JAVA

Classes and Methods

Lab Exercise No:14

Exercise Objective(s):*Construction of an object*

Exercise:*Create a class called Calculator which has 4 different methods add, diff, mul and div which*

*accepts two numbers as parameters. Create an object to access these methods and invoke*

*these methods with two numbers and display the result in the corresponding methods.*

Recommended duration:*20Mins*

Solution Guidance (if applicable): *NA*

*package com.hsbc.pack;*

*/\**

*\* A program to perform basic four operations of calculator!!*

*\* \*/*

*public class Calculator {*

*public void add(int num1,int num2){*

*System.out.println("Addition: "+(num1+num2));*

*}*

*public void sub(int num1,int num2){*

*System.out.println("Subtraction: "+(num1-num2));*

*}*

*public void mul(int num1,int num2){*

*System.out.println("Multiplication: "+(num1\*num2));*

*}*

*public void div(int num1,int num2){*

*System.out.println("Division: "+(num1/num2));*

*}*

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

*// TODO Auto-generated method stub*

*// Creating object of calculator*

*Calculator cal = new Calculator();*

*cal.add(10,2);*

*cal.sub(10,2);*

*cal.mul(10,2);*

*cal.div(10,2);*

*}*

*}*

Lab Exercise No:15

Exercise Objective(s):*Construction of an object, constructors*

Exercise:*Create a class called Sample. Write a program to display the no of objects created for that*

*class or the no of times that class is instantiated.*

Recommended duration:*15Mins*

Solution Guidance (if applicable): *Use a static int variable and increment that variable inside the*

*constructor.*

*package com.hsbc.pack;*

*/\*\**

*\* Program to use static counter to count number of object which are instantiated.*

*\*/*

*public class Counter {*

*public static int count;*

*// Default Constructor*

*Counter()*

*{ //incrementing each time object is instantiated.*

*count++;*

*}*

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

*// TODO Auto-generated method stub*

*Counter c1 =new Counter();*

*Counter c2 =new Counter();*

*Counter c3 =new Counter();*

*Counter c4 =new Counter();*

*// Printing number of objects*

*System.out.println("Total number of Objects: "+c4.count);*

*}*

*}*

Lab Exercise No:16

Exercise Objective(s):*Construction of an object, this keyword, accessors(getters) and mutators(setters),*

*public and private access specifiers, instance and class member variables*

Exercise:*Create a class called Student with the following details: RollNo, StudName, MarksInEng,*

*MarksInMaths and MarksInScience. Write getters and setters for the all variables. RollNo*

*should be automatically generatedwhenever a newstudent is added.*

*Create a class called Standard with 8 students’ details and write separate method for each of*

*the following tasks and invoke the same.*

1. *To display the entire roll no and the name of the students in the class in the ascending order of roll no.*
2. *To display the roll no and the name of the student who has got the highest percentage.*
3. *To display the roll no and the name of the student who scored highest mark*

*inmathematics.*

1. *To display the roll no and the name of the student in the ascending order of the total marks in mathematics and science alone.*
2. *To display the roll no, name, total marks, percentage and rank of all the students in the descending order of rank.*

package com.hsbc.pack;

/\*

\* Construction of an object, this keyword, accessors(getters) and mutators(setters),

\* public and private access specifiers, instance and class member variables

\* \*/

import java.util.Arrays;

class Student implements Comparable<Student>{

private static int *count* = 0;

private static double *maxMarksInEng* = Double.*MIN\_VALUE*;

private static int *maxMarksEnglishRollNo* = -1;

private static double *maxMarksInMaths* = Double.*MIN\_VALUE*;

private static int *maxMarksMathsRollNo* = -1;

private static double *maxMarksInScience* = Double.*MIN\_VALUE*;

private static int *maxMarksScienceRollNo* = -1;

private int rollNo;

private String studName;

private double marksInEng;

private double marksInMaths;

private double marksInScience;

private double averageMarks;

private double total;

public Student(String studName, double marksInEng, double marksInMaths, double marksInScience) {

super();

*count*++;

this.rollNo = *count*;

this.studName = studName;

this.marksInEng = marksInEng;

this.marksInMaths = marksInMaths;

this.marksInScience = marksInScience;

this.total = (marksInEng + marksInMaths + marksInScience);

this.averageMarks = this.total / (double)3;

this.computeMaxMarks();

}

public void computeMaxMarks() {

if(this.marksInEng > *maxMarksInEng*) {

*maxMarksInEng* = this.marksInEng;

*maxMarksEnglishRollNo* = this.rollNo;

}

if(this.marksInMaths > *maxMarksInMaths*) {

*maxMarksInMaths* = this.marksInMaths;

*maxMarksMathsRollNo* = this.rollNo;

}

if(this.marksInScience > *maxMarksInScience*) {

*maxMarksInScience* = this.marksInScience;

*maxMarksScienceRollNo* = this.rollNo;

}

}

public static void getMathsMax() {

System.*out*.println("Max marks in Maths: " + *maxMarksInMaths*);

System.*out*.println("Obtained by Roll No.: " + *maxMarksMathsRollNo*);

System.*out*.println();

}

public static void getEnglishMax() {

System.*out*.println("Max marks in English: " + *maxMarksInEng*);

System.*out*.println("Obtained by Roll No.: " + *maxMarksEnglishRollNo*);

System.*out*.println();

}

public static void getScienceMax() {

System.*out*.println("Max marks in Science: " + *maxMarksInScience*);

System.*out*.println("Obtained by Roll No.: " + *maxMarksScienceRollNo*);

System.*out*.println();

}

public void getDetails() {

System.*out*.println("Roll No.:" + this.rollNo + " Name: " + this.studName);

}

public int getRollNo() {

return rollNo;

}

public void setRollNo(int rollNo) {

this.rollNo = rollNo;

}

public String getStudName() {

return studName;

}

public void setStudName(String studName) {

this.studName = studName;

}

public double getMarksInEng() {

return marksInEng;

}

public double getTotal() {

return total;

}

public void setTotal(double total) {

this.total = total;

}

public double getAverageMarks() {

return averageMarks;

}

public void setAverageMarks(double averageMarks) {

this.averageMarks = averageMarks;

}

public void setMarksInEng(double marksInEng) {

this.marksInEng = marksInEng;

}

public double getMarksInMaths() {

return marksInMaths;

}

public void setMarksInMaths(double marksInMaths) {

this.marksInMaths = marksInMaths;

}

public double getMarksInScience() {

return marksInScience;

}

public void setMarksInScience(double marksInScience) {

this.marksInScience = marksInScience;

}

@Override

public String toString() {

return "Student [rollNo=" + rollNo + ", studName=" + studName + ", averageMarks=" + averageMarks + ", total="

+ total + "]";

}

@Override

public int compareTo(Student student) {

return (int)(student.total - this.total);

}

}

public class Solution16 {

public static void main(String[] args) {

Student[] students = new Student[5];

students[0] = new Student("A", 70, 80, 90);

students[1] = new Student("B", 80, 70, 80);

students[2] = new Student("C", 65, 80, 90);

students[3] = new Student("D", 90, 90, 90);

students[4] = new Student("E", 70, 50, 50);

for(int i = 0; i < students.length; i++)

students[i].getDetails();

System.*out*.println();

Student.*getEnglishMax*();

Student.*getMathsMax*();

Student.*getScienceMax*();

System.*out*.println();

Arrays.*sort*(students);

for(int i = 0; i < students.length; i++) {

int rank = i + 1;

System.*out*.println("Position " + rank + " : " + students[i]);

}

}

}

Recommended duration:*40Mins*

Solution Guidance (if applicable): *NA*

Lab Exercise No:17

Exercise Objective(s):*String class, String immutability*

Exercise:*Write class that declares the following String.*

*“The quick brown fox jumps over the lazy dog”.*

*Perform the following modifications to the above string using appropriate methods.*

