

DEPARTMENT OF COMPUTER SCIENCE

Government Arts and Science College (Women)

Sathankulam



CERTIFICATE

This is to certify that the bonafide record of the practical work done in
by Reg. No.....
of during the academic year 2020-2021.

Head of the Department

Staff In-Charge

Submitted for the practical examination held on

External Examiner

- 1.
- 2.

CONTENTS

Sl. no	Ex. No	Date	Title	Page No	Staff Sign
1	1	17-08-2020	STUDENT DETAILS	1	
2	2a	20-08-2020	OVERLOADING CONSTRUCTOR	4	
3	2b	20-08-2020	METHOD OVERLOADING	6	
4	b	24-08-2020	COMPLEX NUMBER ADDITION	8	
5	4	28-08-2020	METHOD OVERRIDING	11	
6	5	08-09-2020	INTERFACE DEMO	14	
7	6	12-09-2020	THREAD DEMO	16	
8	7	15-09-2020	DESIGN A CALCULATOR	19	
9	8	19-09-2020	DETAILS OF CAR	23	
10	9	22-09-2020	BACKGROUND COLOR	28	
11	10a	26-09-2020	NegativeArraySizeException	35	
12	10b	29-09-2020	ArrayIndexOutOfBoundsException	29	

JAVA PRACTICAL LIST

1. Define a class called Student with the attributes name, reg_number and marks obtained in four subjects(m1,m2,m3,m4).Write a suitable constructor and methods to find the total mark obtained by the student and display the details of the student.
2. Write a Java program to find the area of a square, rectangle and triangle by
(i) Overloading Constructor (ii) Overloading Method.
3. Write a java program to add two complex numbers. [Use passing object as argument and return object].
4. Define a class called Student_super with data members name, roll number and age. Write a suitable constructor and a method output () to display the details. Derive another class Student from Student_super with data members height and weight. Write a constructor and a method output () to display the details which overrides the super class method output().[Apply method Overriding concept].
5. Write a java program to create an interface called Demo, which contains a double type constant, and a method called area () with one double type argument. Implement the interface to find the area of a circle.
6. Write a java program to create a thread using Thread class.
7. Write a java program to Design a calculator to perform only addition and division. It must contains three Buttons with labels +, / and =, and a TextField to get input and display the result.
8. Create an applet with four Checkboxes with labels MARUTI-800, ZEN, ALTO and ESTEEM and a Text area object. The program must display the details of the car while clicking a particular Checkbox.
9. Write a Java program, which creates a window with a check box group with boxes for the colors, Violet, Indigo, Yellow, Orange, Red, Blue, and Green. When the button is selected the background color must change accordingly.
10. Write a Java program to throw the following exception,
 - 1) Negative Array Size
 - 2) Array Index out of Bounds

Ex.No : 1
17-08-2020

STUDENT DETAILS

Aim: To display the details of student using constructor

Algorithm:

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step 3 : Create a class “student”.

Step 3.1: Declare variables for name, registerno and four marks.

Step 3.2: Define the constructor with six parameters.

Assign values for the variables.

Step 3.3: Define a method “display”;

Find total and display all the details of student.

Step 4: Define the main class “prg1”.

Step 4.1: Define the main method.

Step 4.1.1: Create object “s1” for the class “student” with the value for details of student as arguments.

Step 4.1.2: using object “s1” call the method “display”

Step 5: Save the program; compile and execute.

//prg1 -- student Mark details

```
import java.io.*;
class student
{
    String name;
    int regno,m1,m2,m3,m4;
    student(String name1,int regno1,int mm1,int mm2,int mm3,int mm4)
    {
        name=name1;
        regno=regno1;
        m1=mm1;
        m2=mm2;
        m3=mm3;
        m4=mm4;
    }
    void display()
    {
        int total;
        total=m1+m2+m3+m4;
        System.out.println("STUDENT DETAILS");
        System.out.println("Name      : "+name);
        System.out.println("Regno   : "+regno);
        System.out.println("MArk1   : "+m1);
        System.out.println("MArk2   : "+m2);
        System.out.println("MArk3   : "+m3);
        System.out.println("MArk4   : "+m4);
    }
}

class prg1
{
    public static void main(String args[])
    {
        student s1=new student("Alagu Amsa Jothi",101,89,90,87,99);
        s1.display();
    }
}
```

OUTPUT

```
E:\java>javac prg1.java
```

```
E:\java>java prg1
```

```
STUDENT DETAILS
```

```
Name   : Alagu Amsa Jothi
```

```
Regno  : 101
```

```
MArk1  : 89
```

```
MArk2  : 90
```

```
MArk3  : 87
```

```
MArk4  : 99
```

```
E:\java
```

Result: Thus the java program to display the details of the student using constructor and method is executed successfully.

Ex. No: 2a
20-08-2020

OVERLOADING CONSTRUCTOR

Aim: To find the area of square, rectangle and triangle by using constructor overloading

Algorithm:

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step3 : Create a class “overload”.

Step 3.1: Define constructor with one integer type argument to find and print the area of square.

Step 3.2: Define constructor with two integer type arguments to find and print the area of rectangle.

Step 3.3: Define constructor with two double type arguments to find and print the area of triangle.

Step 4: Define the main class “prg2a”.

Step 4.1: Define the main method.

Step 4.1.1: Create object for the class “overload” with one integer value as argument

Step 4.1.2: Create object for the class “overload” with two integer values as arguments.

Step 4.1.3: Create object for the class “overload” with two double type values as arguments.

Step 5: Save the program; compile and execute.

```

// prg2a
//constructor overloading
import java.io.*;
class overload
{
    overload(int a)
    {
        System.out.println("Area of Square = "+(a*a));
    }
    overload(int l,int b)
    {
        System.out.println("Area of Rectangle = "+(l*b));
    }
    overload(double b,double h)
    {
        System.out.println("Area of Triangle = "+(b*h/2));
    }
}
class prg2a
{
    public static void main(String args[])
    {
        overload o1=new overload(10);
        overload o2=new overload(10,20);
        overload o3=new overload(10.5,20.5);
    }
}

```

OUTPUT

E:\java>javac prg2a.java

E:\java>java prg2a

Area of Square = 100

Area of Rectangle = 200

Area of Triangle = 107.625

E:\java>

Result : Thus the program to find the area of a square, rectangle and triangle using Overloading Constructor is executed successfully.

Ex. No 2b
20-08-2020

METHOD OVERLOADING

Aim: To find the area of square, rectangle and triangle by using method overloading

Algorithm:

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step3 : Create a class “overload”.

Step 3.1: Define method area() with one integer type argument to find and print the area of square.

Step 3.2: Define method area() with two integer type arguments to find and print the area of rectangle.

Step 3.3: Define method area() with two double type arguments to find and print the area of triangle.

Step 4: Define the main class “prg2b”.

Step 4.1: Define the main method.

Step 4.1.1: Create object “o1” for the class “overload”

Step 4.1.2: Using object “o1” call method area() with one integer value.

Step 4.1.2: Using object “o1” call method area() with two integer values.

Step 4.1.2: Using object “o1” call method area() with two double values.

Step 5: Save the program; compile and execute.

```

// prg2b---Method overloading
import java.io.*;
class overload
{
    void area(int a)
    {
        System.out.println("Area of Square = "+(a*a));
    }
    void area(int l,int b)
    {
        System.out.println("Area of Rectangle = "+(l*b));
    }
    void area(double b,double h)
    {
        System.out.println("Area of Triangle = "+(b*h/2));
    }
}
class prg2b
{
    public static void main(String args[])
    {
        overload o1=new overload();
        o1.area(10);
        o1.area(10,20);
        o1.area(10.5,20.5);
    }
}

```

OUTPUT

E:\java>javac prg2b.java

E:\java>java prg2b

Area of Square = 100

Area of Rectangle = 200

Area of Triangle = 107.625

E:\java>

Result : Thus the program to find the area of a square, rectangle and triangle by method overloading is executed successfully.

Ex. No:3
24-08-2020

COMPLEX NUMBER ADDITION

Aim: To add two complex numbers by passing objects as arguments.

Algorithm:

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step3 : Create a class “complex”.

Step 3.1: Declare two double type variables a, b.

Step 3.2: Define constructor with no parameters.

Step 3.3: Define constructor with two double type variables as arguments, and initialize a and b.

Step 3.4: Define method display to print the values of a and b

Step 3.5: Define a method add, with one object of type complex and return type complex.

Step 3.5.1: Create a complex object.

Step 3.5.2: Add the a values of both complex objects; add the b values of two complex objects and store in a complex object.

Step3.5.3: Return the complex object

Step 4: Define the main class “prg3”.

Step 4.1: Define the main method.

Step 4.1.1: Create object “c1” for the class “complex” with two integer values as arguments

Step 4.1.2: Create object “c2” for the class “complex” with two integer values as arguments

Step 4.1.3: Create object reference variable c3 for the class “complex”.

Step 4.1.4: using object “c1” call method “add” by using object “c2” as argument. The result returned is stored in object c3.

Step 4.1.5. Call method display() using object c1

Step 4.1.5. Call method display() using object c2

Step 4.1.5. Call method display() using object c3

Step 5: Save the program; compile and execute.

//prg3 - Complex Number

```
import java.io.*;
class complex
{
    double a,b;
    complex()
    {
    }
    complex(double a1,double b1)
    {
        a=a1;
        b=b1;
    }
    void display()
    {
        System.out.println(a+" "+b+"i");
    }
    complex add(complex o)
    {
        complex temp=new complex();
        temp.a=a+o.a;
        temp.b=b+o.b;
        return temp;
    }
}
class prg3
{
    public static void main(String args[])
    {
        complex c1=new complex(10,15);
        complex c2=new complex(20,12);
        complex c3;
        c3=c1.add(c2);
        System.out.println("Complex number1");
        c1.display();
        System.out.println("Complex number2");
        c2.display();
        System.out.println("Complex number3");
        c3.display();
    }
}
```

OUTPUT

```
E:\java>javac prg3.java
```

```
E:\java>java prg3  
Complex number1  
10.0+15.0i  
Complex number2  
20.0+12.0i  
Complex number3  
30.0+27.0i
```

```
E:\java>
```

Result: Thus a java program to add two complex numbers is executed successfully.

Ex. No: 4
28-08-2020

METHOD OVERRIDING

Aim: To display the details of the student using method overriding.

Algorithm

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step 3 : Create a class “student_super”.

Step 3.1: Declare variables , name, rollno, age.

Step 3.2: Define a constructor with 3 arguments name 2, rollno2, age2, and assign value to member variables name, rollno age .

Step 3.3: Define the output () method to display the student details, name, roll no, and age.

Step 4: Define another class “ student” inherited from “student- super.

Step 4.1: Declare variable height and weight

Step 4.2 Create constructor for student with five arguments name1, rollno1, age1 , height1 and weight1

Step 4.2.1: Call super class constructor using super (name1 , rollno1 , age1)

Step 4.1.2: Assign height1 and weight1 to height and weight .

Step 4.3: Define the method output()

Step 4.3.1: Invoke super class’ output() method

Step 4.3.2: Display the value of height and weight

Step 5: Define the class prg4

Step 5.1 : Define the main method

Step 5.1.1: Create object “sobj”, for subclass student with values for arguments name, rollno, age, height, weight.

Step 5.1.2: Invoke the output() method using subclass object.

Step 6: Save the program; compile and execute.

// prg 4 – Method Overriding

```
import java.io.*;
class student_super
{
    String name;
    int rollno,age;
    student_super(String name2,int rollno2,int age2)
    {
        name=name2;
        rollno=rollno2;
        age=age2;
    }
    void output()
    {
        System.out.println("Name="+name);
        System.out.println("Rollno="+rollno);
        System.out.println("Age="+age);
    }
}
class student extends student_super
{
    double height,weight;
    student(String name1,int rollno1,int age1, double height1, double weight1)
    {
        super(name1,rollno1,age1);
        height=height1;
        weight=weight1;
    }
    void output()
    {
        super.output();
        System.out.println("Height="+height);
        System.out.println("Weight="+weight);
    }
}
class prg4
{
    public static void main(String args[])
    {
        student subj=new student("Kanista", 14,19,5.5,53);
        subj.output();
    }
}
```

OUTPUT

```
E:\java>javac prg4.java
```

```
E:\java>java prg4
```

```
Name=Kanista
```

```
Rollno=14
```

```
Age=19
```

```
Height=5.5
```

```
Weight=53.0
```

```
E:\java>
```

Result: Thus a java program using method overriding is executed successfully.

Ex. No: 5
08-09-2020

INTERFACE DEMO

Aim: To implement an interface and find the area of a circle.

Algorithm

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step3 : Create an interface called “demo”

Step 3.1: Declare double type constant $\pi=3.14159$

Step 3.2: Declare a method area(), with one double type argument to find the area of a circle.

Step 4: Define a class “ prg5” which implements the interface “demo”.

Step 4.1: Define the method “circle ()” to find the area of the circle.

Step 4.2: Define a method “display ()” to display the area.

Step 4.3: Define the “main ()” method.

Step 4.3.1: Define object for the class “prg5”

Step 4.3.2: Invoke the method “area ()”

Step 4.3.3: Invoke the method “display ()”.

Step 5: Save the program; compile and execute.

```

// prg5 – Interface Demo
import java.io.*;
interface demo
{
    public static final double pi=3.14159;
    public void area (double r);
}
public class prg5 implements demo
{
    double ar;
    public void area(double r)
    {
        ar=pi*r*r;
    }
    void display()
    {
        System.out.println("Area of circle= "+ar);
    }
    public static void main(String args[])
    {
        prg5 p5=new prg5 ();
        p5.area(10);
        p5.display();
    }
}

```

OUTPUT

E:\java>javac prg5.java

E:\java>java prg5
Area of circle= 314.159

E:\java>

Result: Thus a java program to create an interface is executed successfully.

Ex. No 6
12-09-2020

Thread Using Thread Class

Aim: To write a program for demonstrating thread using Thread class

Algorithm

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step3 : Create a class “NewThread” which is inherited from the class “Thread””

Step 3.1: Define the constructor for “NewThread” and inside constructor invoke start() method

Step 3.2: Define run() method and write the thread handling code within try/catch block.

Step 4: Define a class “ prg6”.

Step 4.1: Define the “main ()” method.

Step 4.1.1: In main() method create a reference for “NewThread” class . Write code for main thread inside a try/catch block

Step 5: Save the program; compile and execute.

//prg6 – Thread Demo

```
import java.io.*;
class NewThread extends Thread
{   NewThread()
    {
        super("Demo Thread");
        System.out.println("Child Thread"+this);
        start();
    }
    public void run()
    {   try
        {
            for (int i=10;i>0;i--)
            {
                System.out.println("Child Thread : "+i);
                Thread.sleep(500);
            } // end of for
        }
        catch(InterruptedException e)
        {   System.out.println("Child Interrupted");   }
        System.out.println("Exiting Child Thread");
    }
}
class prg6
{   public static void main(String args[])
    {
        new NewThread();
        try
        {
            for(int i=10;i>0;i--)
            {
                System.out.println("Main Thread"+i);
                Thread.sleep(1000);
            }
        }
        catch(InterruptedException e)
        {   System.out.println("Main Thread interrupted");   }
        System.out.println("Main Thread Exiting");
    }
}
```

OUTPUT

```
E:\java>java prg6
Child ThreadThread[Demo Thread,5,main]
Main Thread10
Child Thread : 10
Child Thread : 9
Child Thread : 8
Main Thread9
Child Thread : 7
Main Thread8
Child Thread : 6
Child Thread : 5
Main Thread7
Child Thread : 4
Child Thread : 3
Main Thread6
Child Thread : 2
Child Thread : 1
Main Thread5
Exiting Child Thread
Main Thread4
Main Thread3
Main Thread2
Main Thread1
Main Thread Exiting
```

```
E:\java>
```

Result: Thus a program to demonstrate the usage of Thread class is executed successfully.

Ex. No 7
15-09-2020

Design a Calculator

Aim: To design a calculator to perform addition and division using frames.

Algorithm

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step 3: Create a class “prg7” which extends Frame and implements ActionListener interface.

Step 3.1: Define textfield t1 and three Buttons b1, b2, b3 with caption +, /, =

Step 3.2: Create a Font object; declare integer variables a,b,c and a char type variable p

Step 3.3 : Define the constructor.

Step 3.3.1: Set the Layout.

Step 3.3.2: Add TextField t1, and three buttons b1, b2, b3;

Step 3.3.3: Set the Font.

Step 3.3.4: Register ActionListener to b1, b2, b3.

Step 3.3.5: Set the title.

Step 3.3.6: Set the size.

Step 3.3.7: Set the visibility of frame as true.

Step 3.3.8: Write the window closing method.

Step 3.4: Write the code for actionPerformed() method

Step 3.4.1: If the button pressed is “+”

$a \leftarrow t1.getText()$

$p \leftarrow '+'$

Step 3.4.2: If the button pressed is “/”

$a \leftarrow t1.getText()$

$p \leftarrow '/'$

Step 3.4.3: If the button pressed is “=”

$b \leftarrow t1.getText()$

If $p = '+'$

$c \leftarrow a + b$

set c in t1

if $p = '/'$

$c \leftarrow a / b$

set c in t1

Step 3.5: Define the main () method

Step 3.5.1: Create object for class prg7

Step 5: Save the program; compile and execute.

