

QUESTION BANK FOR III SEMESTER (Term: Nov 2022-Jan 2023)**Object Oriented Programming Laboratory (ADL37/AIL37)****I.A. Marks: 20****Exam Hours: 02****Credits: 0:0:1**

- 1 **Write a java program to display the employee details (ID, Name, Age, Salary) for three employees. Read details using Scanner class. Display the names of employees in the order of their age.**

Program:

```
import java.util.Scanner;

class Employee
{
    int id;
    String name;
    int age;
    double salary;

    Employee(int id, String name, int age, double salary)
    {
        this.id = id;
        this.name = name;
        this.age = age;
        this.salary = salary;
    }
}

public class Employee_Details
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        Employee[] e = new Employee[3];

        for (int i = 0; i < 3; i++)
        {
            System.out.print("Enter employee ID: ");
            int id = sc.nextInt();
            sc.nextLine();
            System.out.print("Enter employee name: ");
            String name = sc.nextLine();
            System.out.print("Enter employee age: ");
            int age = sc.nextInt();
            System.out.print("Enter employee salary: ");
            double salary = sc.nextDouble();

            e[i] = new Employee(id, name, age, salary);
        }

        for (int i = 0; i < 3; i++)
        {
            for (int j = i + 1; j < 3; j++)
```

```

        {
            if (e[i].age > e[j].age)
            {
                Employee temp = e[i];
                e[i] = e[j];
                e[j] = temp;
            }
        }
    }

    // Display employee details in order of age

    System.out.println("Employee details in order of age:");
    for (int i = 0; i < 3; i++)
    {
        System.out.println("ID: " + e[i].id + ", Name: " + e[i].name
                            + ", Age: " + e[i].age + ",
                            Salary: " + e[i].salary);
    }
}
}

```

2 **Create a File Sales.java that should contain a Java program that prompts for and reads in the sales for each of 5 salespeople in a company. It should then print out the id and amount of sales for each salesperson and the total sales. Now modify the program as follows:**

Compute and print the average sale. (You can compute this directly from the total; no loop is necessary.)

Find and print the maximum sale. Print both the id of the salesperson with the max sale and the amount of the sale, e.g., "Salesperson 3 had the highest sale with \$4500."

Note that you don't need another loop for this; you can do it in the same loop where the values are read and the sum is computed. Do the same for the minimum sale.

Ask the user to enter a value. Then print the id of each salesperson who exceeded that amount, and the amount of their sales.

Program:

```

import java.text.NumberFormat;
import java.util.Scanner;

public class Sales
{
    public static void main(String[] args)
    {
        int SALESPEOPLE = 5;
        int sum;
        int maxman = 0;
        int maxsale = Integer.MIN_VALUE;
        int minman = 0;
        int minsale = Integer.MAX_VALUE;
        double average;

        NumberFormat fmt = NumberFormat.getCurrencyInstance();
        Scanner scan = new Scanner(System.in);
        System.out.print("Please enter the number of sales people:
                        ");
        SALESPEOPLE = scan.nextInt();
        int[] sales = new int[SALESPEOPLE];

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

```

```

        System.out.print("Enter sales for salesperson " + (i+1) +
                           ": ");
        sales[i] = scan.nextInt();

        if (maxsale < sales[i])
        {
            maxsale = sales[i];
            maxman = i + 1;
        }
        if (minsale > sales[i]){
            minsale = sales[i];
            minman = i + 1;
        }
    }

    System.out.println("\nSalesperson Sales");
    System.out.println(" ----- ");
    sum = 0;

    for (int i = 0; i < sales.length; i++)
    {
        System.out.println(" " + (i+1) + " " + sales[i]);
        sum += sales[i];
    }
    average = sum / SALESPeOPLE;

    System.out.println("\nTotal sales: " + sum);
    System.out.println("The average: " + average);
    System.out.println("Salesperson " + maxman
        + " had the highest sale with " +
        fmt.format(maxsale)+ ".");
    System.out.println("Salesperson " + minman
        + " had the lowest sale with " + fmt.format(minsale)
        + ".");
    System.out.print("\nPlease enter a value: ");

    int line;
    sum = 0;
    line = scan.nextInt();
    System.out.println("\nSalesperson Sales");
    System.out.println(" ----- ");

    for (int i = 0; i < sales.length; i++)
    {
        if (sales[i] > line)
        {
            System.out.println(" " + (i+1) + " " + sales[i]);
            sum += sales[i];
        }
    }
    System.out.println("\nTotal sales for who exceeded the line:
        "+sum);
}

