**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.*

**Solution:**

**package** hsbc.training.day2;

**import** java.util.Scanner;

//Calculator class

**public** **class** Calculator

{

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

{

//Creating object of Calculator class

Calculator C = **new** Calculator();

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

**int** num1,num2;

//Accept two numbers

System.***out***.println("Enter two numbers ==>");

num1 = s.nextInt();

num2 = s.nextInt();

//calling addition function

C.add(num1, num2);

//calling difference function

C.diff(num1, num2);

//calling multiplication function

C.mul(num1, num2);

//calling division function

C.div(num1, num2);

}

//Addition function

**public** **void** add(**int** no1,**int** no2)

{

//adding two numbers and storing the result in res

**int** res = no1+no2;

System.***out***.println("Addition of "+no1+" and "+no2+" is "+res); }

//Multiplication function

**public** **void** mul(**int** no1,**int** no2)

{

//multiplying two numbers and storing the result in res

**int** res = no1\*no2;

System.***out***.println("Multiplication of "+no1+" and "+no2+" is "+res); }

//Difference function

**public** **void** diff(**int** no1,**int** no2)

{

//subtracting two numbers and storing the result in res

**int** res = no1-no2;

System.***out***.println("Difference of "+no1+" and "+no2+" is "+res);

}

//Division function

**public** **void** div(**int** no1,**int** no2)

{

//dividing two numbers and storing the result in res

**int** res = no1/no2;

System.***out***.println("Division of "+no1+" and "+no2+" is "+res);

}

}

**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.*

**Solution:**

**package** hsbc.training.day2;

**public** **class** Sample

{

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

{

//Creating objects of class Sample

Sample s=**new** Sample();

Sample s1=**new** Sample();

//calling the numOfObject method to display the count

*numOfObject*();

//Creating objects of class Sample

Sample s2=**new** Sample();

Sample s3=**new** Sample();

Sample s4=**new** Sample();

//calling the numOfObject method to display the count

*numOfObject*();

//Creating objects of class Sample

**new** Sample();

**new** Sample();

//calling the numOfObject method to display the count

*numOfObject*();

}

//initializing the counter

**public** **static** **int** *count*=0;

//Constructor to count the instantiation of the Class

**public** Sample()

{

//setting counter

*count*++;

}

//displays the number of objects instantiated

**public** **static** **void** numOfObject()

{

System.***out***.println("Number of objects instantiated ==>"+*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*

*In mathematics.*

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.*

**Solution:**

**package** hsbc.training.day2;

**import** java.util.Comparator;

/\*

Class Student with the following details: RollNo, StudName, MarksInEng,MarksInMaths and

MarksInScience. Write getters and setters for the all variables. RollNo should be

automatically generated whenever a new student is added.

\*/

**public** **class** Students **implements** Comparable<Students>

{

**public** **static** **int** *count*=0;

**private** **int** rollNo;

**private** String studName;

**private** **int** marksInEng;

**private** **int** marksInMaths;

**private** **int** marksInScience;

**private** **int** total;

**private** **double** percent;

//automatically generating roll no. whenever a new student object is created

**public** Students()

{

*count*++;

**this**.rollNo = *count*;

**this**.total=**this**.marksInEng+**this**.marksInMaths+**this**.marksInScience;

**this**.percent=(**double**)**this**.total/3;

}

//setter getter methods for all the variables

**public** **int** getRollNo()

{

**return** rollNo;

}

**public** String getStudName()

{

**return** studName;

}

**public** **void** setStudName(String name)

{

studName = name;

}

**public** **int** getMarksInEng()

{

**return** marksInEng;

}

**public** **void** setMarksInEng(**int** eng)

{

marksInEng = eng;

}

**public** **int** getMarksInMaths()

{

**return** marksInMaths;

}

**public** **void** setMarksInMaths(**int** maths)

{

marksInMaths = maths;

}

**public** **int** getMarksInScience()

{

**return** marksInScience;

}

**public** **void** setMarksInScience(**int** sci)

{

marksInScience = sci;

}

**public** **int** getTotal()

{

**return** total;

}

//calculating total marks of all subjects

**public** **void** setTotal()

{

**this**.total=**this**.marksInMaths+**this**.marksInEng+**this**.marksInScience;

}

**public** **double** getPercent()

{

**return** percent;

}

//calculating average marks of the subjects

**public** **void** setPercent()

{

**this**.percent = (**double**) **this**.total/3;

}

//comparing the Students by their RollNo.