//prg7 --- Design a Calculator

```
import java.awt.*;
import java.io.*;
import java.awt.event.*;
import java.lang.*;
public class prg7 extends Frame implements ActionListener
{
    TextField t1=new TextField("0",20);
    Button b1=new Button("+");
    Button b2=new Button("/");
    Button b3=new Button("=");
    int a,b,c;
    char p;
    Font f=new Font("Arial",Font.BOLD,25);
    prg7()
    {
        setLayout(new FlowLayout());
        add(t1);
        t1.setFont(f);
        add(b1);add(b2); add(b3);
        b1.setFont(f); b2.setFont(f); b3.setFont(f);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        setTitle("CALCULATOR");
        setSize(700,200);
        setVisible(true);
        addWindowListener(new WindowAdapter()
        { public void windowClosing(WindowEvent we)
          {
              System.exit(0);
          }
        });
    }
    public void actionPerformed(ActionEvent ae)
    {
        if (ae.getActionCommand().equals("+"))
        {
            a=Integer.parseInt(t1.getText());
            t1.setText("0");
            p='+';
        }
        if (ae.getActionCommand().equals("/"))
```

```

    {
        a=Integer.parseInt(t1.getText());
        t1.setText("0");
        p='/';
    }
    if (ae.getActionCommand().equals("="))
    {
        b=Integer.parseInt(t1.getText());
        switch(p)
        {

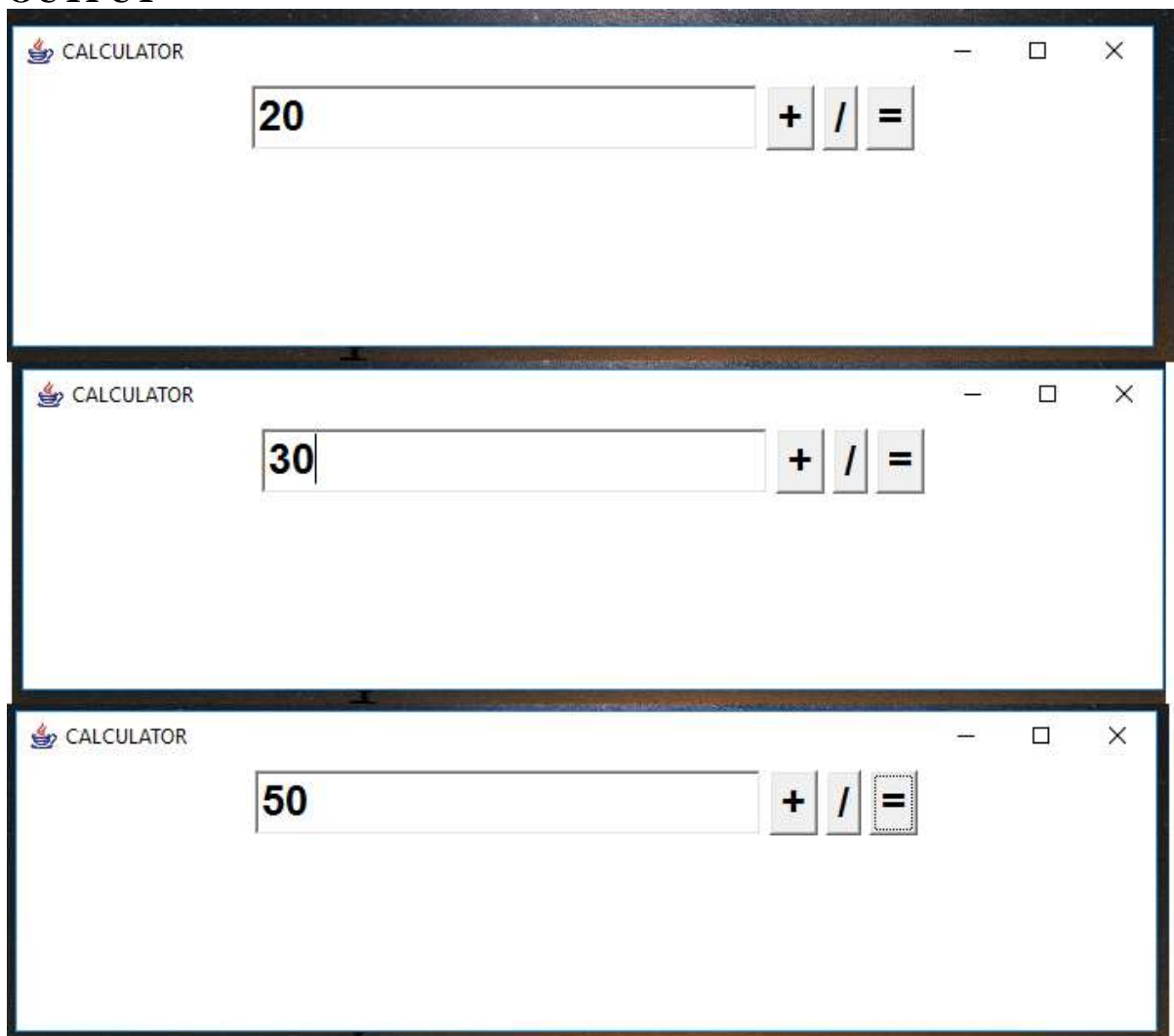
            case '+':
                c=a+b;
                t1.setText(c+" ");
                break;
            case '/':
                c=a/b;
                t1.setText(c+" ");
                break;

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

    prg7 p7=new prg7();
}
}

```


OUTPUT



Result: Thus the program to implement simple calculator is executed successfully

Ex. No 8
19-09-2020

Details of Car

Aim: To write an applet program to display the details of car.

Algorithm

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step 3 : Include the <applet> tag.

Step 4 : Create a class “prg8” which extends Applet and implements ItemListener interface.

Step 4.1: Declare four checkboxes m,z,a,e and a TextArea ta.

Step 4.2: Define the init() method.

Step 4.2.1: Create TextArea ta and Checkboxes m,z,a,e.

Step 4.2.2: Add all the Components.

Step 4.2.3: Register all the Checkboxes (m,z,a,e) to ItemListener interface by invoking addItemListener() method

Step 4.3: Define ItemStateChanged() method

Step 4.3.1: if “m” is selected, store the details of “Maruthi 800” in variable fn.

Step 4.3.2: if “z” is selected, store the details of “Zen” in variable fn.

Step 4.3.3: if “a” is selected, store the details of “Alto” in variable fn.

Step 4.3.4: if “e” is selected, store the details of “Esteem” in variable fn.

Step 4.3.5: Display the value of fn in TextArea ta.

Step 4: Save the program; compile and execute.

```

//prg8 –Details of car
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code = prg8 width = 700 height = 200>
</applet>
*/
public class prg8 extends Applet implements ItemListener
{
    String c1=" ";
    String fn=" ";
    Label l1,l2;
    CheckboxGroup cbg;
    Checkbox m,z,a,e;
    TextArea ta;
    public void init()
    {
        l1=new Label(" ");
        l2=new Label("Select your option");
        cbg=new CheckboxGroup();
        m=new Checkbox("Maruti 800",cbg,true);
        z=new Checkbox("Zen",cbg,false);
        a=new Checkbox("Alto",cbg,false);
        e=new Checkbox("Esteem",cbg,false);
        ta=new TextArea("",10,50);
        add(l1); add(l2);
        add(m); add(z); add(a); add(e);
        add(ta);
        m.addItemListener(this);
        z.addItemListener(this);
        a.addItemListener(this);
        e.addItemListener(this);
    }
    public void itemStateChanged(ItemEvent ie)
    {
        c1=cbg.getSelectedCheckbox().getLabel();
        if (c1.equals("Maruti 800"))
        {
            fn="Maruti 800 is a small city car that was manufactured \n "+
            "by Maruti Suzuki in India from 1983 to 2014.\n"+
            " The first generation (SS80) was based on the 1979 \n"+
            "Suzuki Alto and had an 800 cc F8B engine, hence the moniker.";
            l1.setText("Details of Maruti 800");
        }
    }
}

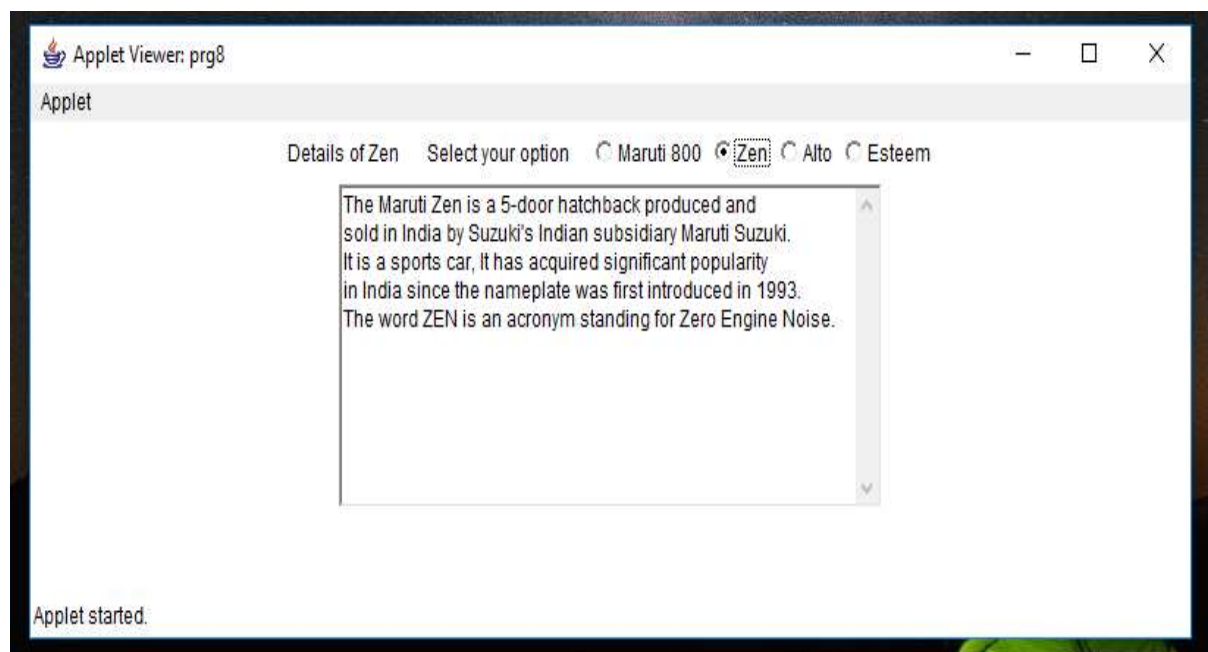
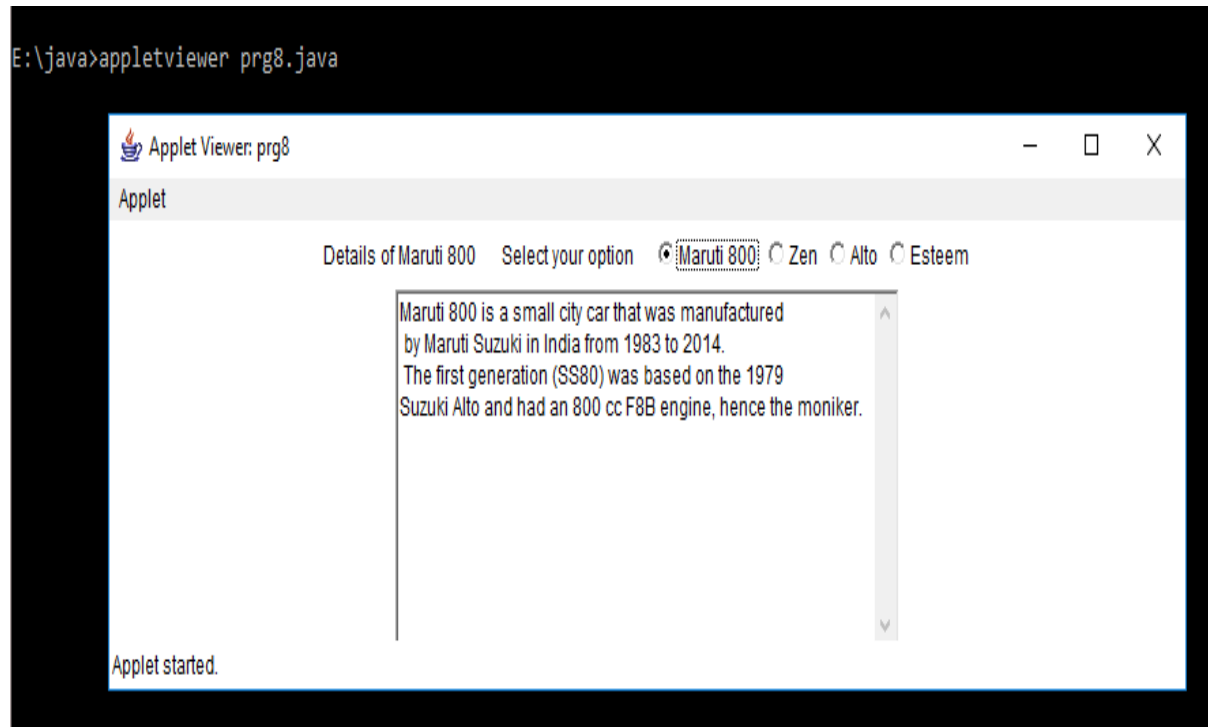
```

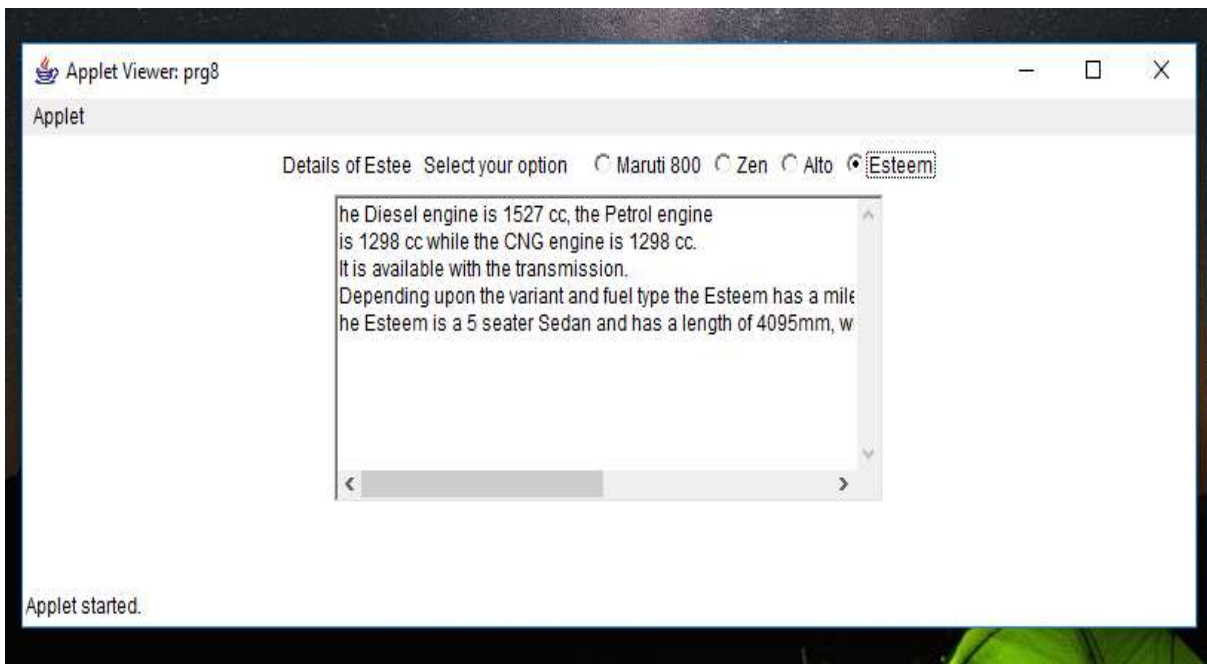
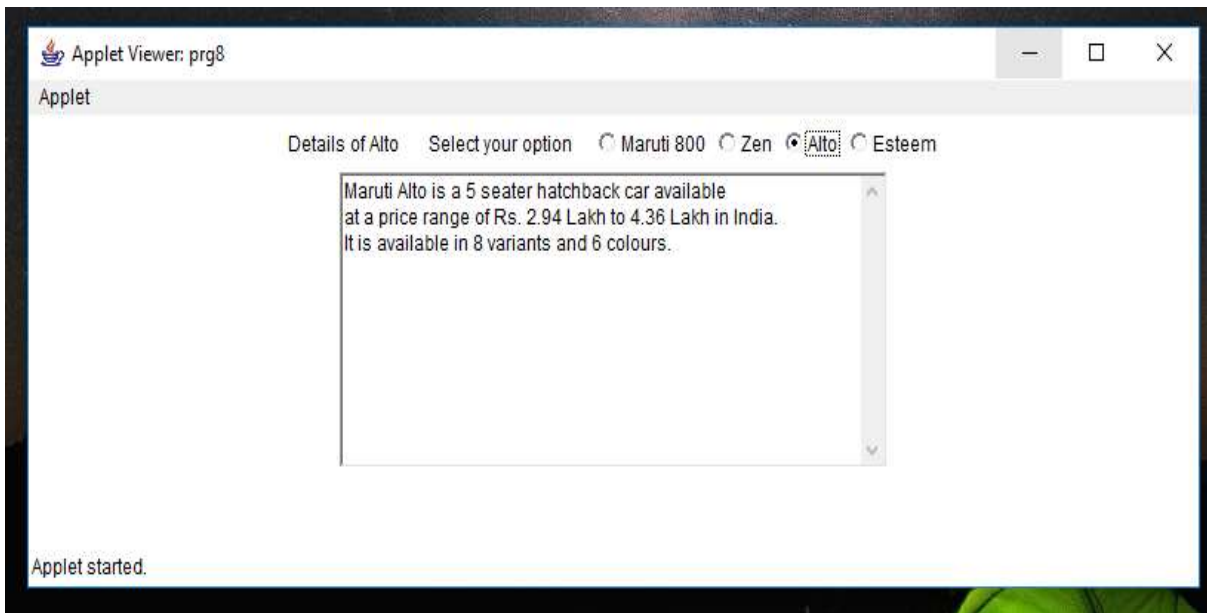
```

    }
    else if (c1.equals("Zen"))
    {
        fn="The Maruti Zen is a 5-door hatchback produced and\n"+
        "sold in India by Suzuki's Indian subsidiary Maruti Suzuki.\n"+
        "It is a sports car, It has acquired significant popularity \n"+
        "in India since the nameplate was first introduced in 1993. \n"+
        "The word ZEN is an acronym standing for Zero Engine Noise.";
        l1.setText("Details of Zen");
    }
    else if (c1.equals("Alto"))
    {
        fn="Maruti Alto is a 5 seater hatchback car available \n"+
        "at a price range of Rs. 2.94 Lakh to 4.36 Lakh in India. \n"+
        "It is available in 8 variants and 6 colours.";
        l1.setText("Details of Alto");
    }
    else if (c1.equals("Esteem"))
    {
        fn="he Diesel engine is 1527 cc, the Petrol engine \n"+
        "is 1298 cc while the CNG engine is 1298 cc. \n"+
        "It is available with the transmission. \n"+ "Depending upon the
        variant and fuel type the Esteem has a mileage of 15.9 kmpl.\n"+
        "he Esteem is a 5 seater Sedan and has a length of 4095mm, "+
        "width of 1575mm and a wheelbase of 2365mm.May 17, 2020";
        l1.setText("Details of Esteem");
    }
    ta.setText(fn);
}
}

```

OUTPUT





Result: Thus an applet program to display car details was executed successfully

Ex. No 9
22-09-2020

BACKGROUND COLOR

Aim: To change the background color of window using applet.

Algorithm

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step 3 : Include the <applet> tag.

Step 4 : Create a class “prg9” which extends Applet and implements ItemListener interface.

Step 4.1: Declare a CheckboxGroup object cbg. Declare Checkbox objects m, c, y, o, r, b, g and a label l

Step 4.2: Define the init() method.

Step 4.2.1: Define the label object l

Step 4.2.2: Define the Checkbox objects m, c, y, o, r, b, g with cbg as an argument and with labels “Magenta”, “Cyan”, “Yellow”, “Orange”, ”Red”, “Blue” and “Green”.

Step 4.2.3: Add all the components.

Step 4.2.4: Register m, c, y, o, r, b, g with ItemListener interface.

Step 4.3: Define ItemStateChanged() method

Step 4.3.1: Get the label of selected checkbox and store it in variable c1.

Step 4.3.2: if c= “Magenta”

set “Magenta” as background and “Black” as Foreground

Step 4.3.3: if c= “Cyan”

set “Cyan” as background and “Black” as Foreground

Step 4.3.4: if c= “Yellow”

set “Yellow” as background and “Black” as Foreground

Step 4.3.5: if c= “Orange”

set “Orange” as background and “Black” as Foreground

Step 4.3.6: if c= “Red”

set “Red” as background and “Black” as Foreground

Step 4.3.7: if c= “Blue”

set “Blue” as background and “Black” as Foreground

Step 4.3.8: if c= “Green”

set “Green” as background and “Black” as Foreground

Step 4.4: Define the paint() method

Step 4.5: Save the program; compile and execute.

```

//prg9 ---Background Color
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="prg9" width=300 height = 300>
</applet>*/
public class prg9 extends Applet implements ItemListener
{
    String c1=" ";
    String msg=" ";
    Label l;
    CheckboxGroup cbg;
    Checkbox m,i,y,o,r,b,g;
    public void init()
    {
        l=new Label("Select your option");
        cbg=new CheckboxGroup();
        m=new Checkbox("Magenta",cbg,true);
        i=new Checkbox("Indigo",cbg,false);
        y=new Checkbox("Yellow",cbg,false);
        o=new Checkbox("Orange",cbg,false);
        r=new Checkbox("Red",cbg,false);
        b=new Checkbox("Blue",cbg,false);
        g=new Checkbox("Green",cbg,false);
        add(l);
        add(m);
        add(i);
        add(y);
        add(o);
        add(r);
        add(b);
        add(g);
        m.addItemListener(this);
        i.addItemListener(this);
        y.addItemListener(this);
        o.addItemListener(this);
        r.addItemListener(this);
        b.addItemListener(this);
        g.addItemListener(this);
    }
    public void itemStateChanged(ItemEvent ie)
    {