1. *Print the character at the 12th index.*
2. *Check whether the String contains the word “is”.*
3. *Add the string “and killed it” to the existing string.*
4. *Check whether the String ends with the word “dogs”.*
5. *Check whether the String is equal to “The quick brown Fox jumps over the lazy Dog”.*
6. *Check whether the String is equal to “*THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG*”.*
7. *Find the index position of the character “a”.*
8. *Find the last index position of the character “e”.*
9. *Find the length of the String.*
10. *Check whether the String matches to “The quick brown Fox jumps over the lazy Dog”.*
11. *Replace the word “The” with the word “A”.*
12. *Split the above string into two such that two animal names do not come together.*
13. *Print the animal names alone separately from the above string.*
14. *Print the above string in completely lower case.*
15. *Print the above string in completely upper case.*

Recommended duration:*30Mins*

Solution Guidance (if applicable): *NA*

package com.hsbc.pack;

public class Solution17 {

public static void main(String[] args) {

// TODO Auto-generated method stub

String str = "The quick brown fox jumps over the lazy dog";

System.out.println("12th index character is : "+ str.charAt(12));

System.out.println("Word \*is\* is present : "+ str.contains("is")); // 2.Check whether the String contains the word “is”.

str += " and killed it";

System.out.println("Appended string : "+ str); //3.Add the string “and killed it” to the existing string.

System.out.println("String ends with \*dogs\* : "+ str.endsWith("dogs")); // 4.Check whether the String ends with the word “dogs”.

String checkingString = "The quick brown Fox jumps over the lazy Dog";

String checkingString2 = "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG";

System.out.println("String is equal with (The quick brown Fox jumps over the lazy Dog) : "+ (str==checkingString)); //5.Check whether the String is equal to “The quick brown Fox jumps over the lazy Dog”.

System.out.println("String is equal with (THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG) : "+ (str==checkingString2)); //6.Check whether the String is equal to “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”.

System.out.println("Index of char 'a' is : "+ str.indexOf("a")); //7.Find the index position of the character “a”.

System.out.println("Last index of char 'e' is : "+ str.lastIndexOf("e")); //8.Find the last index position of the character “e”.

System.out.println("Length of string is "+ str.length()); //9.Find the length of the String.

System.out.println("String is matches with (The quick brown Fox jumps over the lazy Dog) : "+ (str.matches(checkingString))); //10.Check whether the String matches to “The quick brown Fox jumps over the lazy Dog”.

System.out.println("After replacing word 'The' with 'A' : "+ str.replace("The", "A")); //11.Replace the word “The” with the word “A”.

System.out.println("Split string as 2 animal separates : 1st string is -- "+ str.substring(0, str.indexOf("dog")) + " -- and 2nd string is -- " + str.substring(str.indexOf("dog")) ); //12.Split the above string into two such that two animal names do not come together.

String brwn = "brown";

String dg = "dog";

System.out.println("Animal names only : " + str.substring(str.indexOf("fox"), str.indexOf("fox")+3 ) + " " + str.substring(str.indexOf("dog"), str.indexOf("dog")+3 ) ); //13.Print the animal names alone separately from the above string.

System.out.println("Lowercase : "+ str.toLowerCase()); //14.Print the above string in completely lower case.

System.out.println("Uppercase : "+ str.toUpperCase()); //15.Print the above string in completely upper case.

}

}

Lab Exercise No:18

Exercise Objective(s):*String class, String immutability*

Exercise:*Write a program to demonstrate the difference between equals and == operator with*

*appropriate example.*

Recommended duration:*15Mins*

Solution Guidance (if applicable): *NA*

package com.hsbc.pack;

/\*

\* A program to demonstrate difference betweeen equals and '==' operator.

\* \*/

public class TestString {

public static void main(String[] args) {

// TODO Auto-generated method stub

String s1 = new String("Hello");

String s2 = new String("Hello");

// will return true

System.out.println(s1.equals(s2));

// will return false

System.out.println(s1==s2);

}

}

Lab Exercise No:19

Exercise Objective(s):*Arrays*

Exercise:*Write a program to declare an array with 8 elements and copy the 8 elements into another*

*array and display the same.*

Recommended duration:*15Mins*

Solution Guidance (if applicable): *NA*

*package com.hsbc.pack;*

*/\**

*\* A program to copy array.*

*\* \*/*

*public class TestArray {*

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

*// TODO Auto-generated method stub*

*int arr1[] = {1,2,3,4,5,6,7,8};*

*// creating array of same size*

*int arr2[] = new int[arr1.length];*

*for(int i=0;i<arr1.length;i++)*

*{ // making copy of elements*

*arr2[i] = arr1[i];*

*}*

*System.out.println("Content of first array: ");*

*for(int i=0;i<arr1.length;i++)*

*{ // Printing elements of first array*

*System.out.println(arr1[i]);*

*}*

*System.out.println("Content of second array: ");*

*for(int i=0;i<arr1.length;i++)*

*{ // Printing elements of second array*

*System.out.println(arr2[i]);*

*}*

*}*

*}*

Lab Exercise No:20

Exercise Objective(s):*Arrays*

Exercise:*Write a program to display the sum and the average of elements in the array.*

Recommended duration:*15Mins*

Solution Guidance (if applicable): *NA*

*package com.hsbc.pack;*

*/\*\**

*\* Program to find sum and average of an array.*

*\*/*

*public class Average {*

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

*// TODO Auto-generated method stub*

*// declaring an array*

*double[] arr = {10,20,30,40,50};*

*double sum = 0;*

*for(int i=0;i<arr.length;i++)*

*{ // finding sum*

*sum = sum + arr[i];*

*}*

*System.out.println("Average of array: "+(sum/arr.length));*

*System.out.println("Sum of array: "+sum);*

*}*

*}*

Lab Exercise No:21

Exercise Objective(s):*Arrays*

Exercise:*Write a program to construct two matrices and display the sum of those.*

Recommended duration:*20Mins*

Solution Guidance (if applicable): *NA*

*package com.hsbc.pack;*

*/\*\**

*\* Program to add two matrix.*

*\*/*

*import java.util.Scanner;*

*public class ArraySum {*

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

*// TODO Auto-generated method stub*

*int row, col, i, j;*

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

*System.out.println("Enter Rows and Column: ");*

*row = in.nextInt();*

*col = in.nextInt();*

*int first[][] = new int[row][col]; //creating matrix*

*int second[][] = new int[row][col];*

*int sum[][] = new int[row][col];*

*System.out.println("Values of First matrix"); //input of matrix*

*for (i = 0; i < row; i++)*

*for (j = 0; j < col; j++)*

*first[i][j] = in.nextInt();*

*System.out.println("Values of Second matrix");*

*for (i = 0 ; i < row; i++)*

*for (j = 0 ; j < col; j++)*

*second[i][j] = in.nextInt();*

*for (i = 0; i < row; i++)*

*for (j = 0; j < col; j++)*

*sum[i][j] = first[i][j] + second[i][j];*

*System.out.println("Sum of the matrices:");*

*for (i = 0; i < row; i++)*

*{*

*for (j = 0; j < col; j++)*

*System.out.print(sum[i][j] + "\t");*

*System.out.println();*

*}*

*}*

*}*

Lab Exercise No:22

Exercise Objective(s):*Arrays*

Exercise:*Write a program to display the square of the elements of a two dimensional array.*

Recommended duration:*20Mins*

Solution Guidance (if applicable): *NA*

*package com.hsbc.pack;*

*import java.util.Scanner;*

*/\**

*\* Program to find the square of each elements of the matrix.*

*\*/*

*public class MatrixSquare {*

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

*// TODO Auto-generated method stub*

*int row, col, i, j;*

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

*System.out.println("Value of Rows and Columns of Matrix");*

*row = s.nextInt();*

*col = s.nextInt();*

*// Matrix creation*

*int first[][] = new int[row][col];*

*int square[][] = new int[row][col];*

*System.out.println("Elements of first Matrix");*

*for (i = 0; i < row; i++)*

*for (j = 0; j < col; j++)*

*first[i][j] = s.nextInt();*

*for (i = 0; i < row; i++)*

*{*

*for (j = 0; j < col; j++)*

*{ square[i][j] = first[i][j] \* first[i][j]; }*

*}*

*System.out.println("Square of the matrices:");*

*for (i = 0; i < row; i++)*

*{*

*for (j = 0; j < col; j++)*

*System.out.print(square[i][j] + " ");*

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

*}*

*}*

*}*

Lab Exercise No:23

Exercise Objective(s):*Arrays*

Exercise:*Write a program to construct an array with 10 elements and to find the number of*

