## LAB # 5

## Mirza Muhammad Abbas

## 21SW101

Question # 1:

Create a class having 4 functions, add, sub, mul and div. Each function accepts 2 parameters and returns the sum, difference, multiplication and division of these numbers.

Create a main class having main function that uses the above class.

Solution:

import java.util.Scanner;

class calculator {

public float add ( float n1, float n2 ) {

return ( n1 + n2 ) ;

}

public float sub ( float n1, float n2 ) {

return ( n1 - n2 ) ;

}

public float mlt ( float n1, float n2 ) {

return ( n1 \* n2 ) ;

}

public float div ( float n1, float n2 ) {

return ( n1 / n2 ) ;

}

}

class Q1 {

public static void main ( String[] args ) {

Scanner S1=new Scanner(System.in);

char op;

calculator c1=new calculator();

System.out.println("Enter First Number: ");

float f1=S1.nextFloat();

System.out.println("Enter Second Number: ");

float f2=S1.nextFloat();

System.out.println("Enter Operator: ");

char ch=S1.next().charAt(0);

switch (ch){

case '+':

System.out.println("Result: " + c1.add(f1,f2));

break;

case '-':

System.out.println("Result: " + c1.sub(f1,f2));

break;

case '\*':

System.out.println("Result: " + c1.mlt(f1,f2));

break;

case '/':

System.out.println("Result: " + c1.div(f1,f2));

break;

default:

System.out.println("Invalid Opeator");

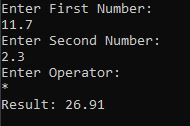
break;

}

}

}

Output:



Question # 2:

Demonstrate the use of the following:

Constructor, Accessors and mutators, static class members.

Solution:

class example{

String ObjName;

static int ObjNo=0;

example(){ //constructor

ObjNo++;

}

example(String n){ //constructor

ObjName=n;

ObjNo++;

}

void setName(String n){ //Mutator

ObjName=n;

}

String getName(){ //Accessor

return ObjName;

}

int getNo(){ //Accessor

return ObjNo;

}

}

class Q2{

public static void main(String[] args){

example ex1=new example();

ex1.setName("TV");

System.out.println("Object No: " + ex1.getNo());

System.out.println("Object Name: " + ex1.getName());

example ex2=new example("Car");

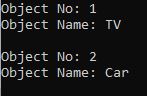
System.out.println("\nObject No: " + ex2.getNo());

System.out.println("Object Name: " + ex2.getName());

}

}

Output:



Question # 3:

Create a Student class that stores student data, provides methods for initializing and displaying student data. Also provide a parameterized constructor for initializing student class data members.

Solution:

class student{

String Name;

String RollNo;

String Dept;

student(String n, String rn, String d){ //Constructor

Name=n;

RollNo=rn;

Dept=d;

}

void set(String n, String rn, String d){

Name=n;

RollNo=rn;

Dept=d;

}

String get(){

return(Name+"\n"+RollNo+"\n"+Dept);

}

}

class Q3{

public static void main(String[] args){

student s1=new student("Kevin", "001", "SW");

System.out.println(s1.get());

}

}

Output: