**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.*

**Input: 4 2**

**Output:** Addition is=6

Subtraction is=2

Multiplication is=8

Division is=2

**SOLUTION**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Concept of class and object\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public static Calculator

{

public void add(int num1,int num2)

{

int num3=num1+num2;

System.out.println(“Addition is=”+ num3);

}

public void sub(int num1,int num2)

{

int num3=num1-num2;

System.out.println(“Subtraction is=”+ num3);

}

public void mul(int num1,int num2)

{

int num3=num1\*num2;

System.out.println(“Multiplication is=”+ num3);

}

public void div(int num1,int num2)

{

int num3=num1/num2;

System.out.println(“Division is=”+ num3);

}

public static void main(String args[])

{

Calculator ob=new Calculator(); //creation of object

ob.add(4,2);

ob.sub(4,2);

ob.mul(4,2);

ob.div(4,2);

}

}

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

**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.*

**Input:** No input

**Output:** 2

**SOLUTION**

*// \*\*\*\*\*\*\*\*\*\* program Find Out the Number of Objects Created \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*class Sample {*

*static int noOfObjects = 0;*

*}*

*Public Sample (String s)*

*{*

*noOfObjects += 1;*

*}*

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

*{*

*Sample s3 = new Sample (" OBJECT 1");*

*Sample s3 = new Sample ("OBJECT 2");*

*System.out.println(Test.noOfObjects);*

*}*

*}*

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

**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.*

**Input:** No input

**Output:** Roll No. & Name credentials(ascending based on roll no.)

1 | Aiman

2 | Aditi

3 | Aayushi

4 | Neha

5 | Pallavi

6 | Suchit

7 | Varun

8 | Sakshi

Topper:

Roll No. : 7 | Name : Varun

Topper in Maths:

Roll No. : 8 | Name : Sakshi

List of Maths & Science sum marks (Ascending order based on marks):

Roll No. | Name | Sum marks in Maths & Science

2 | Aditi | 180

3 | Aayushi | 184

8 | Sakshi | 184

5 | Pallavi | 185

1 | Aiman | 189

6 | Suchit | 191

4 | Neha | 192

7 | Varun | 194

Marklist with percentage(descending based on rank):

Rank | Roll No. | Name | Total Marks | Percentage

0 | 2 | Aditi | 272 | 90.666664

1 | 3 | Aayushi | 275 | 91.666664

2 | 1 | Aiman | 278 | 92.666664

3 | 4 | Neha | 280 | 93.333336

4 | 8 | Sakshi | 282 | 94.0

5 | 5 | Pallavi | 283 | 94.333336

6 | 6 | Suchit | 290 | 96.666664

7 | 7 | Varun | 292 | 97.333336

**SOLUTION:**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*understanding class object and its methods\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class Student {

private static int ctr = 0;

private int rollNo, marksInEng, marksInMaths ,marksInScience;

private String studName;

public Student() {

ctr++;

this.rollNo = ctr;

}

public int getRollNo() {

return rollNo;

}

public int getMarksInEng() {

return marksInEng;

}

public void setMarksInEng(int marksInEng) {

this.marksInEng = marksInEng;

}

public int getMarksInMaths() {

return marksInMaths;

}

public void setMarksInMaths(int marksInMaths) {

this.marksInMaths = marksInMaths;

}

public int getMarksInScience() {

return marksInScience;

}

public void setMarksInScience(int marksInScience) {

this.marksInScience = marksInScience;

}

public String getStudName() {

return studName;

}

public void setStudName(String studName) {

this.studName = studName;

}

}

public class Standard {

public static void func1(Student[] s) {

System.out.println("Roll No. & Name credentials(ascending based on roll no.)");

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

System.out.println(s[i].getRollNo() + " | " + s[i].getStudName());

}

}

public static void func2(Student[] s) {

System.out.println("Topper:");

int[] tot = new int[8];

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

{

tot[i] = s[i].getMarksInEng() + s[i].getMarksInMaths() + s[i].getMarksInScience();

}

int max = tot[0];

int v=0;

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

if(tot[i]>max) {

max = tot[i];

v = i;

}

}

System.out.println("Roll No. : " + s[v].getRollNo() + " | Name : " + s[v].getStudName());

}

public static void func3(Student[] s) {

System.out.println("Topper in Maths:");

int[] m = new int[8];

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

m[i] = s[i].getMarksInMaths();

}

int max = m[0];

int v = 0;

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

if(m[i]>max) {

max = m[i];

v = i;

}

}

System.out.println("Roll No. : " + s[v].getRollNo() + " | Name : " + s[v].getStudName());

}

public static void func4(Student[] s) {

System.out.println("List of Maths & Science sum marks (Ascending order based on marks):");

System.out.println("Roll No. | Name | Sum marks in Maths & Science ");

int[] ms = new int[8];

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

ms[i] = s[i].getMarksInMaths() + s[i].getMarksInScience();

}

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

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

{

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

{

if(ms[i]>ms[j]) {

ms[i] = ms[i]+ms[j];

ms[j] = ms[i]-ms[j];

ms[i] = ms[i]-ms[j];

r[i] = r[i]+r[j];

r[j] = r[i]-r[j];

r[i] = r[i]-r[j];

}

}

}

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

System.out.println(r[i] + " | " + s[r[i]-1].getStudName() + " | " + ms[i]);

}

}

public static void func5(Student[] s) {

System.out.println("Marklist with percentage(descending based on rank):");

System.out.println("Rank | Roll No. | Name | Total Marks | Percentage ");

int[] tot = new int[8];

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

{

tot[i] = s[i].getMarksInEng() + s[i].getMarksInMaths() + s[i].getMarksInScience();

}

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

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

{

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

{

if(tot[i]>tot[j]) {

tot[i] = tot[i]+tot[j];

tot[j] = tot[i]-tot[j];

tot[i] = tot[i]-tot[j];

r[i] = r[i]+r[j];

r[j] = r[i]-r[j];

r[i] = r[i]-r[j];

}

}

}

float[] perc = new float[8];

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

perc[i] = (float)tot[i]/3;

}

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

System.out.println(i + " | " + s[r[i]-1].getRollNo() + " | " + s[r[i]-1].getStudName() + " | " + tot[i] + " | " + perc[i]);

}

}

public static void main(String[] args) {

Student[] s = new Student[8];

Student s1 = new Student();

s1.setStudName("Aiman");

s1.setMarksInEng(89);

s1.setMarksInMaths(95);

s1.setMarksInScience(94);

s[0] = s1;

Student s2 = new Student();

s2.setStudName("Aditi");

s2.setMarksInEng(92);

s2.setMarksInMaths(96);

s2.setMarksInScience(84);

s[1] = s2;

Student s3 = new Student();

s3.setStudName("Aayushi");

s3.setMarksInEng(91);

s3.setMarksInMaths(90);

s3.setMarksInScience(94);

s[2] = s3;

Student s4 = new Student();

s4.setStudName("Neha");

s4.setMarksInEng(88);

s4.setMarksInMaths(98);

s4.setMarksInScience(94);

s[3] = s4;

Student s5 = new Student();

s5.setStudName("Pallavi");

s5.setMarksInEng(98);

s5.setMarksInMaths(91);

s5.setMarksInScience(94);

s[4] = s5;

Student s6 = new Student();

s6.setStudName("Suchit");

s6.setMarksInEng(99);

s6.setMarksInMaths(97);

s6.setMarksInScience(94);

s[5] = s6;

Student s7 = new Student();

s7.setStudName("Varun");

s7.setMarksInEng(98);

s7.setMarksInMaths(95);

s7.setMarksInScience(99);

s[6] = s7;

Student s8 = new Student();

s8.setStudName("Sakshi");

s8.setMarksInEng(98);

s8.setMarksInMaths(100);

s8.setMarksInScience(84);

s[7] = s8;

func1(s);

func2(s);

func3(s);

func4(s);

func5(s);

}

}

--------------------------------------------------------------------------------------------------------------------------------------

**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.*

**Input:**

**Output:** Character at 12th index r

Doesn’t not contain 'is'

After appending: The quick brown fox jumps over the lazy dog and killed it

Ends with 'dog'

Not Equal

Not Equal

36

33

43

Not Equal

After replacing A quick brown fox jumps over A lazy

After splitting

The

quick

brown

fox

jumps

over

the

lazy

dog

Animal Names

fox

dog

String in lowercase: the quick brown fox jumps over the lazy dog

String in uppercase: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

**SOLUTION**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Understanding String class and its methods\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Solution17 {

public static void main(String[] args) {

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

System.out.println("Character at 12th index "+ s.charAt(11));

//checking if string contains 'is'

String word="";

boolean flag=false;

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

{

char ch=s.charAt(i);

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("is"))

{

System.out.println("Contains 'is'");

flag=true;

break;

}

else

{

word="";

}

}

}

if (flag==false)

System.out.println("Doesn not contain 'is'");

//adding 'and killed it' to existing string

String a=s+" and killed it";

System.out.println("After appending: "+a);

//checking if string ends with dogs

word="";

for(int i=s.length()-1;i<s.length();i--)

{

char ch=s.charAt(i);

if(Character.isLetter(ch))

{

word=ch+word;

}

else

{

if(word.equals("dog"))

{

System.out.println("Ends with 'dog'");

break;

}

else

{

System.out.println("Does not end with 'dog'");

}

}

}

// Checking whether the String is equal to “The quick brown Fox jumps over the lazy Dog”

if(s.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

System.out.println("Equal");

else

System.out.println("Not Equal");

//Checking whether the String is equal to “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”

if(s.compareTo("THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG")==0)

System.out.println("Equal");

else

System.out.println("Not Equal");

//Finding the index position of the character a

System.out.println(s.indexOf("a"));

//Finding the last index position of the character “e”

System.out.println(s.lastIndexOf("e"));

//Finding the length of the String.

System.out.println(s.length());

//Checking whether the String matches to “The quick brown Fox jumps over the lazy Dog”

if(s.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

System.out.println("Equal");

else

System.out.println("Not Equal");

//Replacing the word “The” with the word “A”

word="";

String t="";

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

{

char ch=s.charAt(i);

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("The")||word.equals("the"))

{

t=t+" A";

word="";

}

else

{

t=t+" "+word;

word="";

}

}

}

System.out.println("After replacing "+ t);

//Splitting the above string into two such that two animal names do not come together

String d[]=s.split(" ",s.length());

System.out.println("After splitting");

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

{

System.out.println(d[i]);

}

//Printing the animal names alone separately from the above string

t=s+" ";

word="";

System.out.println("Animal Names");

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

{

char ch=t.charAt(i);

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("fox")||word.equals("dog"))

{

System.out.println(word);

word="";

}

else

{

word="";

}

}

}

//Printing the above string in completely lower case

System.out.println("String in lowercase: "+s.toLowerCase());

//Printing the above string in completely upper case

System.out.println("String in uppercase: "+s.toUpperCase());

}

}

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

**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.*

**Input:**

**Output:** s1!=s2

s1 equals s2

**SOLUTION**

*//* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*To demonstrate the immutability of String class\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Solution18 {

public static void main(String[] args) {

String s1 = new String("Comparing == and equals function");

String s2 = new String("Comparing == and equals function");

if(s1 == s2)

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

else

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

if(s1.equals(s2))

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

else

System.out.println("s1 not equals 2");

}

}

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

**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.*

**Input:** No input

**Output:** Contents of a[]

1 8 3

Contents of b[]

1 8 3

**SOLUTION:**

//\*\*\*\*\*\*\*\*\*\*\*understanding concept of 1D array\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Test

{

    public static void main(String[] args)

    {

        int a[] = {1, 8, 3};

        int b[] = new int[a.length];              // Create an array b[] of same size as a[]

        for (int i=0; i<a.length; i++)              // Copy elements of a[] to b[]

            b[i] = a[i];

        System.out.println("Contents of a[] ");    // printing array a[]

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

            System.out.print(a[i] + " ");

        System.out.println("\n\nContents of b[] "); // printing array b[]

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

            System.out.print(b[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.*

**Input:** No input

**Output:** The average is:52.41799999999999

The average is:262.09

**SOLUTION:**

//\*\*\*\*\*\*\*\*\*\*\*understanding concept of 1D array\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Average {

public static void main(String[] args)