```

3 Write a java program to multiply two given matrices.

Program:

```

public class MatrixMultiplicationExample
{
    public static void main(String args[])
    {
        //creating two matrices
        int a[][]={{1,1,1},{2,2,2},{3,3,3}};
        int b[][]={{1,1,1},{2,2,2},{3,3,3}};
    }
}

```

```
//creating another matrix to store the multiplication of two matrices
int c[][]=new int[3][3]; //3 rows and 3 columns

//multiplying and printing multiplication of 2 matrices
for(int i=0;i<3;i++)
{
    for(int j=0;j<3;j++)
    {
        c[i][j]=0;
        for(int k=0;k<3;k++)
        {
            c[i][j]+=a[i][k]*b[k][j];
        }//end of k loop
        /* printing matrix element*/
        System.out.print(c[i][j]+" ");
    }//end of j loop
    System.out.println();//new line
}
}
```

- 4 **Write a Java Program that does the following**
- Create a super class called Car. The Car class has the following fields and methods.**
int speed; double regularPrice; String color; double getSalePrice();
 - Create a sub class of Car class and name it as Truck. The Truck class has the following fields and methods.**
int weight; double getSalePrice();
/If weight>2000,10% discount. Otherwise,20%discount.
 - Create a subclass of Car class and name it as Ford. The Ford class has the following fields and methods**
int year; int manufacturerDiscount; double getSalePrice();
//From the sale price computed from Carclass, subtract the manufacturer Discount.
 - Create a subclass of Car class and name it as Sedan. The Sedan class has the following fields and methods.**
int length; double getSalePrice();
//If length>20 feet, 5% discount, Otherwise, 10% discount.
 - Create MyOwnAutoShop class which contains the main() method. Perform the following within the main() method.**
 - Create an instance of Sedan class and initialize all the fields with appropriate values.**
 - Use super(...) method in the constructor for initializing the fields of the superclass.**
 - Create an instance of the Ford class and initialize all the fields with appropriate values**
 - Use super(...) method in the constructor for initializing the fields of the super class.**
 - Create an instance of Car class and initialize all the fields with appropriate values. Display the sale prices of all instances.**

Program:

```
class Car
{
    int speed;
    double regularPrice;
    String color;
```

```

Car(int s,double price,String c)
{
    speed=s;
    regularPrice=price;
    color=c;
}
double getSalePrice()
{
    return regularPrice;
}
}
class Truck extends Car
{
    int weight;
    Truck(int s,double price,String c,int w)
    {
        super(s,price,c);
        weight=w;
    }
    double getSalePrice()
    {
        if(weight>2000)
        {
            regularPrice=regularPrice*0.9;
            return regularPrice;
        }
        else
        {
            regularPrice=regularPrice*0.8;
            return regularPrice;
        }
    }
}
}
class Ford extends Car
{
    int manufacturerDiscount,year;
    Ford(int s,double price,String c,int m)
    {
        super(s,price,c);
        manufacturerDiscount=m;
    }
    double getSalePrice()
    {
        regularPrice-=manufacturerDiscount;
        return regularPrice;
    }
}
}
class Sedan extends Car
{
    int length;
    Sedan(int s,double price,String c,int l)
    {
        super(s,price,c);
        length=l;
    }
    double getSalePrice()
    {
        if(length>20)
        {
            regularPrice=regularPrice*0.95;
            return regularPrice;
        }
        else
        {
            regularPrice=regularPrice*0.9;