@Override

**public** **int** compareTo(Students stud)

{

**if**(rollNo > stud.rollNo)

{

**return** 1;

}

**else** **if**(rollNo < stud.rollNo)

{

**return** -1;

}

**else**

{

**return** 0;

}

}

}

//comparator to compare marks of math

**class** MaxMarksMaths **implements** Comparator<Students>

{

@Override

**public** **int** compare(Students s1, Students s2)

{

**if**(s1.getMarksInMaths() > s2.getMarksInMaths())

{

**return** 1;

}

**else** **if**(s1.getMarksInMaths() < s2.getMarksInMaths())

{

**return** -1;

}

**else**

{

**return** 0;

}

}

}

//comparator to compare percentage

**class** MaxPercentage **implements** Comparator<Students>

{

@Override

**public** **int** compare(Students s1, Students s2)

{

//double s1per=(s1.getMarksInMaths()+s1.getMarksInScience()+s1.getMarksInEng()/3);

//double s2per=(s2.getMarksInMaths()+s2.getMarksInScience()+s2.getMarksInEng()/3);

**if**(s1.getPercent() > s2.getPercent())

{

**return** 1;

}

**else** **if**(s1.getPercent() > s2.getPercent())

{

**return** -1;

}

**else**

{

**return** 0;

}

}

}

//comparator to compare total marks of math & science

**class** MaxMarksMathsAndSci **implements** Comparator<Students>

{

@Override

**public** **int** compare(Students s1, Students s2)

{

**int** s1total=s1.getMarksInMaths()+s1.getMarksInScience();

**int** s2total=s2.getMarksInMaths()+s2.getMarksInScience();

**if**(s1total > s2total)

{

**return** 1;

}

**else** **if**(s1total < s2total)

{

**return** -1;

}

**else**

{

**return** 0;

}

}

}

**package** hsbc.training.day2;

**import** java.util.Arrays;

**import** java.util.Collections;

**import** java.util.List;

**import** java.util.Scanner;

/\*

Class Standard with 8 students’ details and following methods:

1.details() - To display the entire roll no and the name of the students in the

class in the ascending order of roll no.

2.highestPercentage() - To display the roll no and the name of the student

who has got the highest percentage.

3.highestMarksInMaths() - To display the roll no and the name of the student who

scored highest mark In mathematics.

4.highestMarksInMathsAndSci() - To display the roll no and the name of the student

in the ascending order of the total marks in mathematics and science alone.

5.rank() - To display the roll no, name, total marks, percentage and rank of all the

students in the descending order of rank.

\*/

**public** **class** Standard

{

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

{

//created an array of object of Student Class of size 8

Students[] s = **new** Students[8];

Scanner s1=**new** Scanner(System.***in***);

String name=**new** String();

**int** eng,maths,sci;

//loop to accept details of students from user

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

{

s[i]=**new** Students();

System.***out***.println("Enter the details of Student "+i);

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

name=s1.next();

s[i].setStudName(name);

System.***out***.println("English Marks : ");

eng=s1.nextInt();

s[i].setMarksInEng(eng);

System.***out***.println("Maths Marks : ");

maths=s1.nextInt();

s[i].setMarksInMaths(maths);

System.***out***.println("Science Marks : ");

sci=s1.nextInt();

s[i].setMarksInScience(sci);

//setting the total & percent

s[i].setTotal();

s[i].setPercent();

}

s1.close(); //closing Scanner class

*details*(s); //calling details method

*highestPercentage*(s); //calling highestPercentage method

*highestMarksInMaths*(s); //calling highestMarksInMaths method

*highestMarksInMathsAndSci*(s); //calling highestMarksInMathsAndSci method

*rank*(s); //calling rank method

}

/\*details method to display the roll no & name of student in

\*ascending order of roll no

\*/

**public** **static** **void** details(Students[] stud)

{

//sorts the Students by their RollNo.

Arrays.*sort*(stud);

//displaying the details of student in ascending order of roll no

System.***out***.println("Details of Students Sort by Roll No in Ascending order.");

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

{

System.***out***.println("[Roll No.: "+stud[i].getRollNo() +"],[Name : "+stud[i].getStudName()+"],[English Marks : "+stud[i].getMarksInEng()+"],[Maths Marks : "+stud[i].getMarksInMaths()+"],[Science Marks : "+stud[i].getMarksInScience()+"]");

}

}

/\*calling highestMarksInMaths method to display the roll no & name of student

\* who has scored highest marks in math

\*/

**public** **static** **void** highestMarksInMaths(Students[] stud)

{

//converting Students array of objects to list of Students

List<Students> s=Arrays.*asList*(stud);

//finding the max marks scored in maths

Students s1 = Collections.*max*(s,**new** MaxMarksMaths());

//displaying the details of student who has scored highest marks in math

System.***out***.println("Student with highest marks in Maths");

System.***out***.println("[Roll No.: "+s1.getRollNo() +"| Name : "+s1.getStudName()+"]");

}

/\*calling highestPercentage method to display the roll no and the name of the student who has got the highest percentage

\*/

**public** **static** **void** highestPercentage(Students[] stud)

{

//converting Students array of objects to list of Students

List<Students> s=Arrays.*asList*(stud);

//finding the max percentage scored

Students s1 = Collections.*max*(s,**new** MaxPercentage());

//displaying the details of student who has got highest percentage

System.***out***.println("Student with highest percentage");

System.***out***.println("[Roll No.: "+s1.getRollNo() +" | Name : "+s1.getStudName()+"]");

}

/\*calling highestMarksInMathsAndSci method to display the roll no and the name of the student in the ascending order of the total marks in mathematics and science alone

\*/

**public** **static** **void** highestMarksInMathsAndSci(Students[] stud)

{

//converting Students array of objects to list of Students

List<Students> s=Arrays.*asList*(stud);

//sorting the students based on the total marks scored in math and science

Collections.*sort*(s,**new** MaxMarksMathsAndSci());

//displaying the details of student in ascending order of the total marks in math & science alone

System.***out***.println("Student with total marks of Maths & Science Sorted in Ascending order");

**for**(Students s1:s)

{

System.***out***.println("[Roll No.: "+s1.getRollNo() +" | Name : "+s1.getStudName()+"]");

}

}

/\*calling rank method To display the roll no, name, total marks, percentage and rank of all the students in the descending order of rank.

\*/

**public** **static** **void** rank(Students[] stud)

{

//converting Students array of objects to list of Students

List<Students> s=Arrays.*asList*(stud);

//sorting the students on the basis of percentage

Collections.*sort*(s,**new** MaxPercentage());

//displaying the details of student based on lowest to highest rank

System.***out***.println("Student with lowest Rank to highest");

**int** rank=s.size();

**for**(Students s1:s)

{

System.***out***.println("Rank - "+rank+" : [Roll No.: "+s1.getRollNo() +" | Name : "+s1.getStudName()+" | Total :"+s1.getTotal()+" | Percent : "+s1.getPercent()+"]");

rank--;

}

}

}

**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.*

***Solution:***

**package** hsbc.training.day2;

**public** **class** StringOperations

{

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

{

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

//Print the character at the 12th index

System.***out***.println("The character at the 12th index is "+str.charAt(12));

//Checking whether the String contains the word “is”.

System.***out***.println("Whether the String contains the word “is” ? "+str.contains("is"));

//Adding the string “and killed it” to the existing string.

System.***out***.println("Adding the string “and killed it” to the existing string the string is "+str.concat(" and killed it"));

//Checking whether the String ends with the word “dogs”.

System.***out***.println("Whether the String ends with the word “dogs” ? "+str.endsWith("dogs"));

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

System.***out***.println("Whether the given two strings are equal ? "+str.equals("The quick brown Fox jumps over the lazy Dog"));

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

System.***out***.println("Whether the given two strings are equal ? "+str.equals("THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG"));

//Finding the index position of the character “a”.

