

Q. 1

Programming Lab

PART A: Java Fundamentals OOPs in Java

1. Program to assign two integer values to X and Y. Using the 'if' statement the output of the program should display a message whether X is greater than Y.
2. Program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop. (Hint Fact of 4 = 4*3*2*1)
3. Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.
4. Program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide. A main function should access the methods and perform the mathematical operations.
5. Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.
6. Program
 - a. To find the area and circumference of the circle by accepting the radius from the user.
 - b. To accept a number and find whether the number is Prime or not
7. Program to create a student class with following attributes; Enrollment No: Name, Mark of sub1, Mark of sub2, mark of sub3, Total Marks. Total of the three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50. If a candidate fails in any one of the subjects his total mark must be declared as zero. Using this condition write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.
8. In a college first year class are having the following attributes Name of the class (BCA, BCom, BSc), Name of the staff No of the students in the class, Array of students in the class

9. Define a class called first year with above attributes and define a suitable constructor. Also write a method called best Student () which process a first-year object and return the student with the highest total mark. In the main method define a first-year object and find the best student of this class
10. Program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment. ie, print them as per their seniority.
11. Create a package 'student. Fulltime. BCA ' in your current working directory
- Create a default class student in the above package with the following attributes: Name, age, sex.
 - Have methods for storing as well as displaying

PART B: Exception Handling & GUI Programming

- Program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.
- Program to handle Null Pointer Exception and use the "finally" method to display a message to the user.
- Program which create and displays a message on the window
- Program to draw several shapes in the created window
- Program to create an applet and draw grid lines
- Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother similar details of mother also appear.
- Create a frame which displays your personal details with respect to a button click
- Create a simple applet which reveals the personal information of yours.
- Program to move different shapes according to the arrow key pressed.
- Program to create a window when we press M or m the window displays Good Morning, A or a the window displays Good After Noon E or e the window displays Good Evening, N or n the window displays Good Night
- Demonstrate the various mouse handling events using suitable example.
- Program to create menu bar and pull-down menus.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

JAVA LAB -II SEM

Part A

- 1. Program to assign two integer values to X and Y. Using the 'if' statement the output of the program should display a message whether X is greater than Y.**

```
import java.util.Scanner;
public class GreaterNumber{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int x, y;
        System.out.println("Enter two numbers: ");
        x = sc.nextInt();
        y = sc.nextInt();
        if(x > y)
            System.out.println(x + " is greater than " + y);
        else
            System.out.println(x + " is lesser than " + y);
    }
}
```

Output -

Enter two numbers: 3 2

3 is greater than 2

2. Program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop. (Hint Fact of 4 = 4*3*2*1)

```
public class Factorial{  
    public static void main(String args[]){  
        int i, n, fact;  
        for(i = 1; i <=10; i++)  
        {  
            fact = 1;  
            n = i;  
            while(n > 1)  
            {  
                fact *= n;  
                n--;  
            }  
            System.out.println("Factorial of " + i + " = " + fact);  
        }  
    }  
}
```

Output -

Factorial of 1 = 1
Factorial of 2 = 2
Factorial of 3 = 6
Factorial of 4 = 24
Factorial of 5 = 120
Factorial of 6 = 720
Factorial of 7 = 5040
Factorial of 8 = 40320
Factorial of 9 = 362880
Factorial of 10 = 3628800

3. Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.

```
public class Add{  
    void sum(int x, int y)  
    {  
        int s = x + y;  
        System.out.println("Sum = " + s);  
    }  
    void sum(float x, float y)  
    {  
        float s = x + y;  
        System.out.println("Sum = " + s);  
    }  
    void sum()  
    {  
        int x = 5;  
        int y = 10;  
        int s = x + y;  
        System.out.println("Sum = " + s);  
    }  
    public static void main(String args[])  
    {  
        Add a = new Add();  
        a.sum(1, 2);  
        a.sum(1.1f, 2.2f);  
        a.sum();  
    }  
}
```

Output -

Sum = 3
Sum = 3.3000002
Sum = 15

4.Program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the Superclass. MulDiv should have methods to multiply and divide .A main function should access the methods and perform the mathematical operations.(File name: MulDiv.java)

```
class AddSub //Superclass
{
    int z;
    public void add(int x,int y) // Addition method
    {
        z = x+y;
        System.out.println("The sum of the given numbers:"+z);
    }
    public void sub(int x,int y) // Subtraction method
    {
        z = x-y;
        System.out.println("The difference of the given numbers:"+z);
    }
}

public class MulDiv extends AddSub // Muldiv subclass--Inheritance
{
    public void multiply(int x,int y)// Multiplication method
    {
        z= x*y;
        System.out.println("The product of the given numbers:"+z);
    }
    public void divide(int x,int y)// Division Method
    {
        z= x/y;
        System.out.println("The quotient for the given numbers:"+z);
    }
}
```

```
public static void main(String args[])
{
    int a = 20,b=10;
    MulDiv md = new MulDiv();
    md.add(a,b);
    md.sub(a,b);
    md.multiply(a,b);
    md.divide(a,b);
}
```

Output:

```
Command Prompt
C:\JAVA\LAB>javac MulDiv.java
C:\JAVA\LAB>java MulDiv
The sum of the given numbers:30
The difference of the given numbers:10
The product of the given numbers:200
The quotient for the given numbers:2
```

5. Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.

```
class Student{
    int roll;
    String name;
    static String college;

    Student(int r, String n, String c){
        roll = r;
        name = n;
        college = c;
    }
    void display()
    {
        System.out.println(roll + " " + name + " " + college);
    }
}
class StaticExample{
    public static void main(String args[]){
        Student s1 = new Student(1, "s1", "sample1");
        s1.display();
        Student s2 = new Student(2, "s2", "sample2");
    }
}
```

```
        s2.display();
        s1.display();
    }
}
```

Output-

```
1 s1 sample1
2 s2 sample2
1 s1 sample2
```

6. Program

- a) To find the area and circumference of the circle by accepting the radius from the user.

```
import java.util.Scanner;
class A06aCircle{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the radius: ");
        float r = sc.nextFloat();
        float a = 3.14f * r * r;
        float c = 2 * 3.14f * r;
        System.out.println("Area = " + a);
        System.out.println("Circumference = " + c);
    }
}
```

Output-

```
Enter the radius: 3
Area = 28.26
Circumference = 18.84
```

b) To accept a number and find whether the number is Prime or not.

```
import java.util.Scanner;
class A06bPrime{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int i, c = 0;
        System.out.print("Enter the number: ");
        int n = sc.nextInt();
        if(n <= 1)
            System.out.println("It is not a prime number");
        else
        {
            for ( i = 2; i < n; i++)
            {
                if (n % i == 0)
                    c++;
            }
            if (c == 0)
                System.out.println("It is a prime number.");
            else
                System.out.println("It is not a prime number.");
        }
    }
}
```

Output -

Enter the number: 3

It is a prime number.

Enter the number : 2

It is not a prime number.

7. Program to create a student class with following attributes; Enrollment No: Name, Mark of sub1, Mark of sub2, mark of sub3, Total Marks. Total of the three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50. If a candidate fails in any one of the subjects his total mark must be declared as zero. Using this condition, write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.

```
import java.util.Scanner;
public class A07Constructor{
    public static void main(String args[]){
        StudentInfo stu[] = new StudentInfo[3];
        for(int i = 0; i < 3; i++)
            stu[i] = new StudentInfo();
        System.out.println("\t\tStudent Details");
        System.out.println("Enrollment
no.\t\tName\t\tTotal");
        for(int i = 0; i < 3; i++)
            stu[i].displayInfo();
    }
}
class StudentInfo{
    Scanner sc = new Scanner(System.in);
    String eid;
    String name;
    int s1, s2, s3, total;
    StudentInfo(){
        readStudentInfo();
    }
    public void readStudentInfo(){
        System.out.println("Enter student details");
        System.out.println("Enrollment no: ");
        eid = sc.next();
        System.out.println("Name: ");
        name = sc.next();
        System.out.println("Enter marks of 3 subjects: ");
        s1 = sc.nextInt();
        s2 = sc.nextInt();
        s3 = sc.nextInt();
        if(s1 >= 50 && s2 >= 50 && s3 >= 50)
            total = s1 + s2 + s3;
        else
            total = 0;
    }
    public void displayInfo(){
        System.out.println(eid + "\t\t\t" + name + "\t\t\t" +
total);
    }
}
```

Output -

```
Enter student details
Enrollment no: 1
Name: Geetha
Enter marks of 3 subjects: 99 99 99
Enter student details
Enrollment no: 2
Name: Mohan
Enter marks of 3 subjects: 1 1 1
Enter student details
Enrollment no: 3
Name: Chaitra
Enter marks of 3 subjects: 12 5 1
    Student Details
Enrollment no.      Name      Total
1                  Geeta     297
2                  Mohan      0
3                  Chaitra   0
```

/*

Program-8: In a college first year class are having the following attributes

- Name of the class (BCA, BCom, BSc),**
- Name of the staff (Class Teacher Name)**
- No of the students in the class,**
- Array of students in the class.**

Display Student Information

*/

```
import java.util.Scanner;
public class FirstYearClassRoom {
    String className;
    String classTeacherName;
    int studentCount;
    String studentName[] = new String[50]; // Student Name Array

    Scanner sc = new Scanner(System.in);

    // Below is constructor function that is called when you create object of this class
    public FirstYearClassRoom(){
        getInfo();
    }

    private void getInfo()
```

```
{  
    System.out.println("Please Enter the Class Name");  
    className = sc.nextLine();  
  
    System.out.println("Please Enter Class Teacher Name");  
    classTeacherName = sc.nextLine();  
  
    System.out.println("Please enter total number of students in the class");  
    studentCount = Integer.parseInt(sc.nextLine());  
  
    System.out.println("Please enter names of all " + studentCount + " students in the  
class");  
    for ( int i =0; i < studentCount ; i ++){  
        studentName[i] = sc.nextLine();  
    }  
}  
  
public void display()  
{  
    System.out.println("Class Name: " + className);  
    System.out.println("Class Teacher Name: " + classTeacherName);  
    System.out.println(" Student Names ");  
    System.out.println(" ----- ");  
    for ( int i =0; i < studentCount ; i ++){  
        System.out.println(studentName[i]);  
    }  
}  
  
public static void main (String args[]){  
    FirstYearClassRoom fy = new FirstYearClassRoom(); // this will trigger getinfo  
function  
    fy.display();  
}  
}
```

```
C:\JAVA\LAB>javac FirstYearClassRoom.java

C:\JAVA\LAB>java FirstYearClassRoom
Please Enter the Class Name
BCA
Please Enter Class Teacher Name
Sudha
Please enter total number of students in the class
3
Please enter names of all 3 students in the class
Krishna
Geetha
Arjun
Class Name: BCA
Class Teacher Name: Sudha
Student Names
-----
Krishna
Geetha
Arjun
```

/*

Program-9: Define a class called first year with above attributes and define a suitable constructor.

Also write a method called best Student () which process a first-year object and return the student with the highest total mark. In the main method define a first-year object and find the best student of this class

*/

```
import java.util.Scanner;
public class FirstYearClassBestMarks {

    String className;
    String classTeacherName;
    int studentCount;
    String studentName[] = new String[50]; // Student Name Array
    int studentMarks[] = new int[50]; // Student Marks Array
    Scanner sc = new Scanner(System.in);
    // Below is constructor function that is called when you create object of this class
    public FirstYearClassBestMarks(){
        getInfo();
    }
    private void getInfo(){

        System.out.println("Please Enter the Class Name");
        className = sc.nextLine();

        System.out.println("Please Enter Class Teacher Name");
```

```
classTeacherName = sc.nextLine();

System.out.println("Please enter total number of students in the class");
studentCount = Integer.parseInt(sc.nextLine());

System.out.println("Please enter names of all " + studentCount + " students in the
class");
for ( int i =0; i < studentCount ; i ++){
    studentName[i] = sc.nextLine();
}
System.out.println("Please start entering marks for students :");
for ( int i =0; i < studentCount ; i ++){
    System.out.print("Enter marks for " + studentName[i] + " = ");
    studentMarks[i] =sc.nextInt();
    System.out.println(); // go to next line
}

public void display(){
    System.out.println("Class and Student Info Display");
    System.out.println("Class Name: " + className);
    System.out.println("Class Teacher Name: " + classTeacherName);
    System.out.println(" Student Names Marks");
    System.out.println(" -----");
    for ( int i =0; i < studentCount ; i ++){

        System.out.println(studentName[i] + " " + studentMarks[i]);
    }
}

public void bestStudent(){
    int best = 0; // variable to keep track of best marks till now when looping
    int k = -1 ; // variable to keep track of best student-index. Note that since index start
from 0, initialize this to -1.

    // loop through all student marks and pick best marks.
    for(int i=0; i <= studentCount ; i++){

        if( studentMarks[i] > best ){
            best = studentMarks[i];
            k = i; // keep track of index. As student at this index is best marks.
        }
    }
}
```

```

}

// once out of loop you will have student who got best marks. student index is k.
System.out.println("The best student is - " + studentName[k]);
System.out.println("He/She scored marks which was highest - " + studentMarks[k]);
}
public static void main (String args[]){
    FirstYearClassBestMarks fyBestMarks = new FirstYearClassBestMarks(); // this will
trigger getinfo function
    fyBestMarks.display();
    fyBestMarks.bestStudent();
}
}

```

Command Prompt

```

C:\JAVA\LAB>javac FirstYearClassBestMarks.java

C:\JAVA\LAB>java FirstYearClassBestMarks
Please Enter the Class Name
BCA
Please Enter Class Teacher Name
Deepa
Please enter total number of students in the class
3
Please enter names of all 3 students in the class
Krishna
Arjun
Mythili
Please start entering marks for students :
Enter marks for Krishna = 100

Enter marks for Arjun = 95

Enter marks for Mythili = 85

```



Output:

```

Class and Student Info Display
Class Name: BCA
Class Teacher Name: Deepa
Student Names      Marks
-----
Krishna      100
Arjun        95
Mythili      85
The best student is -Krishna
He/She scored marks which was highest - 100

```

/* Program 10: Program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment. ie, print them as per their seniority */

```
import java.util.Date;
```

```

public class Employee {
    String employeeName;
    Date appointmentDate;

    //Constructor
    public Employee(String name, Date aptDate){
        employeeName = name;
        appointmentDate = aptDate;
    }

    public static void main (String args[]){
        Employee emp[] = new Employee[10]; // array of 10 employee objects

        emp[0] = new Employee("Ritika", new Date(2000,1,25));
        emp[1] = new Employee("Rohit", new Date(1999,1,25));
        emp[2] = new Employee("Himesh", new Date(2010,1,25));
        emp[3] = new Employee("Gajendra", new Date(2022,1,15));
        emp[4] = new Employee("Shilpa", new Date(2008,1,25));
        emp[5] = new Employee("Deepa", new Date(2001,1,25));
        emp[6] = new Employee("Nandini", new Date(2012,1,25));
        emp[7] = new Employee("Ramesh", new Date(2010,1,28));
        emp[8] = new Employee("Naseer", new Date(2011,1,25));
        emp[9] = new Employee("Swetha", new Date(1990,1,25));

        System.out.println("Display Employee List");
        for(int i=0 ; i < emp.length; i++){
            System.out.println( "Employee Name: " + emp[i].employeeName +
                " Appointment date: " + emp[i].appointmentDate.getDate() + "/" +
                emp[i].appointmentDate.getMonth() + "/" + emp[i].appointmentDate.getYear() );
        }
        // Do sorting. Sort based on date.

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

            for (int j = i + 1; j < emp.length; j++) {
                if (emp[i].appointmentDate.after(emp[j].appointmentDate)) {
                    //swap
                    Employee temp = emp[i];
                    emp[i] = emp[j];
                    emp[j] = temp;
                }
            }
        }
    }
}

```

```
        emp[j] = temp;
    }
}
System.out.println("-----");
System.out.println("Display Sorted Employee List as per Seniority ");
for(int i=0 ; i < emp.length; i++){
    System.out.println( "Employee Name: " + emp[i].employeeName +
        " Appointment date: " + emp[i].appointmentDate.getDate() + "/" +
        emp[i].appointmentDate.getMonth() + "/" + emp[i].appointmentDate.getYear() );
}
} // main end
}
```

```
C:\JAVA\LAB>java Employee
Display Employee List
Employee Name: Ritika Appointment date: 25/1/2000
Employee Name: Rohit Appointment date: 25/1/1999
Employee Name: Himesh Appointment date: 25/1/2010
Employee Name: Gajendra Appointment date: 15/1/2022
Employee Name: Shilpa Appointment date: 25/1/2008
Employee Name: Deepa Appointment date: 25/1/2001
Employee Name: Nandini Appointment date: 25/1/2012
Employee Name: Ramesh Appointment date: 28/1/2010
Employee Name: Naseer Appointment date: 25/1/2011
Employee Name: Swetha Appointment date: 25/1/1990
-----
Display Sorted Employee List as per Seniority
Employee Name: Swetha Appointment date: 25/1/1990
Employee Name: Rohit Appointment date: 25/1/1999
Employee Name: Ritika Appointment date: 25/1/2000
Employee Name: Deepa Appointment date: 25/1/2001
Employee Name: Shilpa Appointment date: 25/1/2008
Employee Name: Himesh Appointment date: 25/1/2010
Employee Name: Ramesh Appointment date: 28/1/2010
Employee Name: Naseer Appointment date: 25/1/2011
Employee Name: Nandini Appointment date: 25/1/2012
```

11. Create a package ‘student. Fulltime.BCA’ in your current working directory.
- Create a default class student in the above package with the following attributes; Name, age, sex.
 - Have methods for storing as well as displaying.

Program:

Step 1: The source file should be named *student.java* and stored in the subdirectory *student_Fulltime_BCA*. The resultant *student.class* will be stored in the same subdirectory.

```
package student_Fulltime_BCA;  
public class student  
{  
    String name;  
    int age;  
    String sex;  
    public void input()  
    {  
        name="Ashwini";  
        age=20;  
        sex="Female";
```

```
    }
    public void display()
    {
        System.out.println("Name of the Student:" + name);
        System.out.println("Age:" + age);
        System.out.println("Gender:" + sex);
    }
}
```

Step 2: The following code (packagetest.java) shows the importing *student* class from the *student_Fulltime_BCA* package.

```
// Importing package
import student_Fulltime_BCA.*;
class packagetest
{
    public static void main (String arg[])
    {
        student ob=new student();
        ob.input();
        ob.display();
    }
}
```

Output:

```
Name of the Student: Ashwini
Age:20
Gender: Female
```

JAVA LAB -Part B(II sem)

- 1. Program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.**

```
class B01NegativeSize{
    public static void main(String args[]){
        try{
            int[] a = new int[-5];
        }
        catch(NegativeArraySizeException e){
            System.out.println("Negative Array Size!!");
        }
        System.out.println("Continuing execution... ");
    }
}
```

Output -

Negative Array Size!!
Continuing execution...

- 2. Program to handle Null Pointer Exception and use the “finally” method to display a message to the user.**

```
public class B02NullPointer{
    public static void main(String args[]){
        String str = null;
        //Try-catch-finally blocks
        try{
            // Below code will give null pointer exception as we are
access null str
            // Note that string is a character array. And we are
trying to access the 1st character of this string.
            System.out.println("First character of str is:" +
str.charAt(0));
        }
        catch(NullPointerException e){
            System.out.println("NullPointerException Caught in catch
block...");
```

```
        }
    finally{
        System.out.println("Finally is executed always");
    }
}
```

Output-

NullPointerException Caught in catch block...
Finally is executed always

3. Program which creates and displays a message on the window.

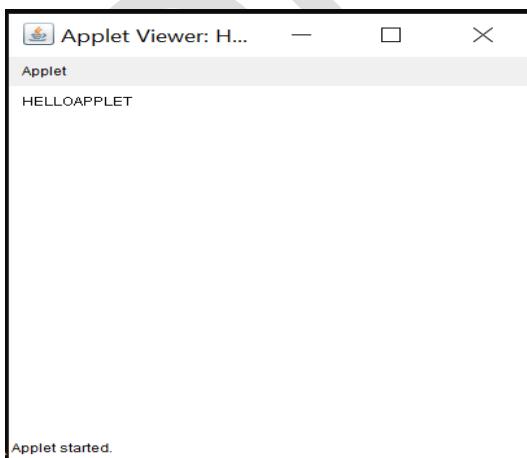
HELLOAPPLET.JAVA:-

```
import java.applet.*;
import java.awt.*;
public class HelloApplet extends Applet
{
public void paint(Graphics g)
{
g.drawString("HELLOAPPLET",10,20);
}
}
```

a.html:-

```
<html>
<applet code="HelloApplet.class" width=400 height=400>
</applet> </html>
```

Output -



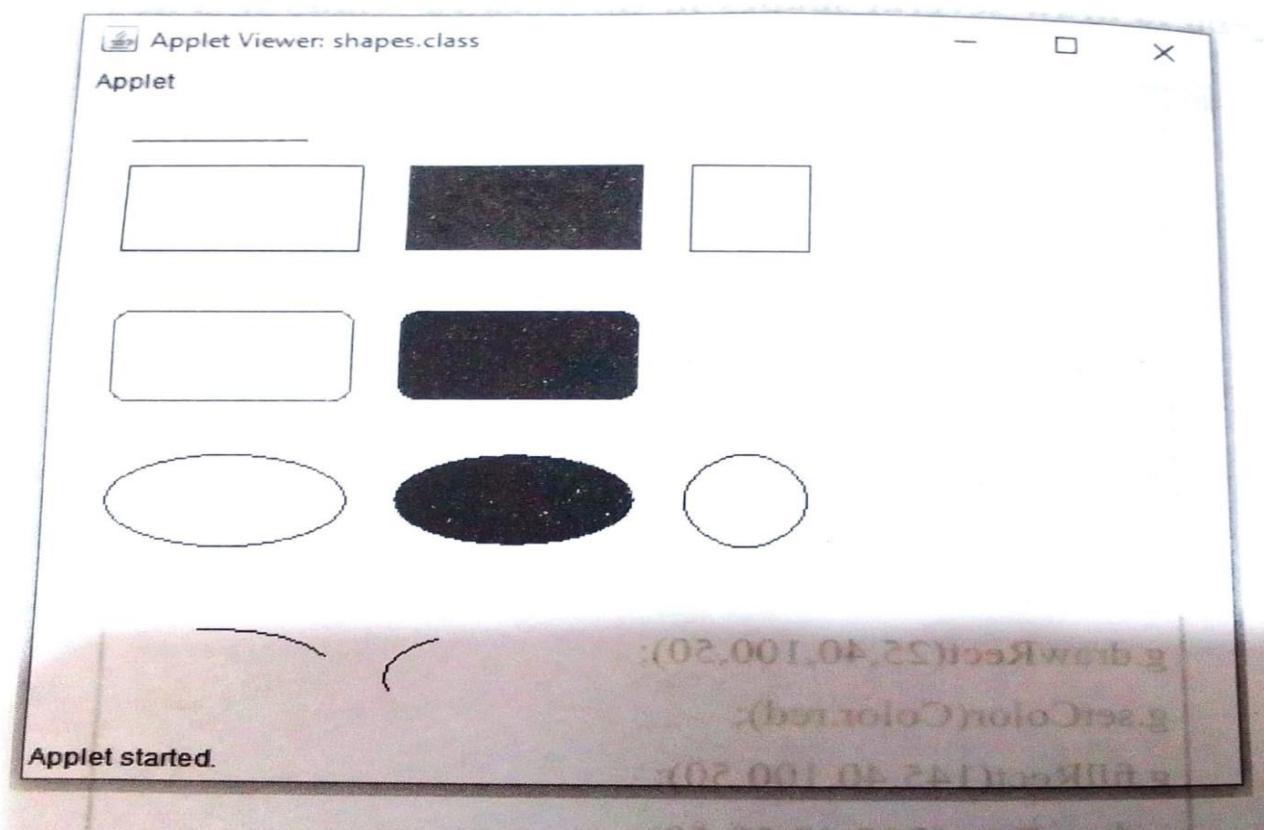
4. Program to draw several shapes in the created window

✓ **Program to draw several shapes in the created window. (File Name: shapes.java)**

program:

```
// Demonstrate to draw Shapes  
import java.awt.*;  
import java.applet.*;  
  
public class shapes extends Applet  
{  
    public void paint(Graphics g)  
    {  
        g.drawLine(25,25,100,25);  
        g.drawRect(25,40,100,50);  
        g.setColor(Color.red);  
        g.fillRect(145,40,100,50);  
        g.drawRect(265,40,50,50);  
        g.drawRoundRect(25,125,100,50,15,15);  
        g.setColor(Color.blue);  
        g.fillRoundRect(145,125,100,50,15,15);  
        g.drawOval(25,205,100,50);  
        g.fillOval(145,205,100,50);  
        g.drawOval(265,205,50,50);  
        g.drawArc(25,300,100,50,25,75);  
        g.drawArc(145,300,100,50,125,75);  
    }  
}
```

Output:



5. Write a program to create an applet and grid lines.

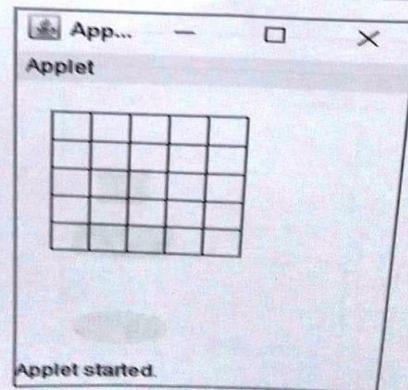
Program 3 Write a program to create an applet and draw grid lines

```
import java.awt.*;
import java.applet.*;

public class Grid extends Applet {
    public void paint(Graphics g) {
        int row, column, x, y = 20;
        // for every row
        for (row = 1; row < 5; row++) {
            x = 20;
            // for every column
            for (column = 1; column < 5; column++) {
                g.drawRect(x, y, 40, 40);
                x = x + 20;
            }
            y = y + 20;
        }
    }
}

/*
* <applet code = "Grid.class" height = 500 width =500> </applet>
*/
```

Output



6./*Program -Buttons */

Program 6

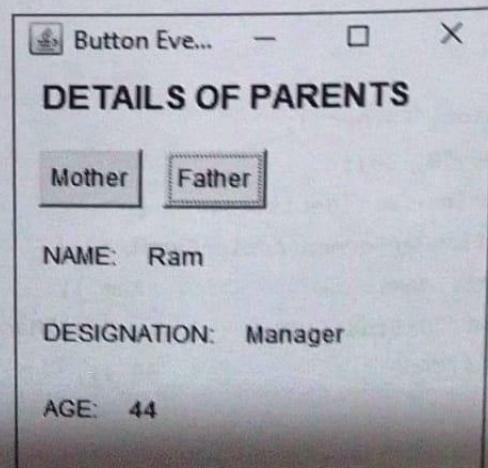
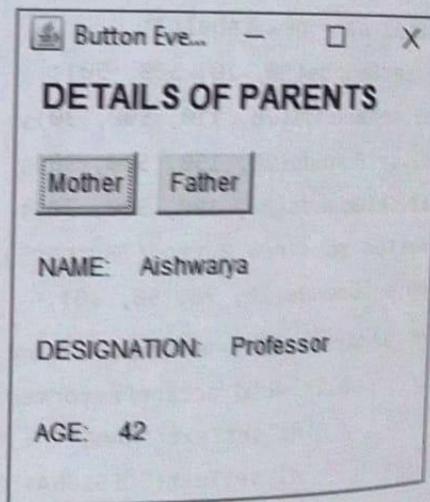
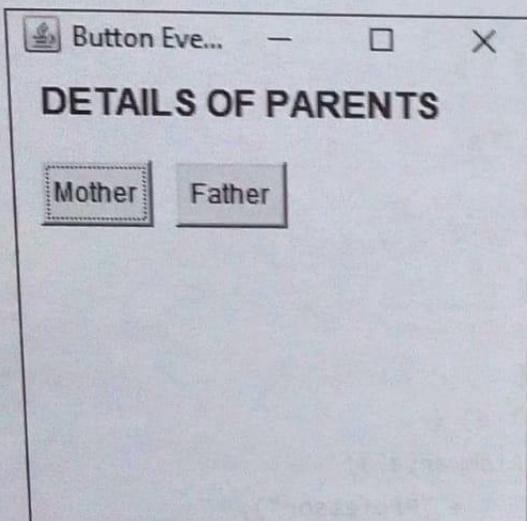
Write a program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother similar details of mother also appear.

```
import java.awt.*;
import java.awt.event.*;

public class ButtonClickActionEvents {
    public static void main(String[] args) {
        Frame f = new Frame("Button Event");
        Label l = new Label("DETAILS OF PARENTS");
        l.setFont(new Font("Calibri", Font.BOLD, 16));
        Label nl = new Label();
        Label dl = new Label();
        Label al = new Label();
        l.setBounds(20, 20, 500, 50);
        nl.setBounds(20, 110, 500, 30);
        dl.setBounds(20, 150, 500, 30);
        al.setBounds(20, 190, 500, 30);
        Button mb = new Button("Mother");
        mb.setBounds(20, 70, 50, 30);
        mb.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                nl.setText("NAME:" + " " + "Aishwarya");
                dl.setText("DESIGNATION:" + " " + "Professor");
                al.setText("AGE:" + " " + "42");
            }
        });
        Button fb = new Button("Father");
        fb.setBounds(80, 70, 50, 30);
        fb.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                nl.setText("NAME:" + " " + "Ram");
                dl.setText("DESIGNATION:" + " " + "Manager");
                al.setText("AGE:" + " " + "44");
            }
        });
    }
}
```

```
// adding elements to the frame  
f.add(mb);  
f.add(fb);  
f.add(l);  
f.add(nl);  
f.add(dl);  
f.add(al);  
// setting size, layout and visibility  
f.setSize(250, 250);  
f.setLayout(null);  
f.setVisible(true);  
}
```

Output



7./* Program 7—personal details with button click */

7. Create a frame which displays your personal details with respect to a button click. (File Name: personal.java)

Program:

```
//Personal Details  
import java.awt.*;  
import java.awt.event.*;  
public class personal extends Frame  
{  
    Button p, close;  
    Label l1,l2,l3,l4,l5,l6;;  
    personal()  
    {  
        setFont(new Font("Arial", Font.BOLD, 20));  
        p = new Button("Personal Details");  
        close = new Button ("Close");  
        l1 = new Label ("");  
        l2 = new Label ("");  
        l3 = new Label ("");
```

```
14 = new Label ("");
15 = new Label ("");
16 = new Label ("");
setLayout (new GridLayout(8,1));
setSize (500,500);
add(p);
add(close);
add(l1);
add(l2);
add(l3);
add(l4);
add(l5);
add(l6);
setVisible(true);
ButtonHandler bh = new ButtonHandler();
p.addActionListener(bh);
close.addActionListener(bh);
}
class ButtonHandler implements ActionListener
{
public void actionPerformed(ActionEvent e)
{
if (e.getSource() == p)
{
l1.setText("Name : CHARLES");
l2.setText("Father Name : James");
l3.setText("Date of Birth : 02/03/1995");
l4.setText("Address : No.50, MG Road, Kolar-563101");
l5.setText("Mobile Number : 9123456789");
l6.setText("Email : charless1995@gmail.com");
}
}
```

```
}

if (e.getSource() == close)

{

System.exit(0);

}

}

public static void main(String args[])

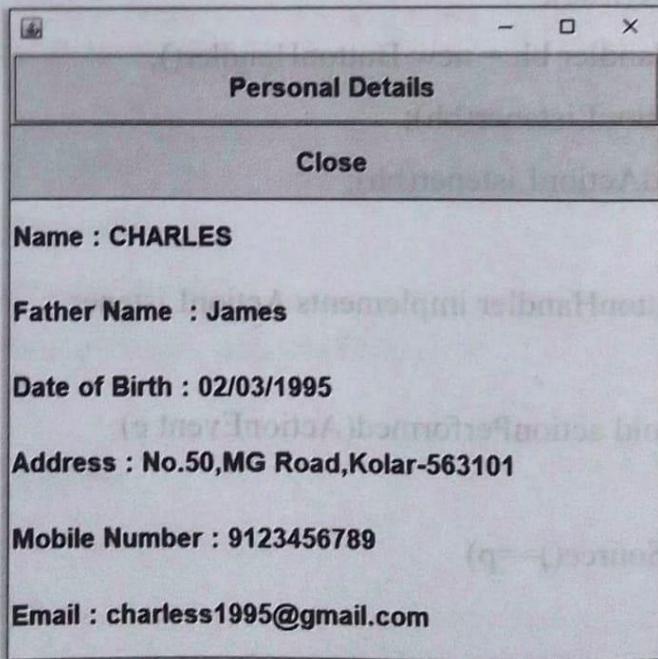
{

new personal();

}

}
```

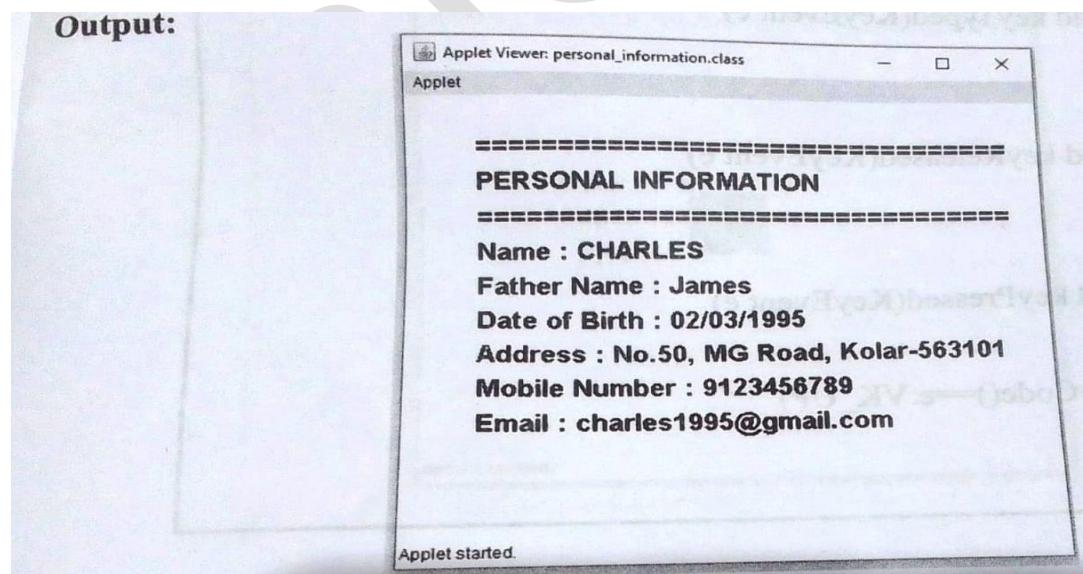
Output:



8. Create a simple applet which reveals the personal information of yours

```
import java.awt.*;
import java.applet.*;
public class Personal extends Applet{
    Font f;
    public void init(){
        setSize(500,500);
        f=new Font ("Arial",Font.BOLD,20);
    }
    public void paint(Graphics g){
        g.drawString("=====", 50, 50);
        g.drawString("Personal Information", 50, 80);
        g.drawString("=====", 50, 110);
        g.drawString("Name:Charles", 50, 140);
        g.drawString("Father Name:James", 50, 170);
        g.drawString("Date of Birth:02/03/1995", 50, 200);
        g.drawString("Address No:50, MG Road,Kolar", 50, 230);
        g.drawString("Mobile Number:9123456789", 50, 260);
        g.drawString("E-mail:charles1995@gmail.com", 50, 290);
    }
}
/* <applet code="Personal.class" height=300 width=500> </applet> */
```

Output -



9./* Program 9- Move shapes Keyboard events*/**Program 9**

Write a program to move different shapes according to the arrow key pressed.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

/*
<applet code="ArrowKeys" Width=400 height=400>
</applet>
*/
public class ArrowKeys extends Applet implements KeyListener {

    int x1 = 100, y1 = 50, x2 = 250, y2 = 200;

    public void init() {
        addKeyListener(this);
    }

    public void keyPressed(KeyEvent ke) {
        showStatus("KeyDown");
        int key = ke.getKeyCode();
```

NEW HORIZON

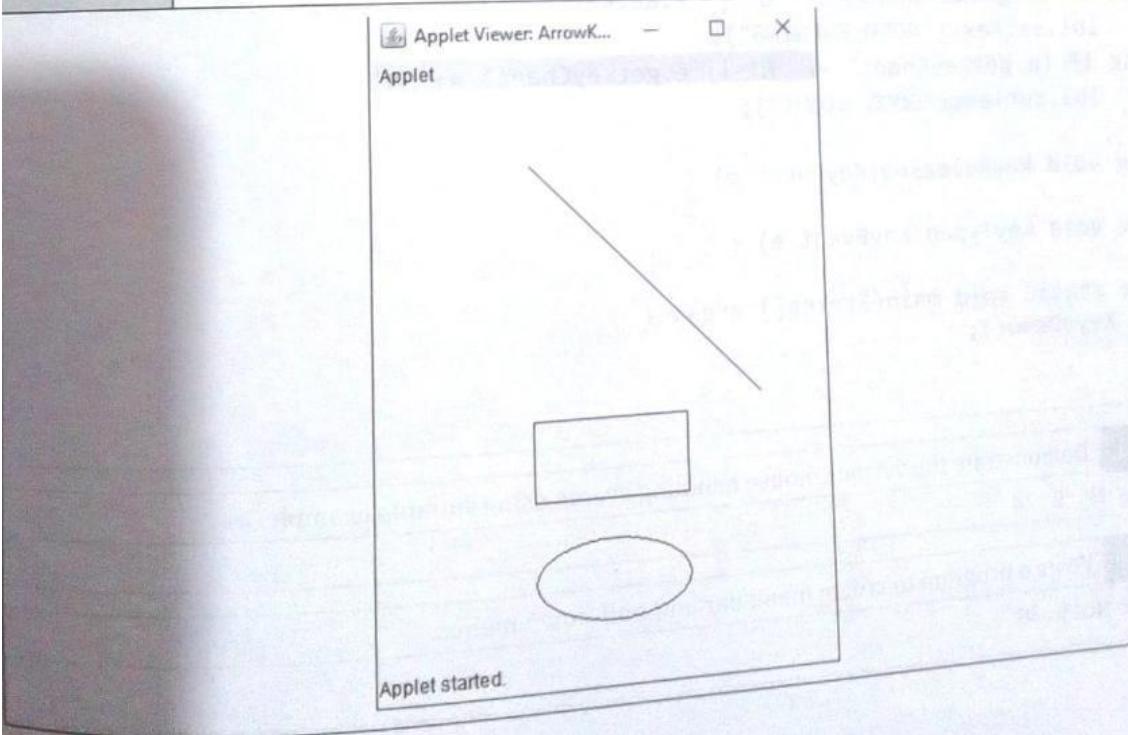
```
switch (key) {
    case KeyEvent.VK_LEFT : x1 = x1 - 10; x2 = x2 - 10;
                            break;
    case KeyEvent.VK_RIGHT: x1 = x1 + 10; x2 = x2 + 10;
                            break;
    case KeyEvent.VK_UP   : y1 = y1 - 10; y2 = y2 - 10;
                            break;
    case KeyEvent.VK_DOWN : y1 = y1 + 10;
                            y2 = y2 + 10;
                            break;
}
repaint();
}

public void keyReleased(KeyEvent ke) {
}

public void keyTyped(KeyEvent ke) {
    repaint();
}

public void paint(Graphics g) {
    g.drawLine(x1, y1, x2, y2);
    g.drawRect(x1, y1 + 160, 100, 50);
    g.drawOval(x1, y1 + 235, 100, 50);
}
}
```

Output



10./* Program 10-To create a window*/

10. Program to create a window when we press M or m the window displays Good Morning, A or a the window displays Good After Noon E or e the window displays Good Evening, N or n the window displays Good Night. (File Name: display.java)

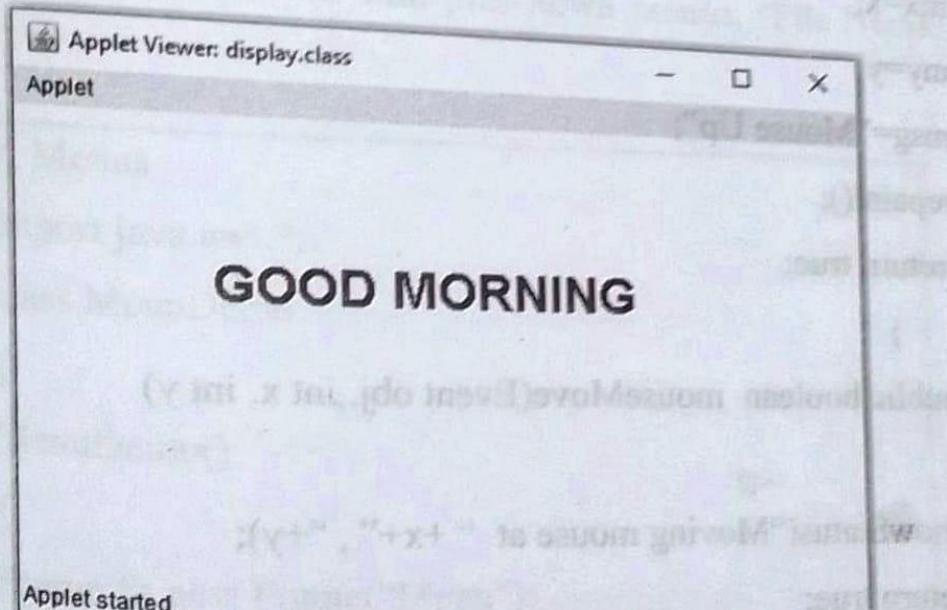
Program:

```
// Display Message
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
public class display extends Applet implements KeyListener
{
    String msg = "";
    Font f;
    public void init()
    {
        setSize (500,500);
        f=new Font("Arial",Font.BOLD,28);
        setForeground(Color.blue);
        addKeyListener(this);
    }
    public void keyReleased(KeyEvent e)
    {
    }
    public void keyTyped(KeyEvent e)
    {
        repaint();
    }
    public void keyPressed(KeyEvent e)
    {
        if (e.getKeyCode()==77)
            msg="GOOD MORNING";
    }
}
```

```
if (e.getKeyCode() == 65)
    msg = "GOOD AFTERNOON";
if (e.getKeyCode() == 69)
    msg = "GOOD EVENING";
if (e.getKeyCode() == 78)
    msg = "GOOD NIGHT";
}

public void paint(Graphics g)
{
    g.setFont(f);
    g.drawString(msg, 100, 100);
}
```

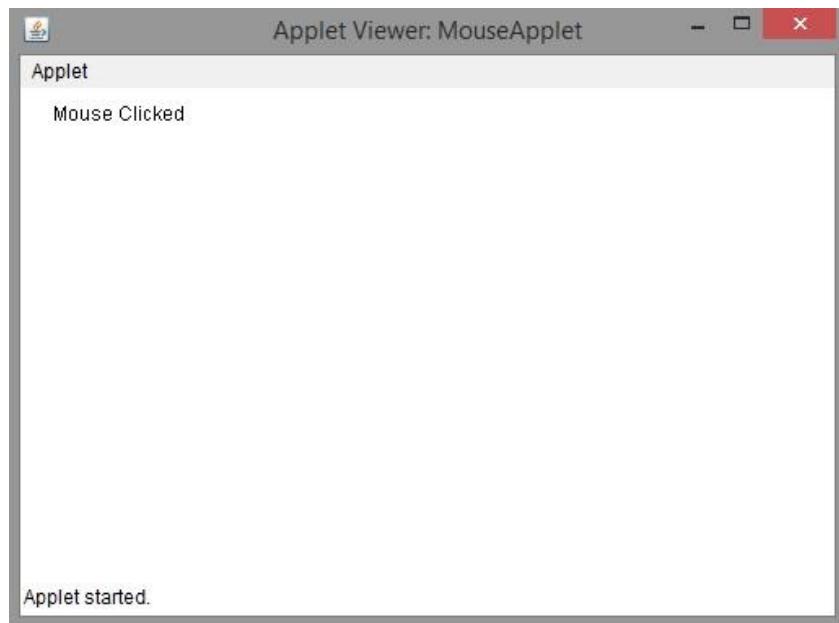
Output:



11. Demonstrate the various mouse handling events using suitable example.

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class B11MouseEvents extends Applet implements MouseListener{
    String msg = "Initial Message";
    public void init(){
        addMouseListener(this);
    }
    public void mouseClicked(MouseEvent m){
        msg = "Mouse Clicked";
        repaint();
    }
    public void mousePressed(MouseEvent m){
        msg = "Mouse Pressed";
        repaint();
    }
    public void mouseReleased(MouseEvent m){
        msg = "Mouse Released";
        repaint();
    }
    public void mouseEntered(MouseEvent m){
        msg = "Mouse Entered";
        repaint();
    }
    public void mouseExited(MouseEvent m){
        msg = "Mouse Exited";
        repaint();
    }
    public void paint(Graphics g){
        g.drawString(msg, 30, 30);
    }
}
```

Output -



12./* Program-Menu bar*/

12. **Program to create menu bar and pull-down menus. (File Name: MenuDemo.java)**

Program:

```
// Menus
import java.awt.*;
class MenuDemo
{
    MenuDemo()
    {
        Frame f= new Frame("Menu");
        MenuBar mb=newMenuBar();
        Menu m=new Menu("Course");
        Menu s=new Menu("Science");
        MenuItem i1=new MenuItem("Arts");
        MenuItem i2=new MenuItem("Commerce");
        MenuItem i3=new MenuItem("BCA");
        MenuItem i4=new MenuItem("B.Sc(PCS)");
        MenuItem i5=new MenuItem("B.Sc(MCS)");
    }
}
```

```
m.add(i1);
m.add(i2);
s.add(i3);
s.add(i4);
s.add(i5);
m.add(s);
mb.add(m);
f.setMenuBar(mb);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
}
public static void main(String args[])
{
new MenuDemo();
}
```

Output:

