**JAVA**

**Classes and Methods**

**Day 2 Solution**

**Ankit Rupapara**

**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 Code: //** Takes two inputs and does all calculation

**package** pack.day2.com;

**public** **class** Calculator

{

**public** **void** add(**int** num1,**int** num2) //addition of 2 number

{

**int** num3 = num1 + num2;

System.***out***.println("Addition of 2 numbers is "+ num3);

}

**public** **void** sub(**int** num1,**int** num2) //subtraction of 2 number

{

**int** num3 = num1 - num2;

System.***out***.println("Subtraction of 2 numbers is "+ num3);

}

**public** **void** mul(**int** num1,**int** num2) //multiplication of 2 number

{

**int** num3 = num1 \* num2;

System.***out***.println("Multiplication of 2 numbers is "+ num3);

}

**public** **void** div(**int** num1,**int** num2) //division of 2 number

{

**int** num3 = num1 / num2;

System.***out***.println("Division of 2 numbers is "+ num3);

}

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

{

Calculator calc=**new** Calculator(); //Object Created

calc.add(5,12);

calc.sub(10,5);

calc.mul(2,13);

calc.div(75,15);

}

}

**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 Code:**

**package** pack.day2.com;

**public** **class** Sample {

**static** **int** *counter* = 0; //initial count is 0 and will be incremented as object gets created

**public** Sample() {

*counter*++;

}

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

// **TODO** Auto-generated method stub

Sample s1 = **new** Sample(); // object creation

Sample s2 = **new** Sample(); // object creation

Sample s3 = **new** Sample(); // object creation

Sample s4 = **new** Sample(); // object creation

Sample s5 = **new** Sample(); // object creation

Sample s6 = **new** Sample(); // object creation

Sample s7 = **new** Sample(); // object creation

System.***out***.println("Total object created : "+ *counter*);

}

}

**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.*

**Solution Code:**

**package** pack.day2.com;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.List;

**public** **class** Solution16 {

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

Student1 s1=**new** Student1("Prince",101,23);

Student1 s2=**new** Student1("Vikram",102,21);

Student1 s3=**new** Student1("Arun",103,24);

Student1 s4=**new** Student1("Tarun",104,21);

Student ss1=**new** Student("Prince",101,23);

Student ss2=**new** Student("Vikram",102,21);

Student ss3=**new** Student("Arun",103,24);

Student ss4=**new** Student("Tarun",104,21);

Student st[]=**new** Student[] {

ss1,ss2,ss3,ss4

};

List<Student1> p=**new** ArrayList<Student1>();

p.add(s1);

p.add(s2);

p.add(s3);

p.add(s4);

MyComparator comp=**new** MyComparator();

Arrays.*sort*(st);

Collections.*sort*(p, comp);

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

**for**(Student1 objs:p) {

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

}

}

**static** {

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

}

}

**class** MyComparator **implements** Comparator<Student1>{

@Override

**public** **int** compare(Student1 o1, Student1 o2) {

// **TODO** Auto-generated method stub

**if**(o1.age>o2.age)

**return** 1;

**else** **if** (o1.age<o2.age)

**return** -1;

**else**

**return** 0;

}

}

**class** Student1 {

**public** Student1() {

**super**();

}

**public** Student1(String name, **int** rollNo, **int** age) {

**super**();

**this**.name = name;

**this**.rollNo = rollNo;

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

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

**return** rollNo;

}

**public** **void** setRollNo(**int** rollNo) {

**this**.rollNo = rollNo;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

@Override

**public** String toString() {

**return** "Student1 [name=" + name + ", rollNo=" + rollNo + ", age=" + age + "]";

}

String name;

**int** rollNo;

**int** age;

}

**class** Student **implements** Comparable<Student>{

**public** Student() {

**super**();

}

**public** Student(String name, **int** rollNo, **int** age) {

**super**();

**this**.name = name;

**this**.rollNo = rollNo;

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

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

**return** rollNo;

}

**public** **void** setRollNo(**int** rollNo) {

**this**.rollNo = rollNo;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

@Override

**public** String toString() {

**return** "Student [name=" + name + ", rollNo=" + rollNo + ", age=" + age + "]";

}

String name;

