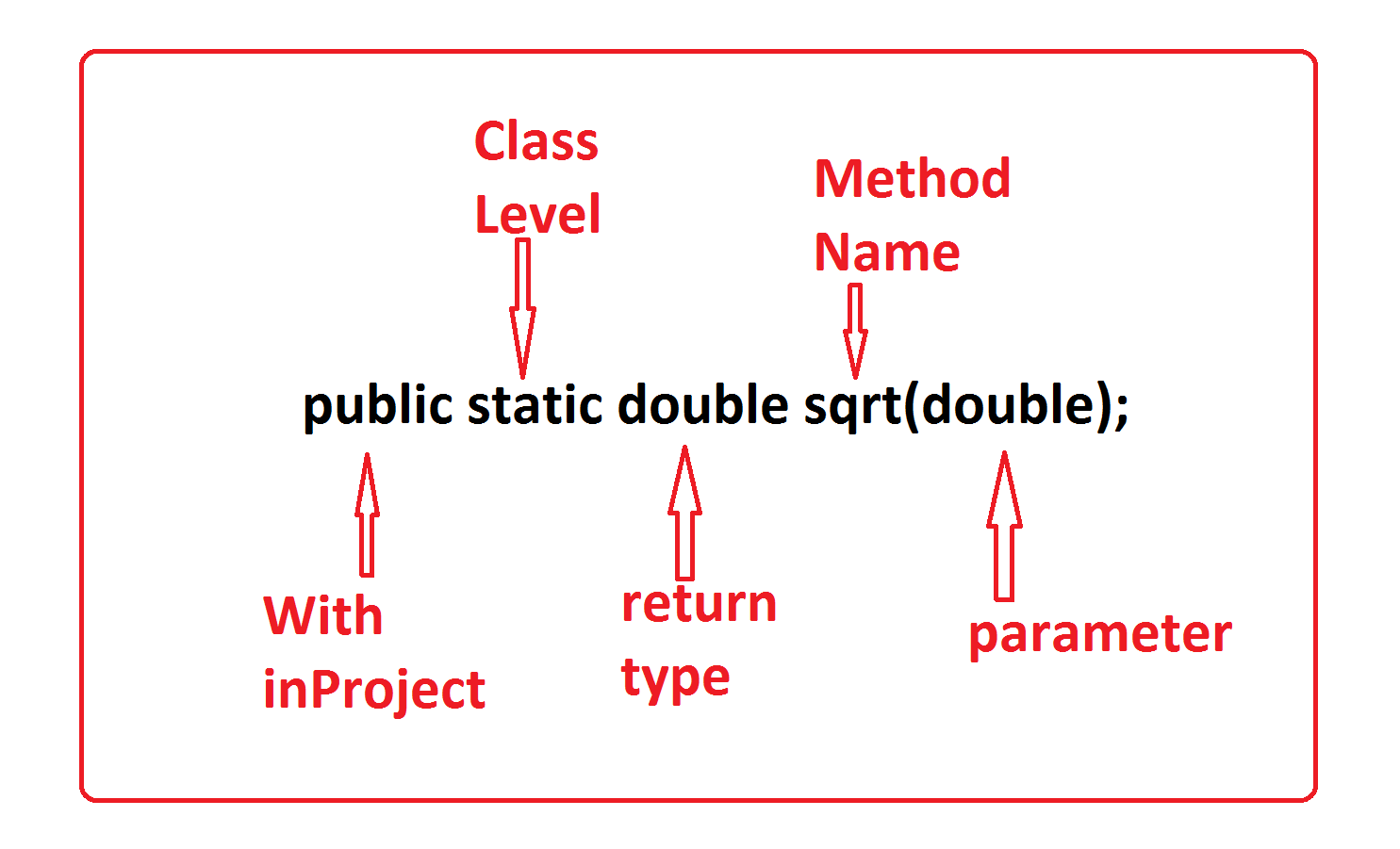
***=>sqrt() is a Pre-defined static method available from java.lang.Math class and which is used to find the sqrt of given number.***

***Method Signature:***

***public static double sqrt(double);***

***syntax:***

***double result = Math.sqrt(var);***

******

***Program : DemoStatic.java***

***package p1;***

***import java.util.Scanner;***

***import static java.lang.Math.\*;***

***public class DemoStatic {***

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

***Scanner s = new Scanner(System.in);***

***System.out.println("Enter the value:");***

***double val = s.nextDouble();***

***double result = sqrt(val);//Calculating\_sqrt***

***System.out.println("Result:"+result);***

***s.close();***

***}***

***}***

***o/p:***

***Enter the value:***

***234***

***Result:15.297058540778355***

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

***faq:***

***define Access Modifiers in Java?***

***=>Access Modifiers specify the visibility of programming***

***Components within the Project.***

***=>The following are some important access modifiers from Java:***

***1.public***

***2.private***

***3.protected***

***4.default***

***1.public:***

***=>public programming components are accessed within the project folder.***

***2.private:***

***=>private programming components are accessed only inside the class.***

***3.protected:***

***=>protected programming components are accessed within the package.***

***Note:***

***=>In inheritance process, protected programming components of ParentClass are available to ChildClass declared outside the package.***

***4.default:***

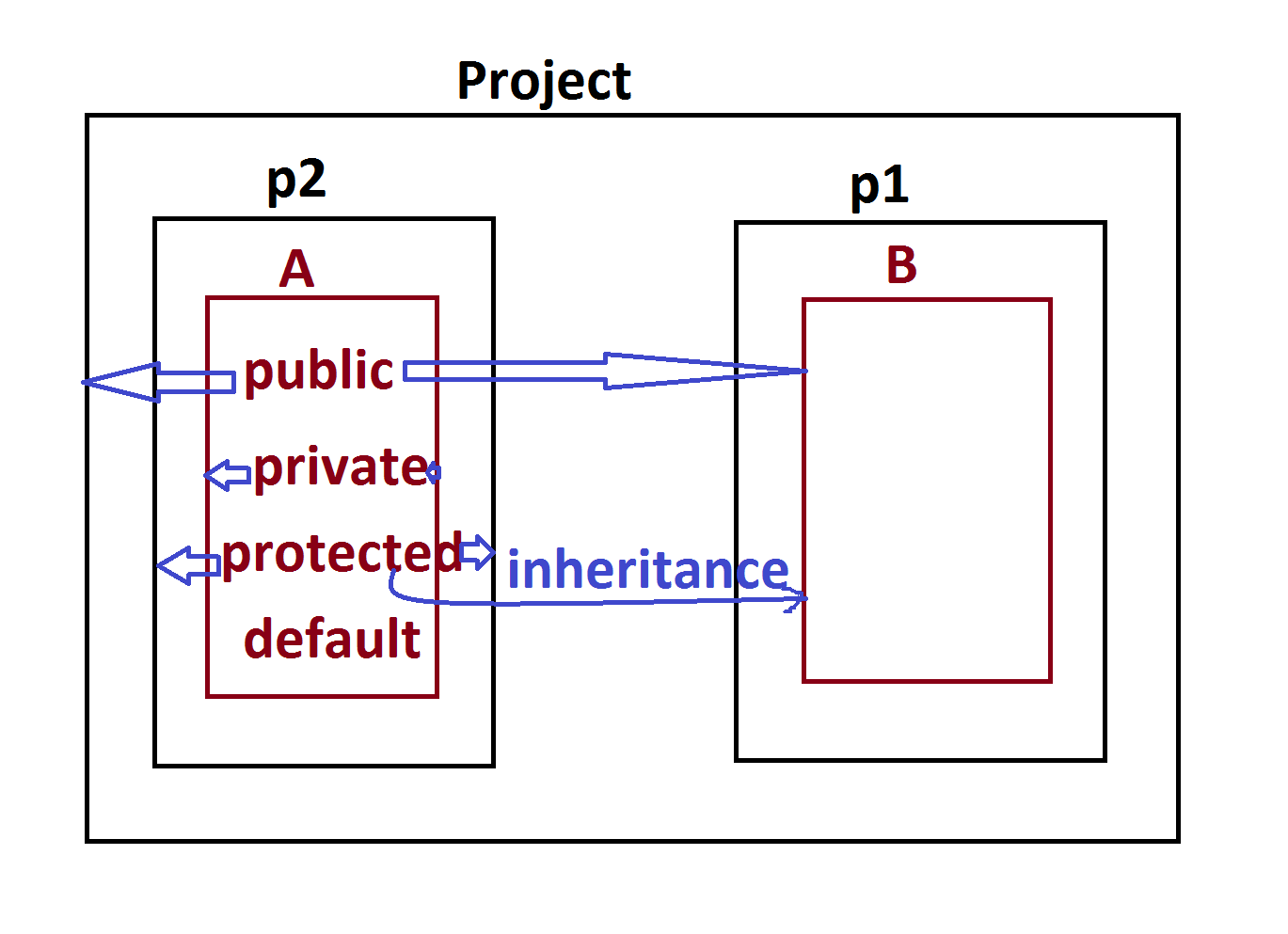
***=>default programming components are accessed only inside the package.***

***Note:***

***=>The programming components in Class which are declared without any access modifier then it is considered as default.***

***=>There is no concept of 'default' keyword in classes.***

***Diagram:***

******

***JVM Architecture:(JVM Internals)***

***=>JVM Stands for Java Virtual Machine and which is used to execute Java Byte Code.***

***=>JVM internally divided into the following partitions:***

***1.Class Loader SubSystem***

***2.Runtime Data Area***

***3.Execution Engine***

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

***=>This Class Loader SubSystem will load the ByteCode(Class file) onto Runtime data area using Loader.***

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

***=>Runtime Data Area,where execution is performed and this Runtime Data Area is 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 memory block where the classes are loaded is known as Method Area or Class Area.***