```

```

        return regularPrice;
    }
}
}
public class MyOwnAutoShop
{
    public static void main(String[] args)
    {
        Truck t=new Truck(65,2500000,"Red",3000);
        System.out.println("Price of truck is "+t.getSalePrice());
        Car c = new Car(100,800000,"Black");
        System.out.println("Price of Car is "+c.getSalePrice());
        Ford f=new Ford(120,2200000,"Yellow",1 20000);
        System.out.println("Price of ford is "+f.getSalePrice());
        Sedan s= new Sedan(100,3500000,"Blue",22);
        System.out.println("Price of Sedan is "+s.getSalePrice());
    }
}

```

5

Write a Java Program that implements the following

- a. Define a class SavingsAccount with following characteristics.**
- b. Use a static variable annualInterestRate to store the annual interest rate for all account holders.**
- c. Private data member savingsBalance indicating the amount the saver currently has on deposit.**
- d. Method calculateMonthlyInterest to calculate the monthly interest as (savingsBalance * annualInterestRate / 12) . After calculation, the interest should be added to savingsBalance.**
- e. Static method modifyInterestRate to set annualInterestRate.**
- f. Parameterized constructor with savingsBalance as an argument to set the value of that instance.**
- g. Test the class SavingsAccount to instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs.2000.00 and Rs3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.**

Program:

```

class SavingsAccount
{
    static int annualInterestRate;
    private double savingsBalance;
    SavingsAccount(double s)
    {
        savingsBalance=s;
    }
    static void modifyInterestRate(int x)
    {
        annualInterestRate=x;
    }

    void calculateMonthlyInterest()
    {
        double d=(savingsBalance*annualInterestRate)/12;
        savingsBalance=savingsBalance+d;
    }
    void display()
    {
        System.out.println(savingsBalance);
    }
}

```

```

public class Main
{
    public static void main(String[] args)
    {
        SavingsAccount saver1=new SavingsAccount(2000);
        SavingsAccount saver2=new SavingsAccount(3000);
        SavingsAccount.modifyInterestRate(4);
        saver1.calculateMonthlyInterest();
        saver2.calculateMonthlyInterest();
        saver1.display();
        saver2.display();
        SavingsAccount.modifyInterestRate(5);
        saver1.calculateMonthlyInterest();
        saver2.calculateMonthlyInterest();
        saver1.display();
        saver2.display();
    }
}

```

- 6 **Write a Java Program that does the following related to Inheritance:**
- Create an abstract class called 'Vehicle' which contains the 'hashelmet','year of manufacture' and two abstract methods 'getData()' and 'putData()'. Demonstrate the error when attempt is made to create objects of 'Vehicle'.**
 - Have two derived classes 'TwoWheeler' and 'FourWheeler', 'FourWheeler' is a final class. Demonstrate the error when attempt is made to inherit from 'FourWheeler'.**
 - Your abstract class should have overloaded constructors that initializes hashelmet and year of manufacture for TwoWheeler and FourWheeler respectively.**
 - 'TwoWheeler' has data elements 'Brand', 'Cost', 'EngineType' (possible values "2 stroke", "4 stroke"), and 'Color'. Demonstrate the various ways in which the two abstract methods can be dealt 'getData()' and 'putData()' can be dealt with by the derived classes, 'TwoWheeler' and 'FourWheeler'.**
 - The sub-class of 'TwoWheeler' called 'MyTwoWheeler' has the element 'OwnerName'.**
 - Use the super keyword to initialize an object of 'MyTwoWheeler' with all the values of its parent class data elements.**

Program:

```

import java.util.*;

abstract class Vehicle
{
    boolean hashelmet;
    int yom;
    abstract void getData();
    abstract void putData();
    Vehicle(boolean h,int n)
    {
        hashelmet=h;
        yom=n;
    }
}

class TwoWheeler extends Vehicle
{
    private String Brand;
    protected int Cost;
    String EngineType;
    public String Color;
    TwoWheeler(int n)
    {
        super(true,n);
    }
}

```

```

    }
    void getData()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Brand name, Cost, EngineType and Colour");
        Brand=sc.next();
        Cost=sc.nextInt();
        EngineType=sc.next();
        Color=sc.next();
    }
    void putData()
    {
        System.out.println("Brand:"+Brand+"\nCost:"+Cost+"\n EngineType:"
                           +EngineType+"\nColor:"+Color+" \nYear of
                           Manufacture:"+yom+"\nHas helmet:"+hashelmet);
    }
}
final class FourWheeler extends Vehicle
{
    FourWheeler(int n)
    {
        super(false,n);
    }
    void getData()
    {
    }
    void putData()
    {
        System.out.println("Year of Manufacture:"+yom+"\nHas
                           helmet:"+hashelmet);
    }
}
class MyTwoWheeler extends TwoWheeler
{
    String name;
    MyTwoWheeler(String name,int n)
    {
        super(n);
        this.name=name;
    }
    void display()
    {
        super.getData();
        putData();
        System.out.println("Name:"+name);
    }
}
/* class A extends FourWheeler
{
    A()
    {
        super(5);
    }
}
*/
public class Main
{
    public static void main(String[] args)
    {
        /*Vehicle v=new Vehicle();
        Cannot Create instance of an abstract class*/
        TwoWheeler t1=new TwoWheeler(1995);
        FourWheeler f1=new FourWheeler(2006);
        t1.getData();
        t1.putData();
    }
}

```



```

        fl.putData();
    }
}

```

7

Write a java program to implement stack operation (Push, Pop). Your class should have an empty constructor. Create two objects to demonstrate the stack operation for 10 items.

Program:

```

class Stack
{
    int stck[] = new int[10];
    int tos;
    Stack()
    {
        tos = -1;
    }
    void push(int item)
    {
        if(tos == 9)
            System.out.println("Stack is full.");
        else
            stck[++ tos] = item;
    }
    int pop()
    {
        if(tos < 0)
        {
            System.out.println("Stack underflow.");
            return 0;
        }
        else
        {
            return stck[tos --];
        }
    }
}
class TestStack
{
    public static void main(String args[])
    {
        Stack obj1 = new Stack();
        Stack obj2 = new Stack();
        for(int i=0 ; i<10; i++)
            obj1.push(i);
        for(int i=0 ; i<10; i++)
            obj2.push(i);
        System.out.println("\n First Stack: \n");
        for(int i=0 ; i<10; i++)
            System.out.println(obj1.pop() + "\t");
        System.out.println("\n Second Stack: \n");
        for(int i=0 ; i<10; i++)
            System.out.println(obj2.pop() + "\t");
    }
}

```

- 9 **Write a java program to maintain the student details like USN, Dept names, 3 subject grades and SGPA in student package and keep the staff details such as Staffid, StaffName, designation and subjects handled in a staff package. In main class use these two packages details for Staff and Student classes and display the student and staff information as requested by the user.**

Program:

```
package details;
public class student
{
    String name,dept,grade1,grade2,grade3,usn;
    double sgpa,cgpa;
    public student(String n,String u,String d,String g1,String g2,String
        g3,double s)
    {
        name=n;
        usn=u;
        dept=d;
        grade1=g1;
        grade2=g2;
        grade3=g3;
        sgpa=s;
    }
    public void show()
    {
        System.out.println("student "+name+"with usn:"+usn+"of department
            "+dept);
        System.out.println("the grades are "+grade1+", "+grade2+", "+grade3);
    }
}
package staffdetails;
public class staff
{
    String stname,stdept,stusn,des;
    public staff(String sn,String sd,String su,String d)
    {
        stname=sn;
        stdept=sd;
        stusn=su;
        des=d;
    }
    public void show1()
    {
        System.out.println("the staff name:"+stname+"doing "+des+"with
            usn:"+stusn+"of department "+stdept);
    }
}

import details.student;
import staffdetails.staff;

public class demo
{
    public static void main(String args[])
    {
        student s=new student("yash","lms21AI032","AIML","O","O","O",10);
        staff st=new staff("Raj","lms21AI099","AIML","professor");
        s.show();
        st.show1();
    }
}
```

