

Lab Program:01

AIM: Write a JAVA program to display default value of all primitive data types of JAVA.

PROGRAM:

```
public class LabTask1
{
    static byte b=100;
    static short s=200;
    static int i=300;
    static long l=400;
    static float f=500;
    static double d=600;
    static boolean bl=true;
    static char c='A';

    public static void main(String args[])
    {
        System.out.println("Byte DataType::"+b);
        System.out.println("Short DataType::"+s);
        System.out.println("Integer DataType::"+i);
        System.out.println("Long DataType ::"+l);
        System.out.println("Float DataType::"+f);
        System.out.println("Double DataType::"+d);
        System.out.println("Character DataType::"+c);
        System.out.println("Boolean DataType::"+bl);

    }
}
```

Output:

```
Byte DataType::100
Short DataType::200
Integer DataType::300
Long DataType ::400
Float DataType::500.0
Double DataType::600.0
Character DataType::A
Boolean DataType::true
```

Lab Program: 02

AIM: Write a JAVA program that displays the roots of a quadratic equation $ax^2+bx+c=0$. Calculate the discriminant D and basing on the value of D , describe the nature of roots.

PROGRAM:

```
import java.util.Scanner;
public class QuadraticEquation{
public static void main(String args[]){
int a,b,c;
double root1,root2,d;
Scanner S=new Scanner(System.in);
System.out.println("Given quadratic equation:  $ax^2+bx+c$ ");
System.out.print("Enter a value: ");
a=S.nextInt();
System.out.print("Enter b value: ");
b=S.nextInt();
System.out.print("Enter c value: ");
c=S.nextInt();
System.out.println("Enter quadratic equation:
"+a+"x^2+("+b+")x+"+c);
d=b*b-4*a*c;
if(d>0){
System.out.println("roots are real and unequal");
root1=(-b+ Math.sqrt(d)/(2*a));
root2=(+b+ Math.sqrt(d)/(2*a));
System.out.println("First root is: "+root1);
System.out.println("Second root is: "+root2);}
else if(d==0){
System.out.println("roots are real and equal");
root1=(-b+ Math.sqrt(d)/(2*a));
System.out.println("Root: "+root1);}
else{
System.out.println("roots are imaginary");
}
}}
```

Output:

```
Given quadratic equation:  $ax^2+bx+c$ 
Enter a value: 1
Enter b value: -5
Enter c value: 6
Enter quadratic equation:  $1x^2+(-5)x+6$ 
roots are real and unequal
First root is: 5.5
Second root is: -4.5
```

Lab Program: 03

AIM: Write a JAVA program to display the Fibonacci sequence.

PROGRAM:

```
import java.util.Scanner;
class fibbnonciseriess{
public static void main(String args[]){
int n1=0,n2=1,n3;
Scanner sc=new Scanner(System.in);
System.out.println("Enter a number: ");
int n=sc.nextInt();
System.out.print(n1+" "+n2);
for(int i=2;i<n;i++)
{
n3=n1+n2;
System.out.print(" "+n3);
n1=n2;
n2=n3;
}
}}
```

Output:

Enter a number:

5

0 1 1 2 3

Lab Program: 04

AIM: Write a JAVA program give example for command line arguments.

PROGRAM:

```
// LabTask4.java
public class LabTask4
{
    public static void main(String args[])
    {
        int x=Integer.parseInt(args[0]);
        int y=Integer.parseInt(args[1]);
        int sum=x+y;
        System.out.println("Sum of two numbers is:"+sum);
    }
}
```

Output:

Sum of two numbers is:30

Lab Program: 05

AIM: Write a JAVA program to give the example for 'this' operator. And also use 'this' keyword as return statement.

PROGRAM:

```
//LabTask5
//biggest.java
import java.util.Scanner;
public class LabTask5{
    private int num1;
    private int num2;
    public LabTask5 SetValues()
    {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter 'num1' value: ");
        int num1=sc.nextInt();
        System.out.print("Enter 'num2' value: ");
        int num2=sc.nextInt();
        this.num1=num1;
        this.num2=num2;
        return this;
    }
    public void display()
    {
        if(num1>num2)
            System.out.print("num1 is bigger");
        else
            System.out.print("num2 is bigger");
    }
    public static void main(String args[])
    {
        LabTask5 obj = new LabTask5();
        obj=obj.SetValues();
        obj.display();
    }
}
```

Output:

```
Enter 'num1' value: 5
Enter 'num2' value: 9
num2 is bigger
```

Lab Program: 06

AIM: Write a JAVA program to demonstrate static variables, methods, and blocks.

PROGRAM:

```
public class Labtask6{
    static String s="GIET";
    static int x=300;
    static int y;
    static void fun(int z){
        System.out.println("s: "+s);
        System.out.println("x= "+x);
        System.out.println("y= "+y);
        System.out.println("z= "+z);
    }
    static {
        System.out.println("Static block is invoked");
        y=x-100;
    }
    public static void main(String args[]){
        fun(10);
    }
}
```

Output:

```
Static block is invoked
s: GIET
x= 300
y= 200
z= 10
```

Lab Program: 07

AIM: Write a JAVA program to search for an element in a given list of elements (linear search).

PROGRAM:

```
import java.util.Scanner;
public class Labtask7{
public static void main(String args[]){
int n,key,i;
Scanner sc=new Scanner(System.in);
System.out.println("Enter no of elements: ");
n=sc.nextInt();
int arr[]=new int[n];
System.out.println("ELEMENTS ARE: ");
for(i=0;i<n;i++)
arr[i]=sc.nextInt();
System.out.println("Enter the key value: ");
key=sc.nextInt();
for(i=0;i<n;i++){
if(arr[i]==key){
System.out.println(key+" is found at index "+i);
break;
}
}
if(i==n)
System.out.println(key+" is not found");
}
}
```

Output:

```
Enter no of elements:
5
ELEMENTS ARE:
85
63
45
21
74

Enter the key value:
63
63 is found at index 1
```

Lab Program: 08

AIM: Write a JAVA program to search for an element in a given list of elements using binary search mechanism.

PROGRAM:

```
import java.util.*;
public class Labtask8{
public static void binarysearch(int arr[],int l,int h,int key){
while(l<=h){
int mid=(l+h)/2;
if(arr[mid]==key){
System.out.println("*****"+key+" is found at index "+mid+"*****");
break;}
else if(arr[mid]<key){
l=mid+1;}
else if(arr[mid]>key){
h=mid-1;}
else{
System.out.println("*****The Element is not found*****");}
}}
public static void main(String args[]){
int n,key,i,j,l,h;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the Size of the array:: ");
n=sc.nextInt();
System.out.println("****Enter the "+n+" elements****");
int arr[]=new int[n];
for(i=0;i<n;i++){
arr[i]=sc.nextInt();}
for(i=0;i<n;i++){
for(j=i+1;j<n;j++){
if(arr[i]>arr[j]){
int temp=arr[i];
arr[i]=arr[j];
arr[j]=temp;}
}}
System.out.println("The Sorted Array Elements are: ");
for(i=0;i<n;i++){
System.out.println(" "+arr[i]);}
System.out.println("Enter the key value: ");
key=sc.nextInt();
l=0;
h=n-1;
binarysearch(arr,l,h,key);
}}
```


Output:

```
Enter the Size of the array::
5
***Enter the 5 elements***
56
43
21
23
74

The Sorted Array Elements are:
21
23
43
56
74
Enter the key value:
23
****23 is found at index 1****
```

Lab Program:09

AIM: Write a JAVA program to sort given list of numbers.

PROGRAM:

```
import java.util.Scanner;
public class LabTask9
{
    public static void main(String args[])
    {
        int n,temp;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter array size:");
        n=sc.nextInt();
        int a[]=new int[n];
        System.out.println("Enter the elements:");
        for(int i=0;i<n;i++)
        {
            a[i]=sc.nextInt();
        }
        for(int i=0;i<n;i++)
        {
            for(int j=i+1;j<n;j++)
            {
                if(a[i]>a[j])
                {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
        System.out.print("Sorting The Numbers In Ascending
Order::");
        for(int i=0;i<n;i++)
            System.out.print(a[i]+" ");
    }
}
```

Output:

```
Enter array size:5
Enter the elements:
56
63
96

74
25
Sorting The No.s In Ascending Order:25 56 63 74 96
```

Lab Program:10

AIM: Write a JAVA program to sort an array of strings.

PROGRAM:

```
import java.util.Arrays;
public class LabTask10
{
    public static void main(String args[])
    {
        String[] Heroes={"Adolf Hitler","Benito Mussolini","Joseph
Stalin","Vladimir Putin","Barack Obama","Elon Musk","Nicholas
Tesla","Satya Nadella"};
        //Arrays.sort(Heroes);
        int n=Heroes.length;
        for(int i=0;i<n;i++)
        {
            for(int j=i+1;j<n;j++)
            {
                if(Heroes[i].compareTo(Heroes[j])>0)
                {
                    String temp=Heroes[i];
                    Heroes[i]=Heroes[j];
                    Heroes[j]=temp;
                }
            }
        }
        System.out.println(Arrays.toString(Heroes));
    }
}
```

Output:

```
[Adolf Hitler, Barack Obama, Benito Mussolini, Elon Musk,
Joseph Stalin, Nicholas Tesla, Satya Nadella, Vladimir Putin]
```

Lab Program:11

AIM: Write a JAVA program to check whether given string is palindrome or not.

PROGRAM:

```
import java.util.Scanner;
public class LabTask11
{
    public static void main(String args[])
    {
        String str, rev="";
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the string: ");
        //mom,dad,madam,civic,malayalam,radar,level,rotor,racecar,refer...
        str=sc.nextLine();
        int l=str.length();
        for(int i=l-1;i>=0;i--){
            rev=rev+str.charAt(i);
        }
        if(str.equals(rev))
            System.out.println(str+" is a Palindrome");
        else
            System.out.println(str+" is not a Palindrome");
    }
}
```

Output:

```
Enter the string:
malayalam
malayalam is a Palindrome
```

Lab Program:12

AIM: Write a JAVA program to determine the addition of two matrices.

PROGRAM:

```
import java.util.Scanner;
public class LabTask12{
    public static void main(String args[])
    {
        int x,y;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter no.of rows of a matrix: ");
        x=sc.nextInt();
        System.out.println("Enter no.of columns of a matrix: ");
        y=sc.nextInt();
        int mat1[][]=new int[x][y];
        int mat2[][]=new int[x][y];
        int sum[][]=new int[x][y];
        System.out.println("Enter elements of 1st matrix: ");
        for (int i=0;i<x;i++)
        {
            for(int j=0;j<y;j++)
                mat1[i][j]=sc.nextInt();
        }
        System.out.println("Enter elements of 2nd matrix: ");
        for(int i=0;i<x;i++)
        {
            for(int j=0;j<y;j++)
                mat2[i][j]=sc.nextInt();
        }
        for(int i=0;i<x;i++)
        {
            for(int j=0;j<y;j++)
                sum[i][j]=mat1[i][j]+mat2[i][j];
        }
        System.out.println("Addition of matrices: ");
        for(int i=0;i<x;i++)
        {
            for(int j=0;j<y;j++)
                System.out.print(sum[i][j]+"\\t");
            System.out.println();
            System.out.println();
        }
    }
}
```

Output:

```
Enter no.of rows of a matrix:
2
Enter no.of columns of a matrix:
2
Enter elements of 1st matrix:
4
3
2
1
Enter elements of 2nd matrix:
4
5
1
3
Addition of matrices:
8  8
3  4
```

Lab Program:13

AIM: Write a JAVA program to determine multiplication of two matrices.

PROGRAM:

```
import java.util.Scanner;
public class LabTask13{
    public static void main(String args[])
    {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter base of matrices: ");
        n=sc.nextInt();
        int mat1[][]=new int[n][n];
        int mat2[][]=new int[n][n];
        int mul[][]=new int[n][n];
        System.out.println("Enter elements of 1st matrix: ");
        for (int i=0;i<n;i++)
        {
            for(int j=0;j<n;j++)
                mat1[i][j]=sc.nextInt();
        }
        System.out.println("Enter elements of 2nd matrix: ");
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<n;j++)
                mat2[i][j]=sc.nextInt();
        }
        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
            {
                for(int k = 0; k < n; k++)
                    mul[i][j]=mul[i][j]+mat1[i][k]*mat2[k][j];
            }
        }
        System.out.println("Multiplication of matrices: ");
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<n;j++)
                System.out.print(mul[i][j]+" ");
            System.out.println();
            System.out.println();
        }
    }
}
```

Output:

```
Enter base of matrices:
2
Enter elements of 1st matrix:
2
3
4
1
Enter elements of 2nd matrix:
3
2
1
4
Multiplication of matrices:
9  16
13 12
```


Lab Program:14

AIM: Write JAVA program for the following

- A. Example for call by value.
- B. Example for call by reference.

PROGRAM:

//14.A.Call By Value.java

```
public class LabTask14a
{
    public static void main(String[] args)
    {
        int a=10;
        int b=20;
        System.out.println("Values of a and b before the call...");
        System.out.println("a="+a+",b="+b);
        swap(a,b);
        System.out.println("Values of a and b after the call...");
        System.out.println("a="+a+",b="+b);
    }
    public static void swap(int a,int b)
    {
        System.out.println("Inside swap(), before swapping...");
        System.out.println("a="+a+",b="+b);
        int temp=a;
        a=b;
        b=temp;
        System.out.println("Inside swap(), after swapping...");
        System.out.println("a="+a+",b="+b);
    }
}
```

Output:

```
Values of a and b before the call...
a=10,b=20
Inside swap(), before swapping...
a=10,b=20
Inside swap(), after swapping...
a=20,b=10
Values of a and b after the call...
a=10,b=20
```

//LabTask14.B Call ByReference

```
public class LabTask14b
{
    int a,b;
    public static void change(LabTask14b obj)
    {
        obj.a=20;
        obj.b=10;
    }
    public static void main(String args[])
    {
        LabTask14b obj=new LabTask14b();
        obj.a=10;
        obj.b=20;
        System.out.println("Before changing...");
        System.out.println("a="+obj.a+",b="+obj.b);
        change(obj);
        System.out.println("After changing...");
        System.out.println("a="+obj.a+",b="+obj.b);
    }
}
```

Output:

```
Before changing...
a=10,b=20

After changing...
a=20,b=10
```

Lab Program:15

AIM: Write a JAVA program that illustrates simple inheritance.

PROGRAM:

```
class Animal{
    void eat(){System.out.println("Eating...");}
}
class Dog extends Animal{
    void bark(){System.out.println("Barking...");}
}
public class LabTask15
{
    public static void main(String args[])
    {
        Dog d=new Dog();
        d.bark();
        d.eat();
    }
}
```

Output:

```
Barking...
Eating...
```

Lab Program:16

AIM: Write a JAVA program that illustrates multi-level inheritance.

PROGRAM:

```
class Animal{
    void eat(){System.out.println("Eating...");}
}
class Dog extends Animal{
    void bark(){System.out.println("Barking...");}
}
class BabyDog extends Dog{
    void weep(){System.out.println("Weeping...");}
}
public class LabTask16
{
    public static void main(String args[])
    {
        BabyDog d=new BabyDog();
        d.weep();
        d.bark();
        d.eat();
    }
}
```

Output:

```
Weeping...
Barking...
Eating...
```

Lab Program:17

AIM: Write a JAVA program demonstrating the difference between method overloading and method overriding.

PROGRAM:

//LabTask17a.java

//Method Overloading

```
class MotorBike
{
    private String startMethod = "Kick";
    public void start()
    {
        System.out.println(startMethod+" starting...");
    }

    public void start(String method)
    {
        this.startMethod = method;
        System.out.println(startMethod+" starting...");
    }
}

public class LabTask17a
{
    public static void main(String args[])
    {
        MotorBike b=new MotorBike();
        b.start();
        b.start("Self");
    }
}
```

Output:

```
Kick starting...
Self starting...
```

```
//Labtask17b.java
//Method Overriding
class MotorBike
{
    public void start()
    {
        System.out.println("Using kick paddle to start...");
    }
}
class SelfStartMotorBike extends MotorBike
{
    public void start()
    {
        System.out.println("Using self start button to start...");
    }
}
public class LabTask17b
{
    public static void main(String args[])
    {
        SelfStartMotorBike b=new SelfStartMotorBike();
        b.start();
    }
}
```

Output:

Using self start button to start...

Lab Program:18

AIM: Write a JAVA program demonstrating the difference between method overloading and constructor overloading.

PROGRAM:

```
//LabTask18a.java
//Method Overloading

class MotorBike
{
    private String startMethod = "Kick";
    public void start()
    {
        System.out.println(startMethod+" starting...");
    }

    public void start(String method)
    {
        this.startMethod = method;
        System.out.println(startMethod+" starting...");
    }
}

public class LabTask18a
{
    public static void main(String args[])
    {
        MotorBike b=new MotorBike();
        b.start();
        b.start("Self");
    }
}
```

Output:

```
Kick starting...
Self starting...
```

```
//LabTask18b
//Constructor Overloading
class LabTask18b
{
    String lang;
    LabTask18b()
    {
        this.lang="Java";
    }
    LabTask18b(String lang)
    {
        this.lang=lang;
    }
    public void getLang()
    {
        System.out.println("Programming Langauage: "+this.lang);
    }
    public static void main(String[] args)
    {
        LabTask18b obj1=new LabTask18b();
        LabTask18b obj2=new LabTask18b("C++");
        obj1.getLang();
        obj2.getLang();
    }
}
```

Output:

```
Programming Langauage: Java
Programming Langauage: C++
```


Lab Program:19

AIM: Write a JAVA program to give the example for 'super' keyword.

PROGRAM:

```
//LabTask19.java
class Animal{
    public void animalSound(){System.out.println("Animal makes sound");}
}
class Dog extends Animal{
    public void animalSound()
    {
        super.animalSound();
        System.out.println("Dog says: bow wow");
    }
}
public class LabTask19
{
    public static void main(String args[])
    {
        Dog d=new Dog();
        d.animalSound();
    }
}
```

Output:

Animal makes sound

Dog says: bow wow

Lab Program:20

AIM: Write a JAVA program illustrating multiple inheritance using interfaces.

PROGRAM:

```
//LabTask20.java

interface Printable
{
    void print();
}

interface Showable
{
    void show();
}

class LabTask20 implements Printable, Showable
{
    public void print(){System.out.println("Hello");}
    public void show(){System.out.println("World");}
    public static void main(String args[])
    {
        LabTask20 obj=new LabTask20();
        obj.print();
        obj.show();
    }
}
```

Output:

```
Hello
World
```

Lab Program:21

AIM: Write a JAVA program to illustrate the concept of final keyword in the program.

PROGRAM:

```
//LabTask21.java
public class LabTask21
{
    final int x=10;
    public static void main(String args[])
    {
        LabTask21 obj=new LabTask21();
        obj.x=12;
        System.out.println(obj.x);
    }
}
```

Output:

```
LabTask21.java:8: error: cannot assign a value to final
variable x
  obj.x=12;
      ^
1 error
```

Lab Program: 22

AIM: Write a JAVA program to create a package named mypack, and implement this package in number class.

PROGRAM:

```
package mypack;  
  
public class number  
{  
    public void add(int a,int b)  
    {System.out.println("Sum="+a+b);}  
}
```

//LabTask22

//ex1.java

```
import mypack.number;  
  
class pack  
{  
    public static void main(String args[])  
    {number obj=new number();  
        obj.add(30,50);  
        obj.add(20,50);  
        obj.add(10,50);  
    }}
```

Output:

C:\package> javac -d . number.java

C:\package> javac pack.java

C:\package> java pack

Sum=80

Sum=70

Sum=60

C:\package>

Lab Program : 23

AIM: Write a JAVA program to create a package named mypack and import it in circle class.

PROGRAM:

//LabTask23

//CircleArea.java

```
package mypack;

public class CircleArea{

    double a;

    public void calcArea(int r){

        a=3.14*r*r;

        System.out.println("Area of Circle is: "+a);

    }

}
```

//LabTask23

//Circle.java

```
package pack2;

import mypack.*;

public class Circle{

    public static void main(String args[])

    {

        CircleArea c=new CircleArea();

        c.calcArea (4);

    }

}
```

Output:

C:\circle> javac -d . CircleArea.java

C:\circle> javac Circle.java

C:\circle> java Circle

Area of Circle is: 5.24

C:\circle>

Lab Program:24

AIM: Write a JAVA program to give a simple example for abstract class.

PROGRAM:

//LabTask24.java

```
abstract class Animal{
    public abstract void animalSound();
    public void eat(){System.out.println("eating...");}

}
class Pig extends Animal{
    public void animalSound(){
        System.out.println("The pig says: wee wee");
    }
}
public class LabTask24
{
    public static void main(String args[])
    {
        Pig p=new Pig();
        p.animalSound();
        p.eat();
    }
}
```

Output:

The pig says: wee wee

Eating...

Lab program:25

AIM: Write a JAVA program that describes exception handling mechanism.

PROGRAM:

//LabTask25.java

```
public class LabTask25
{
    public static void main(String args[])
    {
        try
        {
            int x=67/0;
        }
        catch(ArithmeticException e1)
        {
            System.out.println("Exception 1: "+e1);
        }
        try
        {
            String s=null;
            System.out.println(s.length());
        }
        catch(NullPointerException e2)
        {
            System.out.println("Exception 2: "+e2);
        }
    }
}
```

Output:

```
Exception 1: java.lang.ArithmeticException: / by zero
Exception 2: java.lang.NullPointerException: Cannot invoke
"String.length()" because "s" is null
```

Lab Program:26

AIM: Write a JAVA program for example of try and catch block. In this check whether the given array size is negative or not.

PROGRAM:

//LabTask26.java

```
public class LabTask26
{
    public static void main(String args[])
    {
        try
        {
            int arr[] = new int[-6];
        }
        catch(NegativeArraySizeException e)
        {
            System.out.println("Generated exception :"+e);
        }
        finally
        {
            System.out.println("The 'try catch' is finished...");
        }
    }
}
```

Output:

```
Generated exception :java.lang.NegativeArraySizeException: -6
The 'try catch' is finished...
```


Lab Program:27

AIM: Write a JAVA program to illustrate sub class exception precedence over base class.

PROGRAM:

```
//LabTask27
//Room.java
import java.io.*;
class Building
{
    void color() throws Exception
    {
        System.out.println("Blue");
    }
}
class Room extends Building
{
    void color() throws Exception
    {
        System.out.println("White");
    }
    public static void main(String args[])
    {
        Building obj=new Room();
        try
        {
            obj.color();
        }
        catch(Exception e){}
    }
}
```

Output:

White

Lab Program:28

AIM: Write a JAVA program for handling of user defined exception by using throw.

PROGRAM:

//LabTask28.java

```
import java.util.Scanner;
class NegativeAmountException extends Exception
{
    String msg;
    NegativeAmountException(String msg)
    {    this.msg=msg; }
    public String toString()
    {
        return msg;
    }
}
public class LabTask28
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the amount: ");
        int amt=sc.nextInt();
        try
        {
            if(amt<0)
            {
                throw new NegativeAmountException("Invalid
Amount...");
            }
            else
            {
                System.out.println("Amount Deposited...");
            }
        }
        catch(NegativeAmountException e)
        {
            System.out.println(e);
        }
    }
}
```

Output:

Run1:

```
Enter the amount: 100
Amount Deposited...
```

Run2:

```
Enter the amount: -80
Invalid Amount...
```

Lab Program:29

AIM: Write a JAVA program to illustrate the concept of throws keyword.

PROGRAM:

```
//LabTask29
//Derived.java
import java.io.*;
class Base
{
    void method() throws Exception
    {
        System.out.println("Parent");
    }
}
public class Derived extends Base
{
    public static void main(String args[])
    {
        Base obj=new Base();
        try
        {
            obj.method();
        }
        catch(Exception e){}
    }
}
```

Output: Parent

Lab Program:30

AIM: Write a JAVA program to illustrate creation of threads using runnable class.(start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500 milliseconds).

PROGRAM:

```
//LabTask30.java
//ThreadSleep
public class LabTask30 implements Runnable
{
    public void run()
    {
        for(int i=6;i>=1;i--)
        {
            try
            {
                Thread.sleep(500);
                System.out.println(i);
            }
            catch(InterruptedException e)
            {
                System.out.println(e);
            }
        }
    }
    public static void main(String args[])
    {
        LabTask30 obj1=new LabTask30();
        Thread obj2=new Thread(obj1);
        obj2.start();
    }
}
```

Output:

```
6
5
4
3
2
1
```

Lab Program:31

AIM: Write a JAVA program to create a class MyThread in this class a constructor, call the base class constructor, using super and starts the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently.

PROGRAM:

```
//LabTask31.java
class MyThread extends Thread
{
    MyThread()
    {
        super();
        start();
    }
    public void run()
    {
        try
        {
            for (int i=3;i>=1;i--)
            {
                Thread.sleep(500);
                System.out.println("Running
"+Thread.currentThread().getName()+" : "+i);
            }
        }
        catch (InterruptedException ie)
        {
            System.out.println(ie);
        }
        System.out.println("Exiting
"+Thread.currentThread().getName()+"...");
    }
}
public class LabTask31
{
    public static void main(String args[])
    {
        MyThread obj=new MyThread();
        try
        {
            for(int i=3;i>=1;i--)
            {

                Thread.sleep(500);
                System.out.println ("Running
"+Thread.currentThread().getName()+" : "+i);
            }
        }
    }
}
```

```
        catch (InterruptedException ie)
        {
            System.out.println(ie);
        }
        System.out.println ("Exiting
"+Thread.currentThread().getName()+"...");
    }
}
```

Output:

```
Running main: 3
Running Thread-0: 3
Running main: 2
Running Thread-0: 2
Running main: 1
Running Thread-0: 1
Exiting Thread-0...
Exiting main...
```

Lab Program:32

AIM: Write a JAVA program to illustrate the concept of thread synchronization.

PROGRAM:

```
//LabTask32.java
class First
{
    public void display(String msg)
    {
        System.out.print ("["+msg);
        try
        {
            Thread.sleep(1000);}
        catch (InterruptedException e)
        {
            e.printStackTrace();}
        System.out.println ("]");
    }
}

class Second extends Thread
{
    String msg;
    First fobj;
    Second (First fp,String str)
    {
        fobj = fp;
        msg = str;
        start();
    }
    public void run()
    {
        synchronized(fobj)    //Synchronized block
        {
            fobj.display(msg);
        }
    }
}

public class LabTask32
{
    public static void main (String[] args)
    {
        First fnew = new First();
        Second ss = new Second(fnew, "welcome");
        Second ssl= new Second (fnew,"new");
        Second ss2 = new Second(fnew, "programmer");
    }
}
```

Output:

```
[welcome]
[programmer]
[new]
```

Lab Program:33

AIM: Write Java program by implementing the concepts of different collections as list, map and set.

PROGRAM:

```
//LabTask33.java
import java.util.*;
public class LabTask33
{
    public static void main(String args[])
    {
        //create a HashSet to store Strings
        HashSet<String> hs=new HashSet<String>();
        //Store some String elements
        hs.add("India");
        hs.add("America");
        hs.add("Japan");
        hs.add("China");
        hs.add("America");
        //view the HashSet
        System.out.println ("HashSet = " + hs);
        //add an Iterator to hs
        Iterator it = hs.iterator ();
        //display element by element using Iterator
        System.out.println("Elements Using Iterator: ");
        while(it.hasNext())
        {
            String s=(String)it.next();
            System.out.println(s);
        }

        //create an empty stack to contain Integer objects
        Stack<Integer> st=new Stack<Integer>();
        st.push(10);
        st.push(20);
        st.push(30);
        st.push(40);
        st.push(50);
        System.out.println(st);
        System.out.println

        ("Element at top of the stack is: "+st.peek());
        System.out.println

        ("Removing element at the TOP of the stack: "+st.pop());
        System.out.println("The new stack is: "+st);
    }
}
```



```
HashMap<Integer,String> hm=new HashMap<Integer,String>();  
hm.put((101),"Naresh");  
hm.put((102),"Rajesh");  
hm.put((103),"Suresh");  
hm.put((104),"Mahesh");  
hm.put((105),"Ramesh");  
Set<Integer> set=new HashSet<Integer>();  
set=hm.keySet();  
System.out.println(set);  
}  
}
```

Output:

```
HashSet = [Japan, China, America, India]  
Elements Using Iterator:  
Japan  
China  
America  
India  
[10, 20, 30, 40, 50]  
Element at top of the stack is: 50  
Removing element at the TOP of the stack: 50  
The new stack is: [10, 20, 30, 40]  
[101, 102, 103, 104, 105]
```

Lab Program:34

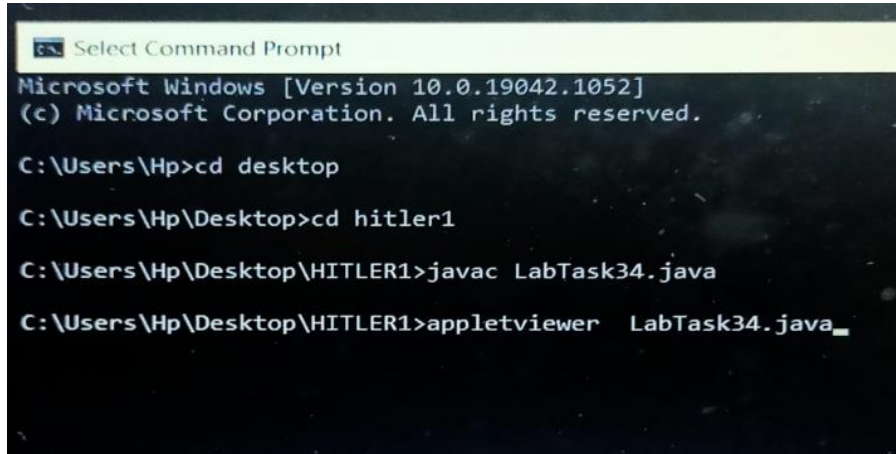
AIM: Write a JAVA program that describes the life cycle of an applet.

PROGRAM:

```
//LabTask34.java
import java.applet.*;
import java.awt.*;
/*<applet code="LabTask34.class" width="500" height="500">
</applet>*/
public class LabTask34 extends Applet
{
    String str;
    public void init()
    {
        str="Welcome to Java Applet";
        System.out.println("Inside init method");
        setBackground(Color.cyan);
        setForeground(Color.blue);
    }
    public void start()
    {
        System.out.println("Inside start method");
    }
    public void paint(Graphics g)
    {
        Font f=new Font("Arial",3,27);
        g.setFont(f);
        g.drawString(str,200,200);
        System.out.println("Inside paint method");
    }
    public void stop()
    {
        System.out.println("Inside stop method");
    }
}
```

```
public void destroy()
{
    System.out.println("Inside destroy method");
}
}
```

OUTPUT:



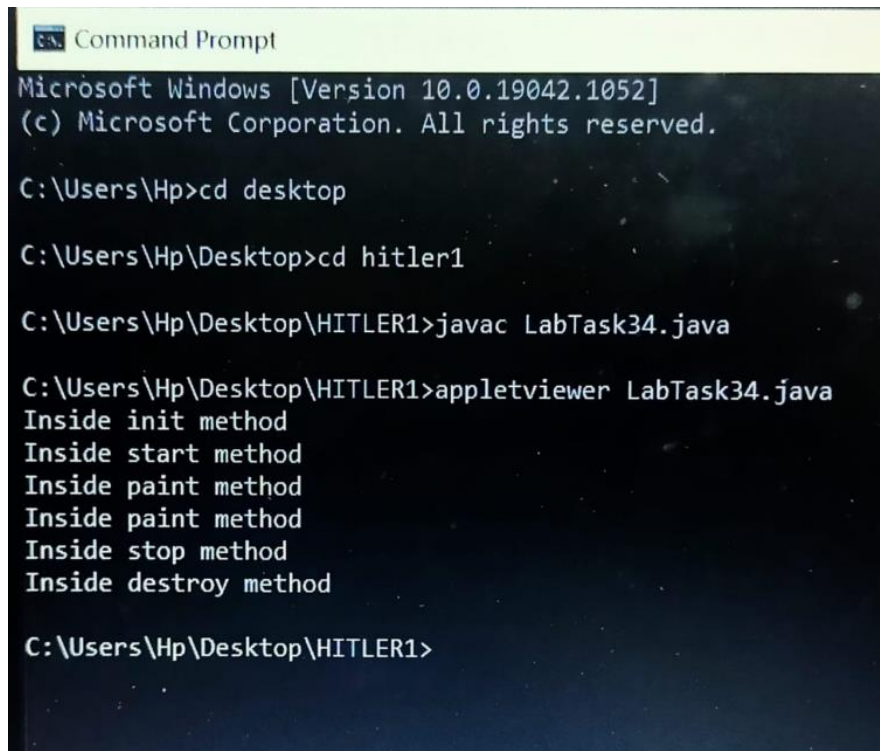
```
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hp>cd desktop

C:\Users\Hp\Desktop>cd hitler1

C:\Users\Hp\Desktop\HITLER1>javac LabTask34.java

C:\Users\Hp\Desktop\HITLER1>appletviewer LabTask34.java_
```



```
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

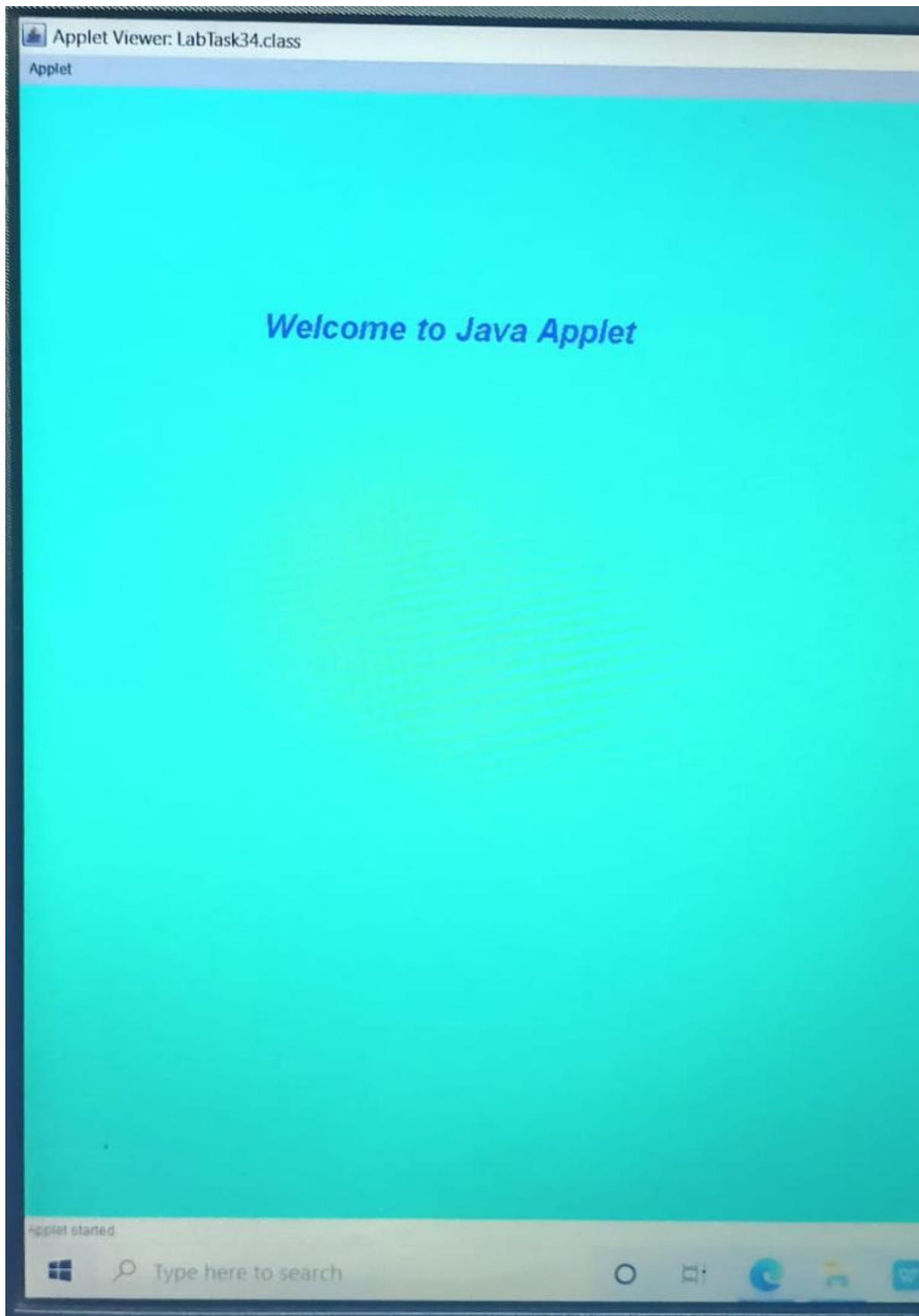
C:\Users\Hp>cd desktop

C:\Users\Hp\Desktop>cd hitler1

C:\Users\Hp\Desktop\HITLER1>javac LabTask34.java

C:\Users\Hp\Desktop\HITLER1>appletviewer LabTask34.java
Inside init method
Inside start method
Inside paint method
Inside paint method
Inside stop method
Inside destroy method

C:\Users\Hp\Desktop\HITLER1>
```



Lab Program:35

AIM: Write a JAVA program to design a laughing baby face.

PROGRAM:

```
//LabTask35.java
import java.applet.*;
import java.awt.*;
/*<applet code="LabTask35.class" width="600" height="800">
</applet>*/
public class LabTask35 extends Applet
{
    public void init()
    {
        setBackground(Color.black);
        setForeground(Color.yellow);
    }
    public void paint(Graphics g)
    {
        Font f=new Font("Arial",1,28);
        g.setFont(f);
        g.drawString("Folks! I'm a Laughing Baby Face",340,550);
        g.drawOval(75,460,40,40);
        g.fillOval(75,460,40,40);
        g.drawLine(147,460,147,560);
        g.drawOval(175,460,40,40);
        g.fillOval(175,460,40,40);
        g.drawOval(20,400,260,260);
        g.drawArc(80,535,135,80,180,180);
    }
}
```

OUTPUT :

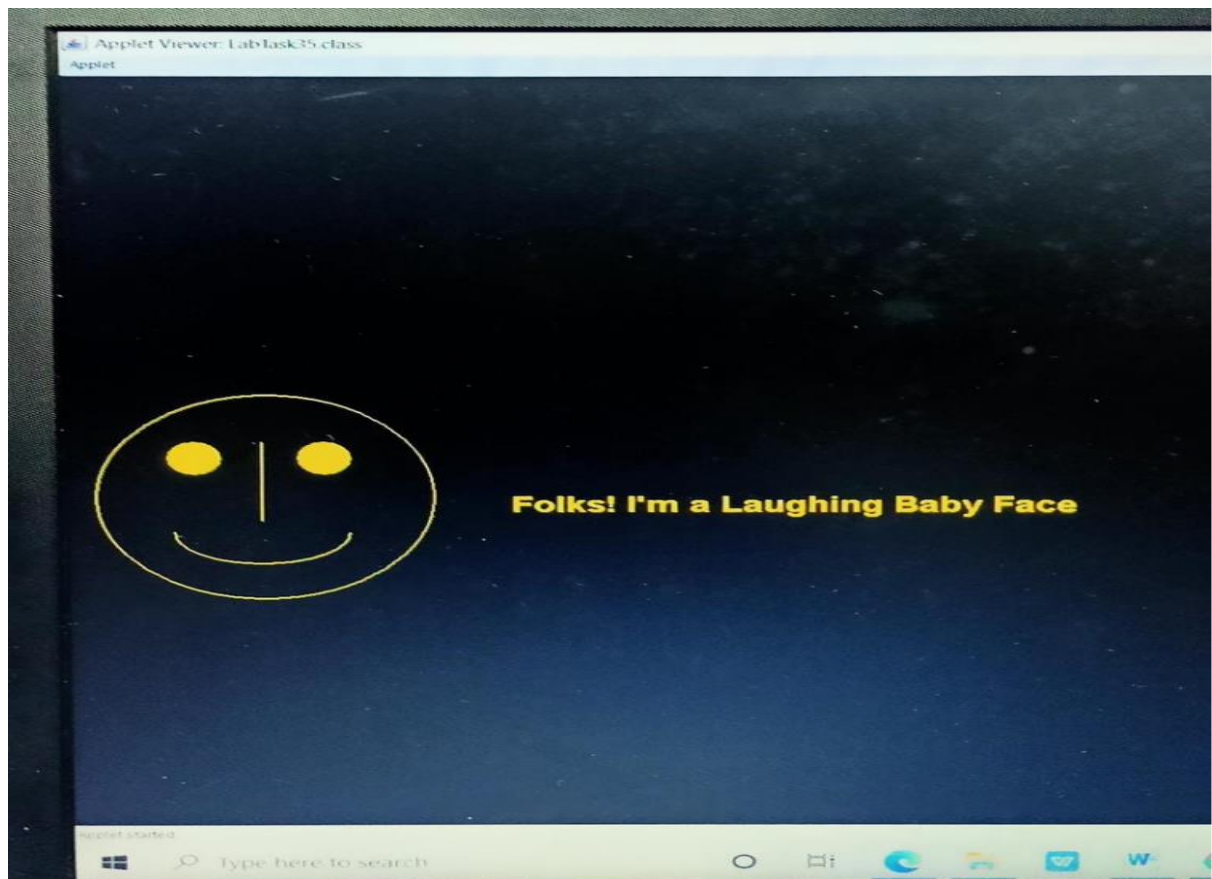
```
Command Prompt
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hp>cd desktop

C:\Users\Hp\Desktop>cd hitler1

C:\Users\Hp\Desktop\HITLER1>javac LabTask35.java

C:\Users\Hp\Desktop\HITLER1>appletviewer LabTask35.java
```



Lab Program:36

AIM: Write a JAVA program to create a simple calculator.

PROGRAM:

```
//LabTask36.java
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="LabTask36.class" width="700" height="200">
  </applet>*/
public class LabTask36 extends Applet implements ActionListener
{
    String str="";
    TextField t1,t2,t3;
    Button b1,b2,b3,b4,b5;
    Label l1,l2,l3;
    public void init()
    {
        l1=new Label("Enter 1st value:");
        add(l1);
        l2=new Label("Enter 2nd value:");
        add(l2);
        l3=new Label("Result: ");
        add(l3);
        t1=new TextField(10);
        add(t1);
        t2=new TextField(10);
        add(t2);
        t3=new TextField(10);
        add(t3);
        b1=new Button("add");
        b2=new Button("sub");
        b3=new Button("mul");
        b4=new Button("div");
```

```

        b5=new Button("mod");
        add(b1);
        add(b2);
        add(b3);
        add(b4);
        add(b5);

        l1.setBounds(50,100,100,20);
        l2.setBounds(50,140,100,20);
        l3.setBounds(50,180,100,20);
        t1.setBounds(200,100,100,20);
        t2.setBounds(200,140,100,20);
        t3.setBounds(200,180,100,20);
        b1.setBounds(50,250,50,20);
        b2.setBounds(110,250,50,20);
        b3.setBounds(170,250,50,20);
        b4.setBounds(230,250,50,20);
        b5.setBounds(290,250,50,20);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);

        setLayout(null);
        setVisible(true);
        setSize(400,350);
        setBackground(Color.black);
        setForeground(Color.white);
    }

    public void paint(){}

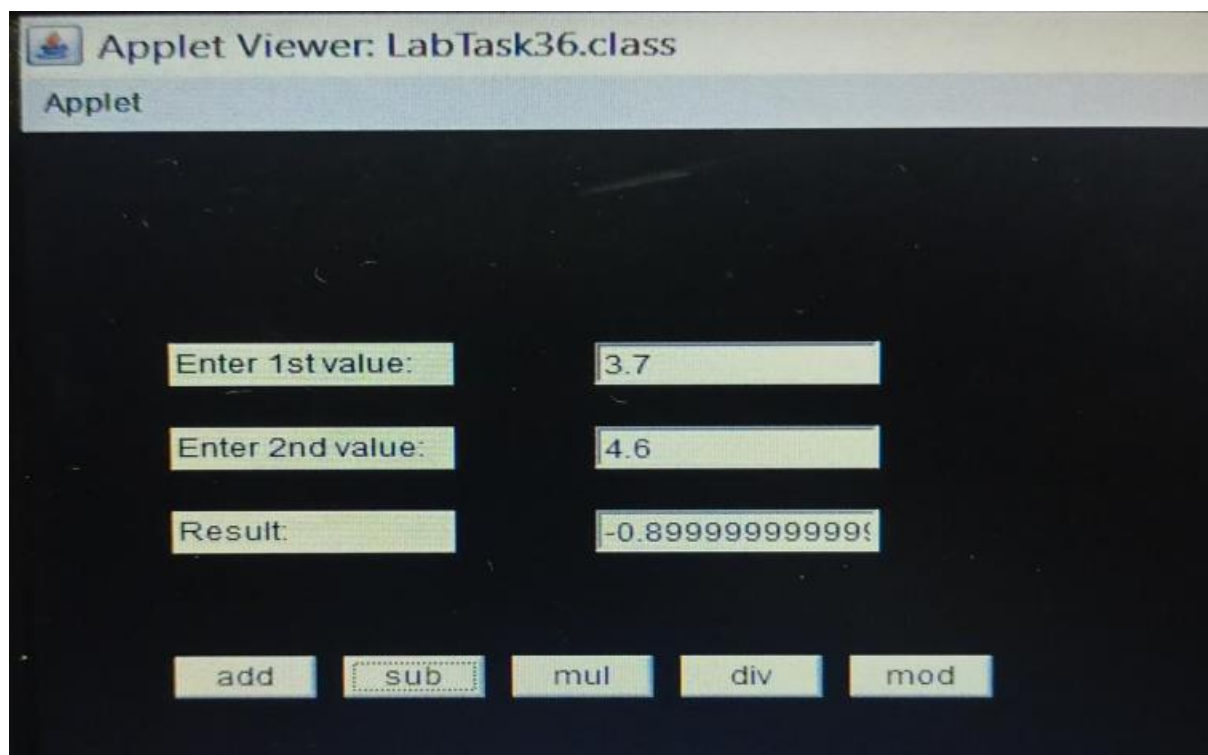
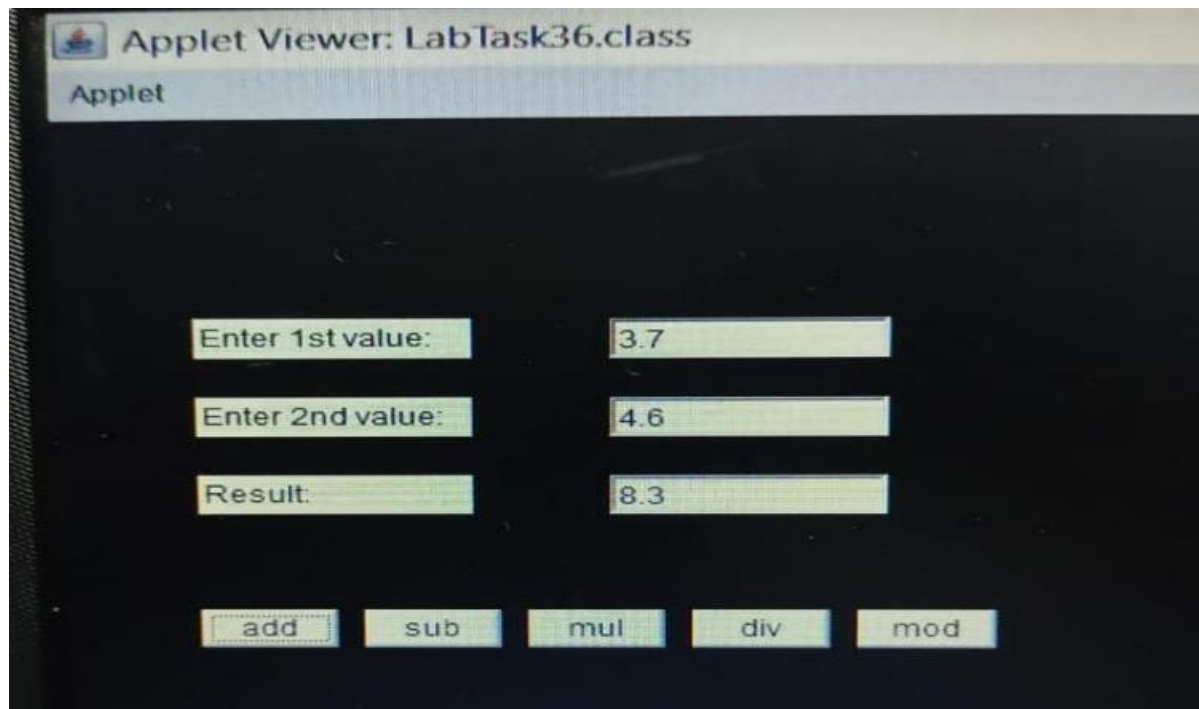
    public void actionPerformed(ActionEvent e)
    {
        str=e.getActionCommand();
        double a=Double.parseDouble(t1.getText());
        double b= Double.parseDouble(t2.getText());

```



```
        if(str=="add")
        {
            double sum=a+b;
            t3.setText(""+sum);
        }
        else if(str=="sub")
        {
            double sub=a-b;
            t3.setText(""+sub);
        }
        else if(str=="mul")
        {
            double mul=a*b;
            t3.setText(""+mul);
        }
        else if(str=="div")
        {
            double div=a/b;
            t3.setText(""+div);
        }
        else if(str=="mod")
        {
            int x=Integer.parseInt(t1.getText());
            int y=Integer.parseInt(t2.getText());
            int mod=x%y;
            t3.setText(""+mod);
        }
        repaint();
    }}
}
```

OUTPUT :



Applet Viewer: LabTask36.class

Applet

Enter 1st value: 3.7

Enter 2nd value: 4.6

Result: 17.02

add sub mul div mod

Applet Viewer: LabTask36.class

Applet

Enter 1st value: 3.7

Enter 2nd value: 4.6

Result: 0.804347826086

add sub mul div mod

Applet Viewer: LabTask36.class

Applet

Enter 1st value: 3

Enter 2nd value: 4

Result: 3

add sub mul div mod