*occurrences of each element in the Array.*

Recommended duration:*20Mins*

Solution Guidance (if applicable): *NA*

*package com.hsbc.pack;*

*import java.util.Scanner;*

*/\**

*\* A program to count occurrence of each element of array.*

*\* \*/*

*public class CountElement {*

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

*// TODO Auto-generated method stub*

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

*int[] arr = new int[10];*

*int[] frequency = new int[10];*

*int size, i, j, count;*

*System.out.println("Enter Size of Array upto 10: ");*

*size = sc.nextInt();*

*System.out.println("Enter values of matrix");*

*for(i=0; i<size; i++)*

*{ arr[i] = sc.nextInt();*

*frequency[i] = -1;*

*}*

*for(i=0; i<size; i++)*

*{*

*count = 1;*

*for(j=i+1; j<size; j++)*

*{ //finding duplicate element*

*if(arr[i]==arr[j])*

*{ count++;*

*frequency[j] = 0;*

*}*

*}*

*if(frequency[i] != 0)*

*{ frequency[i] = count; }*

*}*

*System.out.println("Frequency of all elements : ");*

*for(i=0; i<size; i++)*

*{ if(frequency[i] != 0)*

*{ System.out.println(arr[i] +" occurs for "+frequency[i]+" times!!!"); }*

*}*

*}*

*}*

Lab Exercise No:24

Exercise Objective(s):*Overloading*

Exercise:*Create a class called shape with the following methods*

1. *area*
2. *perimeter*

*Overload the area and perimeter method to calculate for both square and rectangle.*

*Create a main class and invoke the area method to calculate the area of the square and*

*rectangle. Also invoke the perimeter method to calculate the perimeter of the square*

*and rectangle.*

Recommended duration:*20Mins*

Solution Guidance (if applicable): *NA*

*package com.hsbc.pack;*

*/\**

*\* Program to give example of we can do method!!*

*\* \*/*

*public class OverLoad {*

*public void area(int height)*

*{*

*int area=height\*height;*

*System.out.println("area of square:"+area);*

*}*

*public void area(int height,int length)*

*{*

*int area=length\*height;*

*System.out.println("area of rectangle:"+area);*

*}*

*public void perimeter(int height)*

*{*

*int perimeter=4\*height;*

*System.out.println("perimeter of square:"+perimeter);*

*}*

*public void perimeter(int height,int length)*

*{*

*int perimeter=2\*(length+height);*

*System.out.println("area of rectangle:"+perimeter);*

*}*

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

*{*

*OverLoad obj = new OverLoad();*

*int a=5,b=10;*

*obj.area(a);*

*obj.area(a,b);*

*obj.perimeter(a);*

*obj.perimeter(a,b);*

*}*

*}*

Lab Exercise No:25

Exercise Objective(s):*Overloading*

Exercise: *Create a class called employee with the following data members*

1. *empName*
2. *empId*
3. *empAge*
4. *empdesgn*
5. *empLocation*
6. *empExpInYrs*

*All these data members should be initialized using constructors. Use constructor overloading*

*and demonstrate by creating different employee objects with*

1. *Employee name alone*
2. *Employee name and id*
3. *Employee name, id and age*
4. *Employee name, id and designation*
5. *Employee name, id, age and designation*
6. *Employee name, id, age and location*
7. *Employee name, id, age and experience*
8. *Employee name, id, designation and experience*
9. *Employee name, id, designation, location and experience*
10. *Employee name, id, age, designation, location and experience*

Recommended duration:*20Mins*

Solution Guidance (if applicable): *NA*

package com.hsbc.pack;

/\*

\* Constuctor Overloading.

\* \*/

public class EmpData {

private String empName;

private int empId;

private int empAge;

private String empdesgn;

private String empLocation;

private int empExpInYrs;

EmpData()

{

// Default constructor

this.empName="JP";

empId=1;

empAge=22;

empdesgn="Trainee software engineer";

empLocation="Pune";

empExpInYrs=1;

}

EmpData(String empName)

{

// First parameterized constructor

this.empName=empName;

empId=1;

empAge=22;

empdesgn="Trainee software engineer";

empLocation="Pune";

empExpInYrs=1;

}

EmpData(String empName, int empId)

{

// Second parameterized constructor

this.empName=empName;

this.empId=empId;

empAge=22;

empdesgn="Trainee software engineer";

empLocation="Pune";

empExpInYrs=1;

}

EmpData(String empName, int empId, int empAge)

{ // Third parameterized constructor

this.empName=empName;

this.empId=empId;

this.empAge=empAge;

empdesgn="Trainee software engineer";

empLocation="Pune";

empExpInYrs=1;

}

EmpData(String empName,int empId, int empAge, String empdesgn)

{ // Fourth parameterized constructor

this.empName=empName;

this.empId=empId;

this.empAge=empAge;

this.empdesgn=empdesgn;

empLocation="Pune";

empExpInYrs=1;

}

EmpData(String empName,int empId, int empAge, String empdesgn, String empLocation)

{ // Fifth parameterized constructor

this.empName=empName;

this.empId=empId;

this.empAge=empAge;

this.empdesgn=empdesgn;

this.empLocation=empLocation;

empExpInYrs=1;

}

EmpData(String empName,int empId, int empAge, String empdesgn, String empLocation, int empExpInyrs)

{

//Last parameterized constructor

this.empName=empName;

this.empId=empId;

this.empAge=empAge;

this.empdesgn=empdesgn;

this.empLocation=empLocation;

this.empExpInYrs= empExpInyrs;

}

public static void main(String args[])

{

// Calling the default constructor

EmpData obj = new EmpData();

EmpData obj1 =new EmpData("JP");

EmpData obj2 = new EmpData("JP",1);

EmpData obj3 = new EmpData("JP",1,10);

EmpData obj4 = new EmpData("JP",1,10,"TSE");

EmpData obj5 = new EmpData("JP",1,10,"TSE","Pune");

EmpData obj6 = new EmpData("JP",1,10,"TSE","Pune",0);

}

}

Lab Exercise No:26

Exercise Objective(s):*Overloading*

Exercise:*Create a class called Calculator which has 4 different methods add, diff, mul and div which*

*accepts two numbers as parameters. Overload the methods such that the parameters can be*

*of the following pattern.*

1. *Both are of int data type.*
2. *Both are of double data type.*
3. *First parameter is of int data type and second parameter is of double data type.*
4. *First parameter is of double data type and second parameter is of int data type.*

*Create anobject to access these methods and invoke these methods with different type of*

*numbers and display the result in the corresponding methods.*

Recommended duration:*20Mins*

Solution Guidance (if applicable): *Re-use the code from Lab Exercise 14*

*package com.hsbc.pack;*

*/\**

*\* A program to overload to methods of calculator for int and double data types.*