**int** rollNo;

**int** age;

@Override

**public** **int** compareTo(Student o) {

// **TODO** Auto-generated method stub

**if**(**this**.age>o.age)

**return** -1;

**else** **if** (**this**.age<o.age)

**return** 1;

**else**

**return** 0;

}

}

**class** Computer{

**public** **static** Computer *computer*;

**static** {

System.***out***.println("Computer class loaded");

*computer*=**new** Computer();

//System.out.println(computer);

}

}

**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 Code:**

**package** pack.day2.com;

**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)); // 1.Print the character at the 12th index.

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 Code:**

**package** pack.day2.com;

**public** **class** Solution18 {

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

// **TODO** Auto-generated method stub

String s1 = **new** String("String");

String s2 = **new** String("String");

**if**(s1 == s2) //since two string was generated via new it has different address

System.***out***.println("Equal for ==");

**else**

System.***out***.println("Not Equal for ==");

**if**(s1.equals(s2)) // it checks only strings

System.***out***.println("Equal for .equals()");

**else**

System.***out***.println("Not Equal for .equals()");

}

}

**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 Code:**

**package** pack.day2.com;

**public** **class** Solution19 {

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

// **TODO** Auto-generated method stub

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

**int**[] newArray = numbers.clone(); //create a copy of array. change in one won't reflect in other due to .clone()

**for**(**int** num : newArray) {

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

}

}

}

**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 Code:**

**package** pack.day2.com;

**public** **class** Solution20 {

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

// **TODO** Auto-generated method stub

**double**[] numbers = {10, 20.5, 30, 40.15};

**double** sum = 0;

**for**(**double** num : numbers)

{

sum += num;

}

System.***out***.println("Sum of all numbers is " + sum);

System.***out***.println("Average of all numbers is "+ sum/numbers.length); // sum / total numbers will give average

}

}

**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 Code:**

**package** pack.day2.com;

**import** java.util.Scanner;

**public** **class** Solution21

{

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

**int** row, column;

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

System.***out***.println("Enter the no. of rows and cols of matrix");

row = scnr.nextInt();

column = scnr.nextInt();

**int** first[][] = **new** **int**[row][column]; //matrix

**int** second[][] = **new** **int**[row][column]; //matrix

System.***out***.println("Enter the elements of first matrix"); //values of fields in matrix

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

{

**for** (**int** j = 0; j < column; j++)

{

first[i][j] = scnr.nextInt();

}

}

System.***out***.println("Enter the elements of second matrix"); //values of fields in matrix

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

{

**for** (**int** j = 0; j < column; j++)

{

second[i][j] = scnr.nextInt();

}

}

System.***out***.println("Sum of the 2 matrix:");

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

{

**for** (**int** j = 0; j < column; j++)

{

System.***out***.println((first[i][j] + second[i][j]) + " ");

}

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

}

}

}

**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 Code:**

**package** pack.day2.com;

**import** java.util.Scanner;

**public** **class** Solution22 {

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

// **TODO** Auto-generated method stub

**int** row, column;

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

System.***out***.println("Enter the no. of rows and cols of matrix");

row = scnr.nextInt();

column = scnr.nextInt();

**int** first[][] = **new** **int**[row][column]; //matrix

System.***out***.println("Enter the elements of first matrix"); //values of fields in matrix

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

{

**for** (**int** j = 0; j < column; j++)

{

first[i][j] = scnr.nextInt();

}

}

System.***out***.println("Square of each element in the matrix:");

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

{

**for** (**int** j = 0; j < column; j++)

{

System.***out***.println((first[i][j]\* first[i][j]) + " "); //multiplication of itself will give square