10	<p>Write java program to create a package called AdvMath, which has two classes. In main class use this package to display the result as requested by the user.</p> <p>i.To calculate $y = \sin(x) + \cos(x) + \tan(x)$</p> <p>Program:</p> <pre> package AdvMath; import java.util.*; public class MATHs { double x; double n; public MATHs(double x) { this.x=x; } public MATHs(int n) { this.n=n; } public void Trigo() { System.out.println("Y="+ (Math.sin(x)+Math.cos(x)+Math.tan(x))); } public void Pyth() { for(int a=1;a<n;a++) { for(int b=a+1;b<n;b++) { for(int c=b+1;c<n;c++) { if((c*c)==(a*a)+(b*b)) { System.out.println("Triplets:\n"+a+"\n"+b+"\n"+c); } } } } } } </pre>
11	<p>Write a java program to keep details of bank customer name and balance in MyPack package, Initialize using constructors and define display function. In main class use this package and pass the information (name and balance) and display using display function.</p> <p>Program:</p> <pre> package MyPack; public class Balance { String name; double bal; public Balance(String n, double b) { name=n; } } </pre>

```

        bal=b;
    }
    public void show()
    {
        if (bal>0)
            System.out.println(name+" "+bal);
    }
}
import MyPack.Balance;

class TestBalance
{
    public static void main(String args[])
    {
        Balance test=new Balance("abc",99.8);
        test.show();
    }
}

```

12 **Write a Java program to create 2 threads to perform the following operations. (Extend Thread class)**

Thread 1 will print all the prime numbers from 1 to 100. Thread will sleep for 0.5 second after printing every number.

Thread 2 will print all the numbers from 1 to 100 which are divisible by 2, 4, & 6. Thread will sleep for 0.5 second after printing every number.

Program:

```

import java.lang.*;
import java.io.*;
import java.util.*;

class Prime extends Thread
{
    public void run()
    {
        try
        {
            int i =0;
            int num =0;
            //Empty String
            //String primeNumbers = "";
            System.out.println("Thread-1: Prime Thread");

            for (i = 1; i <= 100; i++)
            {
                int counter=0;
                for(num =i; num>=1; num--)
                {
                    if(i%num==0)
                    {
                        counter = counter + 1;
                    }
                }
                if (counter ==2)
                {
                    //Appended the Prime number to the String
                    System.out.println("PrimeNumber: " + i);
                    //primeNumbers = primeNumbers + i + " ";
                }
            }

            //System.out.println("Prime numbers from 1 to 100 are :");
            //System.out.println(primeNumbers);
        }
    }
}

```

```

        catch (Exception e){}
    }
}
class printeven extends Thread
{
    public void run()
    {
        try
        {
            System.out.println("Thread-2: Even Thread");
            for(int i=1;i<=100;i++)
            {
                if(i%2==0)
                {
                    System.out.println ("Prime No.= "+i);
                    System.out.println();
                }
                Thread.sleep(500);
            }
        }
        catch (Exception e){}
    }
}
class MainThread
{
    public static void main(String args[])
    {
        Prime primethread = new Prime();
        printeven peventhread = new printeven();

        primethread.start();
        peventhread.start();
    }
}