*\*/*

*class Calculator {*

*public void add(int a, int b) {*

*System.out.println("The sum of a and b is " + (a+b));*

*}*

*public void add(double a, double b) {*

*System.out.println("The sum of c and d is " + (a+b));*

*}*

*public void add(double a, int b) {*

*System.out.println("The sum of c and a is " + (a+(double)b));*

*}*

*public void add(int a, double b) {*

*System.out.println("The sum of b and d is " + (b+(double)a));*

*}*

*public void diff(int a, int b) {*

*System.out.println("The difference of a and b is " + (a-b));*

*}*

*public void diff(double a, double b) {*

*System.out.println("The difference of c and d is " + (a-b));*

*}*

*public void diff(double a, int b) {*

*System.out.println("The difference of c and a is " + (a-(double)b));*

*}*

*public void diff(int a, double b) {*

*System.out.println("The difference of b and d is " + ((double)a-b));*

*}*

*public void mul(int a, int b) {*

*System.out.println("The product of a and b is " + (a\*b));*

*}*

*public void mul(double a, double b) {*

*System.out.println("The product of c and d is " + (a\*b));*

*}*

*public void mul(double a, int b) {*

*System.out.println("The product of c and a is " + (a\*(double)b));*

*}*

*public void mul(int a, double b) {*

*System.out.println("The product of b and d is " + ((double)a\*b));*

*}*

*public void div(int a, int b) {*

*try {*

*float c = (float)a/b;*

*System.out.println("The division of a and b is " + c);*

*}*

*catch(ArithmeticException e)*

*{*

*System.out.println("Message: " + e);*

*}*

*}*

*public void div(double a, double b) {*

*try {*

*double c = a/b;*

*System.out.println("The division of c and d is " + c);*

*}*

*catch(ArithmeticException e)*

*{*

*System.out.println("Message: " + e);*

*}*

*}*

*public void div(int a, double b) {*

*try {*

*double c = (double)a/b;*

*System.out.println("The division of b and d is " + c);*

*}*

*catch(ArithmeticException e)*

*{*

*System.out.println("Message: " + e);*

*}*

*}*

*public void div(double a, int b) {*

*try {*

*double c = a/(double)b;*

*System.out.println("The division of c and a is " + c);*

*}*

*catch(ArithmeticException e)*

*{*

*System.out.println("Message: " + e);*

*}*

*}*

*}*

*public class OverLoadCalC {*

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

*int a = 4;*

*int b = 8;*

*double c = 34.567;*

*double d = 56.234;*

*Calculator cal = new Calculator();*

*cal.add(a, b);*

*cal.add(c, d);*

*cal.add(c, a);*

*cal.add(b, d);*

*cal.diff(a, b);*

*cal.diff(c, d);*

*cal.diff(c, a);*

*cal.diff(b, d);*

*cal.mul(a, b);*

*cal.mul(c, d);*

*cal.mul(c, a);*

*cal.mul(b, d);*

*cal.div(a, b);*

*cal.div(c, d);*

*cal.div(c, a);*

*cal.div(b, d);*

*}*

*}*

Lab Exercise No:27

Exercise Objective(s):*Initializers*

Exercise:*Write a class called Computer such that the object of that class should be created only when*

*the class is loaded.*

Recommended duration:*20Mins*

Solution Guidance (if applicable): *NA*

*package com.hsbc.pack;*

*/\**

*\* A program to explain Initializer in java.*

*\* \*/*

*public class Computer {*

*{*

*System.out.println("Initializer Example");*

*System.out.println("Running before each time class is loaded.");*

*}*

*public int a;*

*Computer(){*

*this.a = 10;*

*}*

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

*// TODO Auto-generated method stub*

*Computer obj = new Computer();*

*System.out.println(obj.a);*

*}*

*}*

Lab Exercise No:28

Exercise Objective(s):*Var-args*

Exercise:*In the calculator (Lab exercise - 14) program, make the add and diff method to accept var-args*

*and demonstrate.*

Recommended duration:*20Mins*

Solution Guidance (if applicable):*NA*

*package com.hsbc.pack;*

/\*

\* Program to display sum and difference of more than 2 numbers.

\*/

public class Solution28 {

public static void main(String[] args) {

// TODO Auto-generated method stub

add(2, 3, 5, 11, 27, 13);

sub(100, 20, 50, 10, 33);

}

// This will add all the arguments.

public static void add(int ...num)

{

int sum = 0;

for(int i = 0; i < num.length; i++)

{

sum += num[i];

}

System.out.println("Addition of numbers is "+ sum);

}

// This will subtract all the elements from first element.

public static void sub(int ...num)

{

int sub = num[0];

for(int i = 1; i < num.length; i++)

{

sub -= num[i];

}

System.out.println("Subtraction of numbers is "+ sub);

}

}