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

***=>In this process,MainClass is loaded onto Method Area first and main() method copied onto JavaStackArea to start the execution process.***

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

***=>The memory block where the object are created is known as***

***HeapArea or Object Area.***

***=>while Object creation process Instance members will get the memory within the Object.***

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

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

***JavaStackArea or ExecutionArea (which is like a RAM for JVM)***

***=>when Method is copied on to Java Stack Area,then separate***

***partition is created to hold method,known as Method Frame.***

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

***destroyed automatically.***

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

***this main() method will call remaining methods for execution.***

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

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

***execution in JavaStackArea.***

***=>Every Method which is copied onto Java Stack Area for***

***execution will be opened with its own PC-Register.***

***=>All these multiple PC-Registers are opened in a Separate***

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

***(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.***

***=>These Native methods are indentified by the***

***ClassLoaderSubSystem and loaded onto separate memory block known***

***as Native method Area.***

***=>ExecutionEngine will take the support of JNI(Java Native method***

***Interface) to execute Native methods.***

***=>while execution JNI will take the support of Native Method***

***Libraries.***

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

***faq:***

***why Native methods are available in javaLib?***

***=>using Native methods available in JavaLib,we can interact***

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

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

***3.Execution Engine:***

***=>Execution Engine is the executor of JVM and which starts the***

***execution process with main() available in JavaStackArea.***

***=>This execution engine internally uses the following two***

***translators:***

***(i)Interpreter***

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

***(i)Interpreter:***

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

***Normal Instructions.(Non-Stream instructions)***

***=>When Interpreter finds Stream Instructions then transfers the***

***control to JIT-Compiler.***

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

***=>JIT-Compiler will execute Stream Instructions or Bulk data***

***Instructions,which means Multimedia Instructions like Audio,Video,***

***Image and animation.***

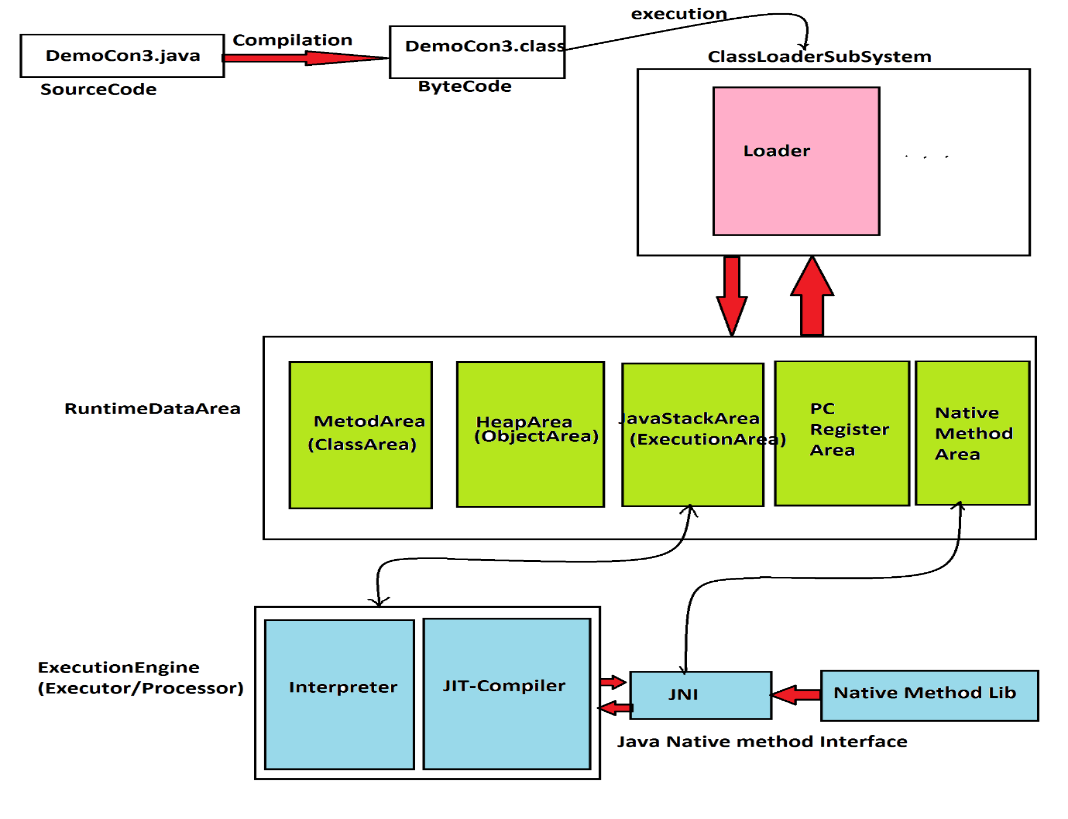
***faq:***

***why Java uses 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.***

******