 {

        double[] arr = {19, 12.89, 16.5, 200, 13.7};

        double total = 0;

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

         total = total + arr[i];

        }

        double average = total / arr.length;

        System.out.println("The average is:"+ average);

System.out.println("The average is:"+ total);

     }

}

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

**Lab Exercise No:**21

**Exercise Objective(s):***Arrays*

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

**Input:** No input

**Output:**

Sum of the matrices

2 3 4

5 6 7

8 9 10

**SOLUTION:**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*understanding 2D array and operations performed over them\*\*\*\*\*\*\*\*\*\*

public class AddMatrix

{

  public static void main(String args[])

  {

    int row, column, c, d;

row=3;

column=3;

 int first[][] = {{1,2,3},{4,5,6},{7,8,9}}     ;

 int second[][] = {1,1,1},{1,1,1},{1,1,1}};

    for (c = 0; c < row; c++)

       for (d = 0; d < column; d++)

        sum[c][d] = first[c][d] + second[c][d];

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

    for (c = 0; c < row; c++)

    {

       for (d = 0; d < column; d++)

         System.out.print(sum[c][d] + "\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.*

**Input:** No input

**Output:**

Square of the matrices:

1 4 9

16 25 36

49 64 81

**SOLUTION:**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*understanding 2D array and operations performed over them\*\*\*\*\*\*\*\*\*\*

public class SquareMatrix

{

  public static void main(String args[])

  {

    int row, column, c, d;

    row = 3;

    column = 3;

    int first[][] = {{1,2,3},{4,5,6},{7,8,9}}      //creating and initializing matrix

    int square[][] = new int[row][column];

    for (c = 0; c < row; c++)

      for (d = 0; d < column; d++)

        square[c][d] = first[c][d] \* first[c][d];   //squaring each element through loop

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

    for (c = 0; c < row; c++)

    {

      for (d = 0; d < column; d++)

        System.out.print(square[c][d] + "\t");

      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.*

**Input:** 1 2 3 1 2 3 1 2 3 4

**Output:**  1 occurs 3 times

2 occurs 3 times

3 occurs 3 times

4 occurs 1 times

**SOLUTION:**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Understanding the concept of arrays \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Frequancy

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int[] arr = new int[100];

int[] freq = new int[100];

int size, i, j, count;

System.out.println(“nEnter size of array: “);

size = sc.nextInt();                      // general program not just for 10 numbers

System.out.println(“Enter elements in array: “);

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

{

arr[i] = sc.nextInt();        //inputing array

freq[i] = -1;

}

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

{

count = 1;

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

{

if(arr[i]==arr[j])     //finding duplicate element

{

count++;

freq[j] = 0;

}

}

if(freq[i] != 0)

{

freq[i] = count;

}

}

System.out.println(“nFrequency of all elements of array : n”);

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

{

if(freq[i] != 0)

{

System.out.println(arr[i] + ” occurs ” + freq[i] + ” times” + “\n”);

}

}

}

}

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

**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.*

**Input:** 5 10

**Output:** area of sqare:25

area of rectangle:50

perimeter of sqare:20

area of rectangle:30

**SOLUTION:**

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Understanding the method overloading\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class CalcArea

{

public void area(int height)

{

int area=height\*height;

System.out.println("area of sqare:"+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 sqare:"+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[])

{

CalcArea c=new CalcArea();

int a=5,b=10;

area(a);

area(a,b);

perimeter(a);

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*

**Input:**

**Output** : aiman 1,22 TSE Pune 1

aiman 1,22 TSE Pune 1

aiman 1,22 TSE Pune 1

aiman 1,22 TSE Pune 1

aiman 1,22 TSE Pune 1

aiman 1,22 TSE Pune 1

aiman 1,22 TSE Pune 1

aiman 1,22 TSE Pune 1

aiman 1,22 TSE Pune 1

aiman 1,22 TSE Pune 1

**SOLUTION:**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Understanding the Constructor overloading \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Employee

{

private String empName;

private int empId;

private int empAge;

private String empdesgn;

private String empLocation;

private int empExpInYrs;

Employee( )

{

}

Employee (String empName)

{

this.empName=empName;

empId=1;

empAge=22;

empdesgn="Trainee software engineer";

empLocation="Pune";

empExpInYrs=1;

}

Employee (String empName, int empId)

{

this.empName=empName;

this.empId=empId;

empAge=22;

empdesgn="Trainee software engineer";

empLocation="Pune";

empExpInYrs=1;

}

Employee (String empName, int empId, int empAge)

{

this.empName=empName;

this.empId=empId;

this.empAge=empAge;

empdesgn="Trainee software engineer";

empLocation="Pune";

empExpInYrs=1;

}

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

{

this.empName=empName;

this.empId=empId;

this.empAge=empAge;

this.empdesgn=empdesgn;

empLocation="Pune";

empExpInYrs=1;

}

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

{

this.empName=empName;

this.empId=empId;

this.empAge=empAge;

this.empdesgn=empdesgn;

this.empLocation=empLocation;

empExpInYrs=1;

}

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

{

//Default constructor

this.empName=empName;

this.empId=empId;

this.empAge=empAge;

this.empdesgn=empdesgn;

this.empLocation=empLocation;

this.empExpInYrs=empExpInyrs;

}

void display(Employee e)

{

System.out.println(e.empName+" "+e.empId+" "+e.empAge+" "+e.empdesg+" "+e.empLocation+" "+e.empExpInyrs+" ");

}

public static void main(String args[])

{

//This object creation would call the default constructor

Employee myobj = new Employee();

myobj.Employee("Aiman");

myobj.display(myobj);

myobj.Employee ("aiman",1);

myobj.display(myobj);

myobj.Employee ("aiman",1,22);

myobj.display(myobj);

myobj.Employee ("aiman",1,22, "TSE");

myobj.display(myobj);

myobj.Employee ("aiman",1,22, "TSE","pune" );

myobj.display(myobj);

myobj.Employee ("aiman",1,22,"TSE","Pune",1);

myobj.display(myobj);

}

}

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

**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 an object to access these methods and invoke these methods with different type of*

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

**Input:** No input

**Output:** The sum of a and b is 99

The sum of c and d is 690.801

The sum of c and a is 148.567

The sum of b and d is 641.234

The difference of a and b is -71

The difference of c and d is -421.66700000000003

The difference of c and a is 120.56700000000001

The difference of b and d is -471.23400000000004

The product of a and b is 1190

The product of c and d is 74850.740678

The product of c and a is 1883.938

The product of b and d is 47279.89000000001

The division of a and b is 0.16470589

The division of c and d is 0.24192516099339487

The division of c and a is 9.611928571428573

The division of b and d is 0.1528133842950988

**SOLUTION**

public 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 Solution26 {

public static void main(String[] args) {

int a = 14;

int b = 85;

double c = 134.567;

double d = 556.234;

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.*

**Input:** No input

**Output:** Running before each time class is loaded.

10

**SOLUTION:**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   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.*

**Input:** No input

**Output:** argument length: 2

sum2 = 6

argument length: 3

sum3 = 9

argument length: 4

diff1 = -12

argument length: 2

diff2 = -8

**SOLUTION:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*to understand the concept of variable arguments\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class VarargExample {

public int sumNumber(int ... args){

System.out.println("argument length: " + args.length);

int sum = 0;

for(int x: args){

sum += x;

}

return sum;

}

public int diffNumber(int ... args){

System.out.println("argument length: " + args.length);

int sum = 0;

for(int x: args){

sum -= x;

}

return sum;

}

public static void main( String[] args ) {

VarargExample ex = new VarargExample();

int sum2 = ex.sumNumber(2, 4);

System.out.println("sum2 = " + sum2);

int sum3 = ex.sumNumber(1, 3, 5);

System.out.println("sum3 = " + sum3);

int diff1 = ex.diffNumber(7, 3, 1, 1);

System.out.println("diff1 = " + diff1);

int diff2 = ex.diffNumber(5, 3 );

System.out.println("diff2 = " + diff2);

}

}

-----------------------------------------------------------------------END---------------------------------------------------------