```

- 13 **Java Program to create 2 threads. Print Prime Numbers from 1 to 50 using Thread1. Print Prime Numbers from 100 to 150 using Thread2. After Every number, put the thread to sleep. Create Threads using Runnable Interface**

Program:

```

class MultithreadDemol implements Runnable
{
    public void run()
    {
        System.out.println("Thread 1");
        for(int i=0; i<=50; i++)
        {
            try
            {
                int counter = 0;
                for(int num = i; num>=1; num--)
                {
                    if(i%num == 0)
                    {
                        counter = counter + 1;
                    }
                }
                if(counter == 2)
                {
                    System.out.println("Prime Number (1 - 50): " + i);
                    Thread.sleep(500);
                }
            }
            catch (InterruptedException e)
            {
                System.out.println("Caught: " + e);
            }
        }
    }
}

```

```

    }
    }
}
class MultithreadDemo2 implements Runnable
{
    public void run()
    {
        System.out.println("Thread 2");
        for(int i=100; i<=150; i++)
        {
            try
            {
                int counter = 0;
                for(int num = i; num>=1; num--)
                {
                    if(i%num == 0)
                    {
                        counter = counter + 1;
                    }
                }
                if(counter == 2)
                {
                    System.out.println("Prime Number (100 - 150): " + i);
                    Thread.sleep(500);
                }
            }
            catch (InterruptedException e)
            {
                System.out.println("Caught: " + e);
            }
        }
    }
}
public class Main
{
    public static void main(String args[])
    {
        MultithreadDemo1 obj1 = new MultithreadDemo1();
        Thread threadobj1 = new Thread(obj1);
        threadobj1.start();
        MultithreadDemo2 obj2 = new MultithreadDemo2();
        Thread threadobj2 = new Thread(obj2);
        threadobj2.start();
    }
}

```

14 Write a Java program to display an exception when we attempt to divide any number by zero. Also demonstrate how it can be handled using exception handling.

Program:

```

import java.io.*;

class GFG
{
    public static void main(String[] args)
    {
        int a = 5;
        int b = 0;
        try
        {
            System.out.println(a / b); // throw Exception
        }
        catch (ArithmeticException e)
        {
            // Exception handling logic
        }
    }
}

```

	<pre> { // Exception handler System.out.println("Divided by zero operation cannot possible") } } } </pre>
15	<p>Write a Java program to demonstrate exception handling using try, multiple catch block and finally block. Throw an error in try block to handle array out of bound index.</p> <p>Program:</p> <pre> public class Main { public static void main(String[] args) { //try block containing exception prone code try { System.out.println("try Block:: Begin"); int myArray[]=new int[5]; myArray [5]=10/0; } //multiple catch blocks catch(ArithmeticException e) { System.out.println("Arithmetic Exception :: Divide by zero!!"); } catch(ArrayIndexOutOfBoundsException e) { System.out.println("ArrayIndexOutOfBoundsException :: Accessed index out of bounds"); } catch(Exception e) { System.out.println("Exception :: " + e.getMessage ()); } System.out.println("rest of the code"); } } </pre>
16	<p>Write java program to demonstrate applet</p> <p>Program:</p> <pre> import java.applet.Applet; import java.awt.Graphics; public class AppletEx1 extends Applet { public void paint(Graphics g) { g.drawString("welcome to applet",150,150); } } /* <applet code="AppletEx1" width="300" height="300"> </applet> */ </pre> <p>Output: > javac AppletEx1.java > appletviewer AppletEx1.java</p>
17	<p>Write java program to perform addition of two numbers using applet.</p>

Program:

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

public class AddinApplet extends Applet
{
    Label l1,l2,l3;
    TextField t1,t2,t3;
    Button b;
    int x,y,z;
    public void init()
    {
        l1 = new Label("Number 1:");
        t1 = new TextField(10);
        l2 = new Label("Number 2:");
        t2 = new TextField(10);
        l3 = new Label("Total=");
        t3 = new TextField(10);
        b = new Button("Add");
        b.addActionListener(new test());

        add(l1); add(t1);
        add(l2); add(t2);
        add(l3); add(t3);
        add(b);
    }
    class test implements ActionListener
    {
        public void actionPerformed(ActionEvent e)
        {
            if(e.getSource() == b)
            {
                x = Integer.parseInt(t1.getText());
                y = Integer.parseInt(t2.getText());
                z = x+y;
                t3.setText(String.valueOf(z));
            }
        }
    }
}
/*<applet code="AddinApplet" width="500" height="600">
</applet>*/
```

Output: > javac AddinApplet.java
> appletviewer AddinApplet.java

18 Write java program to demonstrate swing**Program:**

```
import javax.swing.*;

public class SwingEx1 extends JFrame
{
    JFrame f;

    SwingEx1()
    {
        JButton b=new JButton("Click");
        b.setBounds(130,100,100, 40);

        add(b);
    }
}
```



```

        setSize(400,500);
        setLayout(null);
        setVisible(true);
    }

    public static void main(String[] args)
    {
        new SwingEx1();
    }
}

```

Output: >javac SwingEx1.java
>java SwingEx1

- 19 **Write a java program to create a frame containing three buttons (Yes, No, Close). When button yes or no is pressed, the message "Button Yes/No is pressed" gets displayed in label control. On pressing CLOSE button frame window gets closed.**

Program:

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class ButtonDemo extends JFrame
{
    JButton yes,no,close;
    JLabel lbl;
    ButtonDemo()
    {
        yes = new JButton("YES");
        no = new JButton ("No");
        close = new JButton ("CLOSE");
        lbl = new JLabel ("");
        setLayout (new GridLayout(4,1));
        setSize (400,200);
        add(yes);
        add(no);
        add(close);
        add(lbl);
        setVisible(true);

        //setDefaultCloseOperation(JFrame.EXIT_NO_CLOSE);
        ButtonHandler bh = new ButtonHandler();
        yes.addActionListener(bh);
        yes.addActionListener(bh);
        no.addActionListener(bh);
        close.addActionListener(bh);
    }
    class ButtonHandler implements ActionListener
    {
        public void actionPerformed(ActionEvent ae)
        {
            if (ae.getSource()==yes)
            {
                lbl.setText("Button Yes is pressed");
            }
            if (ae.getSource()==no)
            {
                lbl.setText("Button No is pressed");
            }
            if (ae.getSource()==close)
            {
                System.exit(0);
            }
        }
    }
}

```

```

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

```

20 **Write a program to display "All The Best" in 5 different colors on screen. (Using AWT/Swing)**

Program:

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Color_Demo extends Frame
{
    Label lbl1, lbl2, lbl3, lbl4, lbl5;
    public Color_Demo()
    {
        lbl1 = new Label("All The Best");
        lbl1.setForeground(Color.red);
        add(lbl1);
        lbl2 = new Label("All The Best");
        lbl2.setForeground(Color.magenta);
        add(lbl2);
        lbl3 = new Label("All The Best");
        lbl3.setForeground(Color.blue);
        add(lbl3);
        lbl4 = new Label("All The Best");
        lbl4.setForeground(Color.green);
        add(lbl4);
        lbl5 = new Label("All The Best");
        lbl5.setForeground(Color.cyan);
        add(lbl5);
        setVisible(true);
        setSize(400, 300);
        setLayout(new FlowLayout());
        setBackground(Color.gray);
    }
    public void paint(Graphics g)
    {
        g.setColor(Color.magenta);
        g.drawString("All The Best", 100, 100);
        g.setColor(Color.cyan);
        g.drawString("All The Best", 150, 150);
        g.setColor(Color.red);
        g.drawString("All The Best", 200, 200);
        g.setColor(Color.black);
        g.drawString("All The Best", 250, 250);
    }
    public static void main(String[] args)
    {
        new Color_Demo();
    }
}

```