}

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

}

}

}

**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 Code:**

**package** pack.day2.com;

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

**public** **class** Solution23

{

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

{

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

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

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

**int** size=10, i, j, count;

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

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

{

arr[i] = scnr.nextInt(); //Array values

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.*

**Recommended duration:***20Mins*

**Solution Code:**

**package** pack.day2.com;

**public** **class** Solution24 {

**public** **static** **void** area(**int** height) // for square - area

{

**int** area=height\*height;

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

}

**public** **static** **void** area(**int** height,**int** length) // for rectangle - area

{

**int** area=length\*height;

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

}

**public** **static** **void** perimeter(**int** height) // for square - perimeter

{

**int** perimeter=4\*height;

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

}

**public** **static** **void** perimeter(**int** height,**int** length) // for rectangle - perimeter

{

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

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

}

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

// **TODO** Auto-generated method stub

**int** val1 = 7, val2 = 12;

*area*(val1);

*area*(val1,val2);

*perimeter*(val1);

*perimeter*(val1,val2);

}

}

**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 Code:**

**package** pack.day2.com;

**public** **class** Employee {

**private** String empName;

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

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

**private** String empDesgn;

**private** String empLocation;

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

**public** Employee(String empName) {

**this**.empName = empName;

}

**public** Employee(String empName, **int** empId) {

**this**.empName = empName;

**this**.empId = empId;

}

**public** Employee(String empName, **int** empId, **int** empAge) {

**this**.empName = empName;

**this**.empId = empId;

**this**.empAge = empAge;

}

**public** Employee(String empName, **int** empId, String empDesgn) {

**this**.empName = empName;

**this**.empId = empId;

**this**.empDesgn = empDesgn;

}

**public** Employee(String empName, **int** empId, **int** empAge, String empDesgn) {

**this**.empName = empName;

**this**.empId = empId;

**this**.empAge = empAge;

**this**.empDesgn = empDesgn;

}

**public** Employee(String empName, **int** empId, **int** empAge, **int** empExpInYrs) {

**this**.empName = empName;

**this**.empId = empId;

**this**.empAge = empAge;

**this**.empExpInYrs = empExpInYrs;

}

**public** Employee(String empName, **int** empId, String empDesgn, **int** empExpInYrs) {

**this**.empName = empName;

**this**.empId = empId;

**this**.empDesgn = empDesgn;

**this**.empExpInYrs = empExpInYrs;

}

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

**this**.empName = empName;

**this**.empId = empId;

**this**.empDesgn = empDesgn;

**this**.empLocation = empLocation;

**this**.empExpInYrs = empExpInYrs;

}

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

**this**.empName = empName;

**this**.empId = empId;

**this**.empAge = empAge;

**this**.empDesgn = empDesgn;

**this**.empLocation = empLocation;

**this**.empExpInYrs = empExpInYrs;

}

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

// **TODO** Auto-generated method stub

Employee e1 = **new** Employee("Ankit");

Employee e2 = **new** Employee("Ankit", 13);

Employee e3 = **new** Employee("Ankit", 13, 21);

Employee e4 = **new** Employee("Ankit", 13, "TSE");

Employee e5 = **new** Employee("Ankit", 13, 21, "TSE");

Employee e6 = **new** Employee("Ankit", 13, 21, 1);

Employee e7 = **new** Employee("Ankit", 13, "TSE", 1);

Employee e8 = **new** Employee("Ankit", 13, "TSE", "Pune", 1);

Employee e9 = **new** Employee("Ankit", 13, 21, "TSE", "Pune", 1);

}

}

**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 Code:**

**package** pack.day2.com;

**public** **class** Calculator2

{

**public** **void** add(**int** num1,**int** num2) //addition of 2 number

{

**int** num3=num1+num2;

System.***out***.println("Addition of 2 numbers is "+ num3);

}

**public** **void** add(**double** num1,**double** num2) //addition of 2 number

{

**double** num3=num1+num2;

System.***out***.println("Addition of 2 numbers is "+ num3);

}

**public** **void** add(**int** num1,**double** num2) //addition of 2 number

{

**double** num3=num1+num2;

System.***out***.println("Addition of 2 numbers is "+ num3);

}

**public** **void** add(**double** num1,**int** num2) //addition of 2 number

{

**double** num3=num1+num2;

System.***out***.println("Addition of 2 numbers is "+ num3);

}

**public** **void** sub(**int** num1,**int** num2) //subtraction of 2 number

{

**int** num3=num1-num2;

System.***out***.println("Subtraction of 2 numbers is "+ num3);

}

**public** **void** sub(**double** num1,**double** num2) //subtraction of 2 number

{

**double** num3=num1-num2;

System.***out***.println("Subtraction of 2 numbers is "+ num3);

}

**public** **void** sub(**int** num1,**double** num2) //subtraction of 2 number

{

**double** num3=num1-num2;

System.***out***.println("Subtraction of 2 numbers is "+ num3);

}

**public** **void** sub(**double** num1,**int** num2) //subtraction of 2 number

{

**double** num3=num1-num2;

System.***out***.println("Subtraction of 2 numbers is "+ num3);

}

**public** **void** mul(**int** num1,**int** num2) //multiplication of 2 number

{

**int** num3=num1\*num2;

System.***out***.println("Multiplication of 2 numbers is "+ num3);

}

**public** **void** mul(**double** num1,**double** num2) //multiplication of 2 number

{

**double** num3=num1\*num2;

System.***out***.println("Multiplication of 2 numbers is "+ num3);

}

**public** **void** mul(**int** num1,**double** num2) //multiplication of 2 number

{

**double** num3=num1\*num2;

System.***out***.println("Multiplication of 2 numbers is "+ num3);

}

**public** **void** mul(**double** num1,**int** num2) //multiplication of 2 number

{

**double** num3=num1\*num2;

System.***out***.println("Multiplication of 2 numbers is "+ num3);

}

**public** **void** div(**int** num1,**int** num2) //division of 2 number

{

**int** num3=num1/num2;

System.***out***.println("Division of 2 numbers is "+ num3);

}

**public** **void** div(**double** num1,**double** num2) //division of 2 number

{

**double** num3=num1/num2;

System.***out***.println("Division of 2 numbers is "+ num3);

}

**public** **void** div(**int** num1,**double** num2) //division of 2 number

{

**double** num3=num1/num2;

System.***out***.println("Division of 2 numbers is "+ num3);

}

**public** **void** div(**double** num1,**int** num2) //division of 2 number

{

**double** num3=num1/num2;

System.***out***.println("Division of 2 numbers is "+ num3);

}

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

{

Calculator2 calc=**new** Calculator2(); //Object Created

calc.add(5,12);

calc.add(5.1,12.9);

calc.add(5,12.9);

calc.add(5.1,12);

calc.sub(10,5);

calc.sub(10.2,5.3);

calc.sub(10,5.3);

calc.sub(10.2,5);

calc.mul(2,13);

calc.mul(2.2,13.3);

calc.mul(2,13.3);

calc.mul(2.2,13);

calc.div(75,15);

calc.div(75.5,15.5);

calc.div(75,15.5);

calc.div(75.5,15);

}

}

**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 Code:**

**package** pack.day2.com;

**public** **class** Solution27 {

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

{

System.***out***.println("Hey there !!");

Class.*forName*("pack.day2.com.MyClass");

}

**static** // Static block runs first, even before main method

{

System.***out***.println("Class is loading..");

}

}

**class** MyClass

{

**public** **static** MyClass *myClass*;

**static**

{

System.***out***.println("My class loaded");

*myClass*=**new** MyClass(); // Object created before class loads

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

}

}

**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 Code:**

**package** pack.day2.com;

**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);

}

// ...num will make this function with unlimited arguments.

**public** **static** **void** add(**int** ...num) //addition of 2 number

{

**int** sum = 0;

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

{

sum += num[i];

}

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

}

**public** **static** **void** sub(**int** ...num) //subtraction of number

{

**int** sub = num[0]; //first input to be taken as positive and rest numbers will be deducted from that.

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

{

sub -= num[i];

}

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

}

}