```



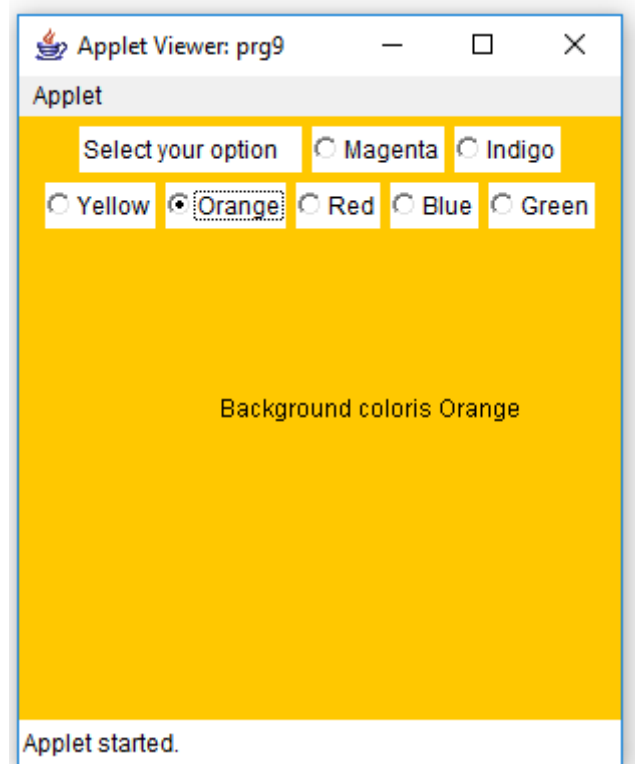
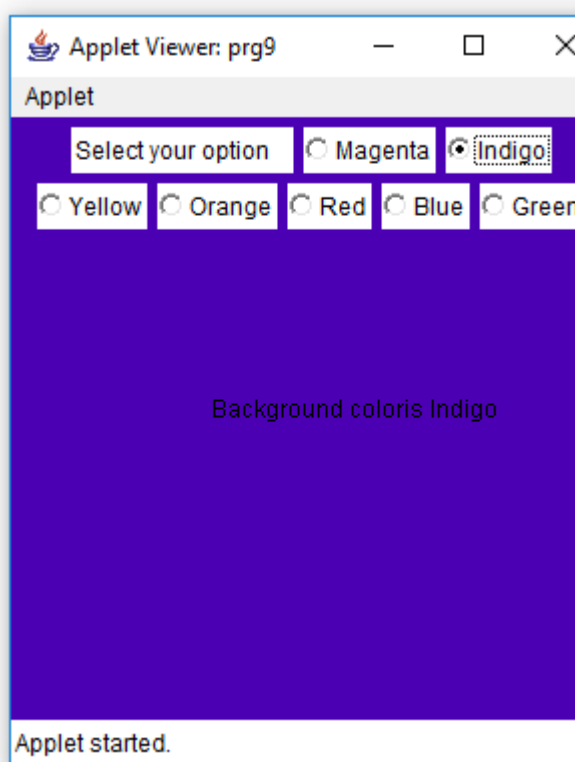
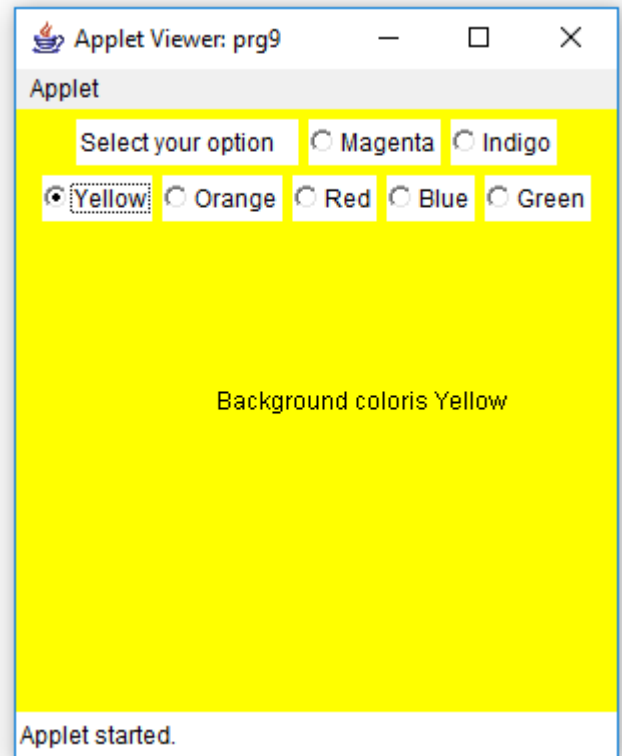
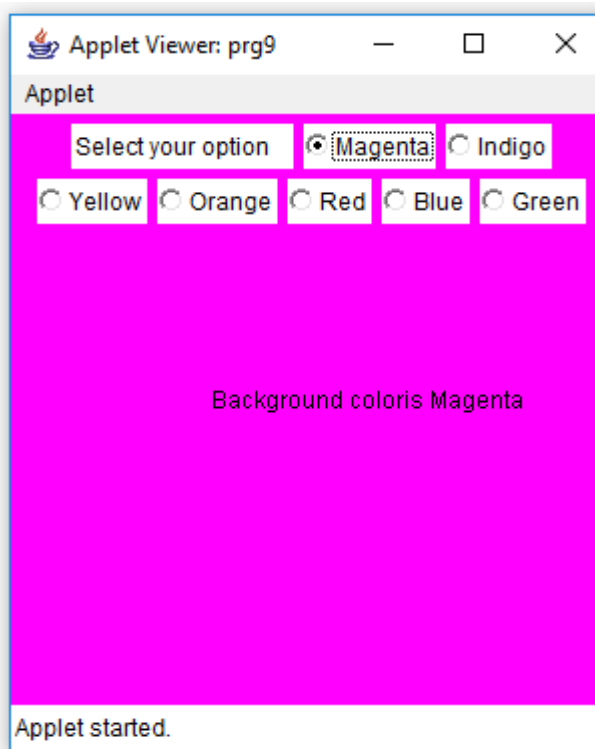
```

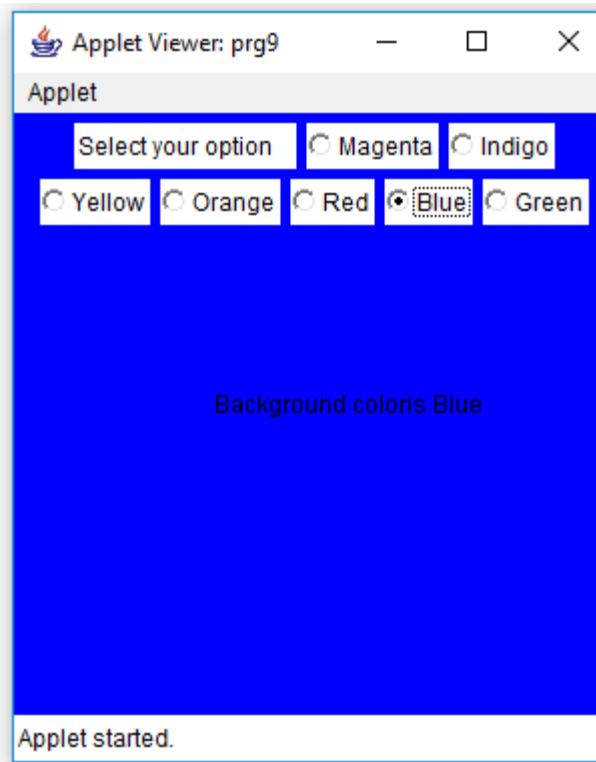
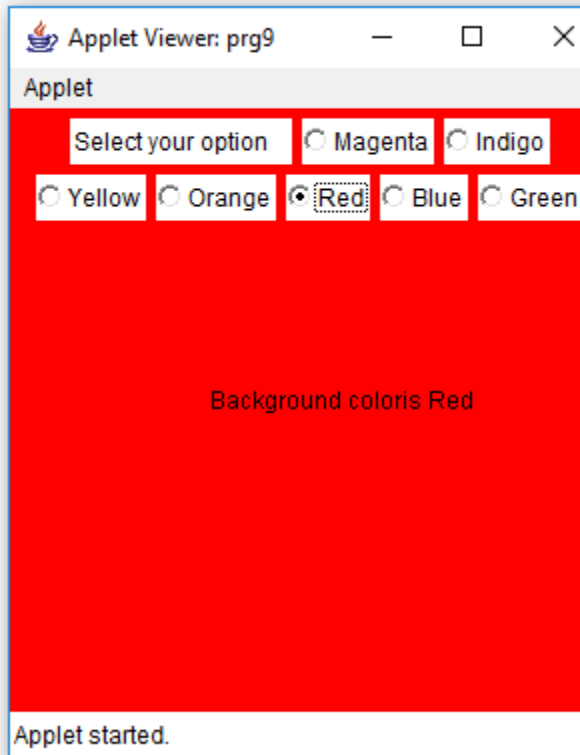
c1=cbg.getSelectedCheckbox().getLabel();
if (c1.equals("Magenta"))
{
    setBackground(Color.magenta);
    setForeground(Color.black);
    msg="Background coloris Magenta";
    repaint();
}
if (c1.equals("Indigo"))
{
    Color c= new Color(75,0,180);
    setBackground(c);
    setForeground(Color.black);
    msg="Background coloris Indigo";
    repaint();
}
if (c1.equals("Yellow"))
{
    setBackground(Color.yellow);
    setForeground(Color.black);
    msg="Background coloris Yellow";
    repaint();
}
if (c1.equals("Orange"))
{
    setBackground(Color.orange);
    setForeground(Color.black);
    msg="Background coloris Orange";
    repaint();
}
if (c1.equals("Red"))
{
    setBackground(Color.red);
    setForeground(Color.black);
    msg="Background coloris Red";
    repaint();
}
if (c1.equals("Blue"))
{
    setBackground(Color.blue);
    setForeground(Color.black);
    msg="Background coloris Blue";
    repaint();
}

```

```
    }
    if (c1.equals("Green"))
    {
        setBackground(Color.green);
        setForeground(Color.black);
        msg="Background coloris Green";
        repaint();
    }
}
public void paint(Graphics g)
{
    g.drawString(msg,100,150);
}
}
```

OUTPUT





Result: Thus an applet program to change the background color of window is executed successfully.

Ex. No: 10a
26-09-2020

NegativeArraySizeException

Aim: To Write a program to handle NegativeArraySizeException

Algorithm:

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step3 : Create a class “prg10a”.

Step 3.1: Declare the main method and declare variables.

Step 3.2: Create a try block and

Step 3.2.1: Read arraysize in variable “n”.

Step 3.2.2: Create an array with size “n”.

Step 3.2.3: If the array size is negative, then throw a NegativeArraySizeException which is handled by a catch block.

Step 3.2.4: Otherwise create an array with the give size

Step 4: Save the program; compile and execute.

```

//Prg10 a---- NegativeArraySizeException
import java.io.*;
class prg10a
{
    public static void main(String args[]) throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        try
        {
            int n;
            System.out.println("\n Enter n, the array size");
            System.out.println("\n Enter negative value to raise exception");
            n=Integer.parseInt(br.readLine());
            int a[]=new int[n];
            System.out.println("Array created with size "+n);
        }
        catch(NegativeArraySizeException ne)
        {
            System.out.println("Exception raised "+ne);
        }
    }
}

```

OUTPUT

```

E:\java>java prg10a
Enter n, the array size
Enter negative value to raise exception
3
Array created with size 3
E:\java>java prg10a
Enter n, the array size
Enter negative value to raise exception
-4
Exception raised java.lang.NegativeArraySizeException
E:\java>

```

Result: Thus a program to illustrate NegativeArraySizeException is executed successfully.

Ex. No: 10b
29-09-2020

ArrayIndexOutOfBoundsException

Aim: To Write a program to handle ArrayIndexOutOfBoundsException

Algorithm:

Step 1: Open notepad.

Step 2: Import the necessary packages.

Step3 : Create a class “prg10b”.

Step 3.1: Declare the main method and declare variables.

Step 3.2: Create a try block and

Step 3.2.1: Declare an integer array and initialize with two values”.

Step 3.2.2: Assign 99 to a[4].

Step 3.3: Write a catch block to handle ArrayIndexOutOfBoundsException

Step 4: Save the program; compile and execute.

//prg10b --- ArrayIndexOutOfBoundsException

```
import java.io.*;
class prg10b
{
    public static void main(String args[]) throws IOException
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        try
        {
            int a[]={ 10,20};
            a[42]=40;
        }
        catch(ArrayIndexOutOfBoundsException ae)
        {
            System.out.println("Exception raised "+ae);
        }
    }
}
```

OUTPUT

E:\java>javac prg10b.java

E:\java>java prg10b

Exception raised java.lang.ArrayIndexOutOfBoundsException: 42

E:\java>

Result: Thus a program to illustrate ArrayIndexOutOfBoundsException is executed successfully.