System.***out***.println("Index position of the character “a” is "+str.indexOf('a'));

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

System.***out***.println("Last index position of the character “e” is "+str.lastIndexOf('e'));

//Finding the length of the String.

System.***out***.println("The length of the String is "+str.length());

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

System.***out***.println("Whether the given two strings matches ? "+str.equalsIgnoreCase("The quick brown Fox jumps over the lazy Dog"));

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

System.***out***.println("Replacing the word “The” with the word “A”."+str.replace("The", "A"));

//initializing variable with the starting index of animals

**int** startIndexOfFox=str.indexOf("fox");

**int** startIndexOfDog=str.indexOf("dog");

//checking for which animal comes first in the string

**if**(startIndexOfFox<startIndexOfDog)

{

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

System.***out***.println("First Part of Split is "+str.substring(0, startIndexOfFox+3));

System.***out***.println("Second Part of Split is "+str.substring(startIndexOfFox+3,str.length()));

}

**else**

{

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

System.***out***.println("First Part of Split is "+str.substring(0, startIndexOfDog+3));

System.***out***.println("Second Part of Split is "+str.substring(startIndexOfDog+3,str.length()));

}

//Printing the animal names alone separately from the given string.

System.***out***.println("Names of the animals are "+str.substring(startIndexOfFox, startIndexOfFox+3)+" , "+str.substring(startIndexOfDog, startIndexOfDog+3));

//Print the above string in completely lower case.

System.***out***.println("String in completely lower case : " +str.toLowerCase());

//Print the above string in completely upper case.

System.***out***.println("String to in completely upper case : " +str.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.*

**Solution:**

**package** hsbc.training.day2;

**public** **class** StringEqualDemo

{

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

{

String str1 = **new** String("Computer");

String str2 = **new** String("Engineering");

String str3 = **new** String("Computer");

String str4="Engineering";

String str5="Data";

String str6="Data";

//Two strings with different value created using new

//comparing them with == operator

**if**(str1 == str2)

{

System.***out***.println("String1 == String2");

}

**else**

{

System.***out***.println("String1 != String2");

}

//Two strings with same value created using new

//comparing them with == operator

**if**(str1 == str3)

{

System.***out***.println("String1 == String3");

}

**else**

{

System.***out***.println("String1 != String3");

}

//Two strings with same value one created using new and other without new

//comparing them with == operator

**if**(str4 == str2)

{

System.***out***.println("String4 == String2");

}

**else**

{

System.***out***.println("String4 != String2");

}

//Two strings with same value created without using new

//comparing them with == operator

**if**(str5 == str6)

{

System.***out***.println("String5 == String6");

}

**else**

{

System.***out***.println("String5 != String6");

}

//Two strings with different value created without using new

//comparing them with == operator

**if**(str5 == str4)

{

System.***out***.println("String4 == String5");

}

**else**

{

System.***out***.println("String4 != String5");

}

//Two strings with different value created using new

//comparing them with contentEquals operator

**if**(str1.contentEquals(str2))

{

System.***out***.println("String1 equals String2");

}

**else**

{

System.***out***.println("String1 !equals String2");

}

//Two strings with same value created using new

//comparing them with contentEquals operator

**if**(str1.contentEquals(str3))

{

System.***out***.println("String1 equals String3");

}

**else**

{

System.***out***.println("String1 != String3");

}

//Two strings with same value one created using new and other without new

//comparing them with contentEquals operator

**if**(str4.contentEquals(str2))

{

System.***out***.println("String4 equals String2");

}

**else**

{

System.***out***.println("String4 !equals String2");

}

//Two strings with same value created without using new

//comparing them with contentEquals operator

**if**(str5.contentEquals(str6))

{

System.***out***.println("String5 equals String6");

}

**else**

{

System.***out***.println("String5 !equals String6");

}

//Two strings with different value created without using new

//comparing them with contentEquals operator

**if**(str5.contentEquals(str4))

{

System.***out***.println("String4 equals String5");

}

**else**

{

System.***out***.println("String4 !equals String5");

}

}

}

**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.*

**Solution:**

**package** hsbc.training.day2;

**public** **class** ArrayCopy

{

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

{

//declaring and initializing the integer array of size 8

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

//declaring the array to store the copy of first array

**int**[] arrCopy=**new** **int**[arr.length];

//copying the elements of one array to another

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

{

arrCopy[i]=arr[i];

}

//displaying both the arrays

System.***out***.println("Original Array | Duplicate Array");

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

{

System.***out***.println(arr[i]+" | "+arrCopy[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.*

**Solution:**

**package** hsbc.training.day2;

**public** **class** ArrayOperations

{

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

{

//declaring and initializing the integer array of size 8

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

**int** sum=0,avg;

//calculating the sum of all array elements

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

{

sum=sum+arr[i];

}

//displaying sum the all array elements

System.***out***.println("Sum of Array Elemnts ==>"+sum);

//calculating the average of the array elements

avg=sum/arr.length;

//displaying sum the all array elements

System.***out***.println("Average of Array Elemnts ==>"+avg);

}

}

**Lab Exercise No:**21

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

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

**Solution:**

**package** hsbc.training.day2;

**public** **class** MatrixSum

{

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

{

**int** matrix1[][] = **new** **int**[][] {{2,3,8},

{1,-2,3},

{15,4,-6},

{7,6,0}};

**int** matrix2[][] = **new** **int**[][] { {1,4,9},

{8,4,11},

{-8,7,5},

{3,4,2}};

//displaying the matrix1

System.***out***.println("Matrix 1 ==>");

*displayMatrix*(matrix1);

//displaying the matrix2

System.***out***.println("Matrix 2 ==>");

*displayMatrix*(matrix2);

//calling the matrixSum method

System.***out***.println("Sum of the two matrices ==> ");

*matrixSum*(matrix1, matrix2);

}

//creating displayMatrix method to display the matrix

**public** **static** **void** displayMatrix(**int** mat[][])

{

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

{

**for**(**int** j=0;j<mat[i].length;j++)

{

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

}

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

}

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

}

//calculating the sum of matrices

**public** **static** **void** matrixSum(**int** mat1[][], **int** mat2[][])

{

**int** matSum[][] = **new** **int**[mat1.length][mat1[0].length];

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

{

**for**(**int** j=0;j<matSum[i].length;j++)

{

matSum[i][j] = mat1[i][j] + mat2[i][j];

}

}

*displayMatrix*(matSum);

}

}

**Lab Exercise No:**22

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

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

**Solution:**

**package** hsbc.training.day2;

**public** **class** TwoDArraySquare

{

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

{

**int** arr[][] = **new** **int**[][] { {2,3,8},

{1,-2,3},

{7,6,0}};

//displaying the 2D array

System.***out***.println(" 2D Array ==>");

*display2dArray*(arr);

//calling the arrSquare method

System.***out***.println("Square of 2D array ==> ");

*arrSquare*(arr);

}

//creating display2dArray method to display the 2D array

**public** **static** **void** display2dArray(**int** a[][])

{

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

{

**for**(**int** j=0;j<a[i].length;j++)

{

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

}

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

}

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

}

//squaring the 2D array

**public** **static** **void** arrSquare(**int** a[][])

{

**int** squareArr[][] = **new** **int**[a.length][a.length];

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

{

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

{

squareArr[i][j]=(**int**) Math.*pow*(a[i][j], 2);

}

}

*display2dArray*(squareArr);

}

}

**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.*

**Solution:**

**package** hsbc.training.day2;

**import** java.util.HashMap;

**import** java.util.Map;

**import** java.util.Map.Entry;

**public** **class** ArrayElementFrequency

{

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

{

**int** arr[]= {1,2,3,3,2,4,5,1,1,4};

//creating a object of Key and Value as Integer

Map<Integer,Integer> obj = **new** HashMap<>();

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

{

//check whether the element is already present or not

//if not put key as 1

**if**(!obj.containsKey(arr[i]))

{

obj.put(arr[i], 1);

}

**else**

{

//increment the frequency if element is already present

**int** freq = obj.get(arr[i]);

freq++;

obj.put(arr[i],freq);

}

}

//displaying the array elements and their occurance frequency

**for**(Entry<Integer,Integer> a : obj.entrySet())

{

System.***out***.println("Array Element : " + a.getKey() +"| Occurance Frequency : " + a.getValue());

}

}

}

**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.*

**Solution:**

**package** hsbc.training.day2;

**public** **class** Shapes

{

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

{

Calculate square = **new** Calculate();

//displaying area and perimeter of square

System.***out***.println("The area of Square ==> " + square.area(4));

System.***out***.println("The perimeter of Square ==> " + square.perimeter(9));

Calculate rectangle = **new** Calculate();

//displaying area and perimeter of square

System.***out***.println("The area of Reactangle ==> " + rectangle.area(5,6));

System.***out***.println("The perimeter of Rectangle ==>" + rectangle.perimeter(5,7));

}

}

**class** Calculate

{

//calculating area of square

**public** **int** area(**int** length)

{

**return** length\*length;

}

//calculating perimeter of square

**public** **int** perimeter(**int** length)

{

**return** length\*4;

}

//overloading the area method

//calculating area of rectangle

**public** **int** area(**int** length, **int** breadth)

{

**return** length\*breadth;

}

//overloading the area perimeter

//calculating perimeter rectangle

**public** **int** perimeter(**int** length, **int** breadth)

{

**return** 2\*(length+breadth);

}

}

**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*

**Solution:**

**package** hsbc.training.day2;

**public** **class** Employee

{

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

{

//created EmployeeDetails array of object

EmployeeDetails[] emp = **new** EmployeeDetails[9];

//initializing the objects using constructor with different parameters

emp[0] = **new** EmployeeDetails("Rasi");

emp[1] = **new** EmployeeDetails("Nivea",123);

emp[2] = **new** EmployeeDetails("Damon",74,56);

emp[3] = **new** EmployeeDetails("Ron",122, "Risk IT");

emp[4] = **new** EmployeeDetails("Rachel", 124, 35, 13);

emp[5] = **new** EmployeeDetails("Joey", 118, 43, "HR");

emp[6] = **new** EmployeeDetails("Pam",56 , "Manager", 2);

emp[7] = **new** EmployeeDetails("Mudita", 127, "Developer", "Pune", 9);

emp[8] = **new** EmployeeDetails("Shivani", 11, 25, "Wholesale IT", "Kolkata", 11);

//displaying Employee Details

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

{

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

}

}

}

**class** EmployeeDetails

{

**private** String empName;

**private** **int** empId;

**private** **int** empAge;

**private** String empDesgn;

**private** String empLocation;

**private** **int** empExpInYrs;

//constructor overloading

//constructor with Employee name alone

**public** EmployeeDetails(String empName)

{

**this**.empName = empName;

}

//constructor with employee name, id

**public** EmployeeDetails(String empName, **int** empId)

{

**this**(empName);

**this**.empId = empId;

}

//constructor with Employee name, id and age

**public** EmployeeDetails(String empName, **int** empId, **int** empAge)

{

**this**(empName, empId);

**this**.empAge = empAge;

}

//constructor with Employee name, id and designation

**public** EmployeeDetails(String empName, **int** empId, String empDesgn)

{

**this**(empName, empId);

**this**.empDesgn = empDesgn;

}

//constructor with Employee name, id, age and designation

**public** EmployeeDetails(String empName, **int** empId, **int** empAge, String empDesgn)

{

**this**(empName, empId, empDesgn);

**this**.empAge = empAge;

}

//constructor with Employee name, id, age and experience

**public** EmployeeDetails(String empName, **int** empId, **int** empAge, **int** empExpInYrs)

{

**this**(empName, empId, empAge);

**this**.empExpInYrs = empExpInYrs;

}

//constructor with Employee name, id, designation and experience

**public** EmployeeDetails(String empName, **int** empId, String empDesgn, **int** empExpInYrs)

{

**this**(empName, empId, empDesgn);

**this**.empExpInYrs = empExpInYrs;

}

//constructor with Employee name, id, designation, location and experience

**public** EmployeeDetails(String empName, **int** empId, String empDesgn, String empLocation, **int** empExpInYrs)

{

**this**(empName, empId, empDesgn, empExpInYrs);

**this**.empLocation = empLocation;

}

//constructor with Employee name, id, age, designation, location and experience

**public** EmployeeDetails(String empName, **int** empId, **int** empAge, String empDesgn, String empLocation, **int** empExpInYrs)

{

**this**(empName, empId, empDesgn, empExpInYrs);

**this**.empAge = empAge;

**this**.empLocation = empLocation;

}

@Override

**public** String toString()

{

**return** "Employee Details: [ Name : " + empName + " | Id : " + empId + " | Age : " + empAge + ", Designation : " + empDesgn+ " | Location=" + empLocation + ", | Experience=" + empExpInYrs + " ]";

}

}

**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.*

**Solution:**

**package** hsbc.training.day2;

**public** **class** CalcOverloading

{

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

{

//Creating a new object of Calculator class

Calc C = **new** Calc();

//calling addition function with both int as parameter

C.add(12, 1);

//calling difference function with both int as parameter

C.diff(11, 4);

//calling multiplication function with both int as parameter

C.mul(4, 1);

//calling division function with both int as parameter

C.div(15, 5);

//calling addition function with both double as parameter

C.add(14.42, 14.3);

//calling difference function with both double as parameter

C.diff(11.1, 5.4);

//calling multiplication function with both double as parameter

C.mul(4.2, 1.1);

//calling division function with both double as parameter

C.div(11.5, 5.5);

//calling addition function with double and int as parameter

C.add(12.6, 1);

//calling difference function with double and int as parameter

C.diff(11.91, 4);

//calling multiplication function with double and int as parameter

C.mul(4.56, 1);

//calling division function with double and int as parameter

C.div(15.5, 5);

//calling addition function with int and double as parameter

C.add(12, 19.7);

//calling difference function with int and double as parameter

C.diff(11, 5.5);

//calling multiplication function with int and double as parameter

C.mul(4, 7.1);

//calling division function with int and double as parameter

C.div(15, 4.05);

}

}

//Calc class

**class** Calc

{

//method overloading

//Addition function with both int as parameter

**public** **void** add(**int** no1,**int** no2)

{

//adding two numbers and storing the result in res

**int** res = no1+no2;

System.***out***.println("Addition of "+no1+" and "+no2+" is "+res);

}

//Multiplication function with both int as parameter

**public** **void** mul(**int** no1,**int** no2)

{

//multiplying two numbers and storing the result in res

**int** res = no1\*no2;

System.***out***.println("Multiplication of "+no1+" and "+no2+" is "+res);

}

//Difference function with both int as parameter

**public** **void** diff(**int** no1,**int** no2)

{

//subtracting two numbers and storing the result in res

**int** res = no1-no2;

System.***out***.println("Difference of "+no1+" and "+no2+" is "+res);

}

//Division function with both int as parameter

**public** **void** div(**int** no1,**int** no2)

{

//dividing two numbers and storing the result in res

**int** res = no1/no2;

System.***out***.println("Division of "+no1+" and "+no2+" is "+res);

}

//Addition function with first double and second int as parameter

**public** **void** add(**double** no1,**int** no2)

{

//adding two numbers and storing the result in res

**double** res = no1+no2;

System.***out***.println("Addition of "+no1+" and "+no2+" is "+res);

}

//Multiplication function with first double and second int as parameter

**public** **void** mul(**double** no1,**int** no2)

{

//multiplying two numbers and storing the result in res

**double** res = no1\*no2;

System.***out***.println("Multiplication of "+no1+" and "+no2+" is "+res);

}

//Difference function with first double and second int as parameter

**public** **void** diff(**double** no1,**int** no2)

{

//subtracting two numbers and storing the result in res

**double** res = no1-no2;

System.***out***.println("Difference of "+no1+" and "+no2+" is "+res);

}

//Division function with first double and second int as parameter

**public** **void** div(**double** no1,**int** no2)

{

//dividing two numbers and storing the result in res

**double** res = no1/no2;

System.***out***.println("Division of "+no1+" and "+no2+" is "+res);

}

//Addition functionwith both double as parameter

**public** **void** add(**double** no1,**double** no2)

{

//adding two numbers and storing the result in res

**double** res = no1+no2;

System.***out***.println("Addition of "+no1+" and "+no2+" is "+res);

}

//Multiplication function with both double as parameter

**public** **void** mul(**double** no1,**double** no2)

{

//multiplying two numbers and storing the result in res

**double** res = no1\*no2;

System.***out***.println("Multiplication of "+no1+" and "+no2+" is "+res);

}

//Difference function with both double as parameter

**public** **void** diff(**double** no1,**double** no2)

{

//subtracting two numbers and storing the result in res

**double** res = no1-no2;

System.***out***.println("Difference of "+no1+" and "+no2+" is "+res);

}

//Division function with both double as parameter

**public** **void** div(**double** no1,**double** no2)

{

//dividing two numbers and storing the result in res

**double** res = no1/no2;

System.***out***.println("Division of "+no1+" and "+no2+" is "+res);

}

//Addition function with first int and second double as parameter

**public** **void** add(**int** no1,**double** no2)

{

//adding two numbers and storing the result in res

**double** res = no1+no2;

System.***out***.println("Addition of "+no1+" and "+no2+" is "+res);

}

//Multiplication function with first int and second double as parameter

**public** **void** mul(**int** no1,**double** no2)

{

//multiplying two numbers and storing the result in res

**double** res = no1\*no2;

System.***out***.println("Multiplication of "+no1+" and "+no2+" is "+res);

}

//Difference function with first int and second double as parameter

**public** **void** diff(**int** no1,**double** no2)

{

//subtracting two numbers and storing the result in res

**double** res = no1-no2;

System.***out***.println("Difference of "+no1+" and "+no2+" is "+res);

}

//Division function with first int and second double as parameter

**public** **void** div(**int** no1,**double** no2)

{

//dividing two numbers and storing the result in res

**double** res = no1/no2;

System.***out***.println("Division of "+no1+" and "+no2+" is "+res);

}

}

**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.*

**Solution:**

**package** hsbc.training.day2;

**public** **class** ClassLoading

{

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

{

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

//calling to the class Loading

Class.*forName*("hsbc.training.day2.Loading");

}

**static**

{

System.***out***.println("Class is Loading !!");

}

}

**class** Loading

{

///creating the object as static

**public** **static** Loading *load*;

**static**

{

System.***out***.println("Class Loaded !!");

*load*=**new** Loading();

}

}

**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.*

**Solution:**

**package** hsbc.training.day2;

**import** java.util.Scanner;

**public** **class** ClaculatorVarArgs {

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

{

//Creating a new object of Calculator class

ClaculatorVarArgs C = **new** ClaculatorVarArgs();

//calling addition function

C.add(1,2,3,4);

//calling difference function

C.diff(9,5,3);

//calling multiplication function

C.mul(5,2);

//calling division function

C.div(4, 2);

}

//Addition function using variable arguments

**public** **void** add(**int** ...no)

{

//adding numbers and storing the result in res

**int** res=0;

**for**(**int** i:no)

{

res=res+i;

}

System.***out***.println("Addition ==> "+res);

}

//Multiplication function

**public** **void** mul(**int** no1,**int** no2)

{

//multiplying two numbers and storing the result in res

**int** res = no1\*no2;

System.***out***.println("Multiplication ==> "+res);

}

//Difference function using variable arguments

**public** **void** diff(**int** ...no)

{

//subtracting numbers and storing the result in res

**int** res=no[0];

**if**(no.length > 1)

{

**for**(**int** i = 1; i < no.length; i++)

{

res=res-no[i];

}

}

System.***out***.println("Difference ==> "+res);

}

//Division function

**public** **void** div(**int** no1,**int** no2)

{

//dividing two numbers and storing the result in res

**int** res = no1/no2;

System.***out***.println("Division ==> "+res);

}

}