BANARSIDAS CHANDIWALA INSTITUTE OF INFORMATION TECHNOLOGY

Affiliated with GGSIPU

Sector-16C, Dwarka, New Delhi

Bachelor's of Computer Application

Subject: Java Programming (272) Practical File

Name: RAUNAK

Course: BCA 272

Enrollment Number: 01711102021

Submitted To :Meetender Adhana

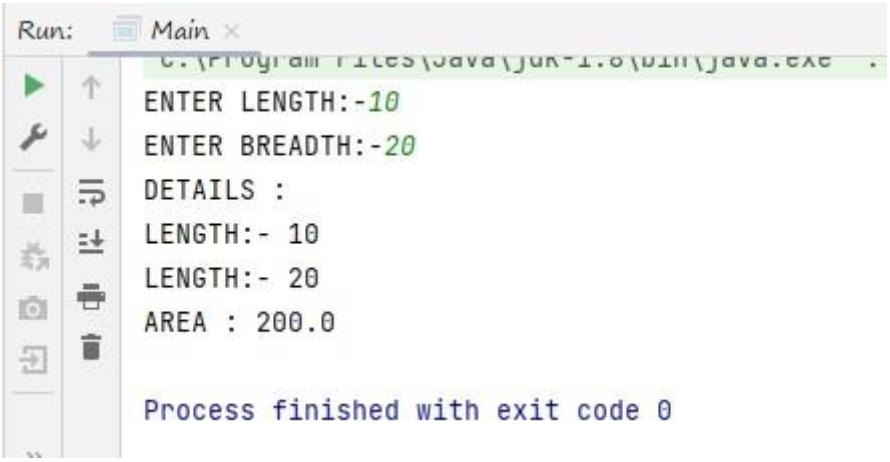
I. Write a program declaring a class Rectangle with data member"s length and breadth and member functions Input, Output and CalcArea.
2. Write a program to demonstrate use of method overloading to calculate area of square, rectangle and triangle.
3. Write a program to demonstrate the use of static variable, static method and static block.
4. Write a program to demonstrate concept of "this".
5. Write a program to demonstrate multi-level and hierarchical inheritance.
6. Write a program to use super() to invoke base class constructor.
7. Write a program to demonstrate run-time polymorphism.
8. Write a program to demonstrate the concept of aggregation.
9. Write a program to demonstrate the concept of abstract class with constructor and "final" method.
l 0. Write a program to demonstrate the concept of interface when two interfaces have unique methods and same data members.
11. Write a program to demonstrate checked exception during file handling.
12. Write a program to demonstrate unchecked exception
13. Write a program to demonstrate creation of multiple child threads.
14. Write a program to use Byte stream class to read from a text file and display the content on the output screen.
15. Write a program to demonstrate any event handling.
16. Create a class employee which have name, age and address of employee, include method:; getdata() and showdata(), getdata() takes the input from the user, showdata() display the data in following format: Name: Age: Address:

17. Write a Java program to perform basic Calculator operations. Make a menu driven program to select operation to perform (+ - * /). Take 2 integers and perform operation as chosen by user.
18. Write a program to make use of BufferedStream to read lines from the keyfuoard until 'STOP' is fyped. –
19. Write a program declaring a Java class called SavingsAccount with members "accountNumber" and "Balance". Provide member functions as "depositAmount ()" and ''withdraw Amount O''. If user tries to withdraw an amount greater than their balance then throw a user-defined exception.
20. Write a program creating 2 threads using Runnable interface. Print your name in "run()" method of first class and "Hello Java" in "run()" method of second thread.
21. Write a swing application that uses atleast 5 swing controls
22. Write ajava program to insert and update details data in the database.
23. Write a java program to retrieve data from database and display it

Q1)Write a program declaring a class Rectangle with data member"s length and breadth and member functions Input, Output and CalcArea. **INPUT**

```
import java.util.Scanner;
class Rectangle{
   int len;
int bre;
        void input()
        {
            Scanner a= new Scanner(System.in);
System.out.print("ENTER LENGTH:-");
len=a.nextInt();
            System.out.print("ENTER BREADTH:-");
            bre=a.nextInt();
        }
        void output()
        {
```

```
            System.out.println("LENGTH:- "+ len);
            System.out.println("LENGTH:- "+ bre);
        }
        double CalArea(){
return len * bre;           }
}
public class Main {     public static void
main(String[] args) {        Rectangle r =
new Rectangle();        r.input();
      System.out.println("DETAILS : ");
r.output();
      System.out.println("AREA : "+r.CalArea());
   }
}
```

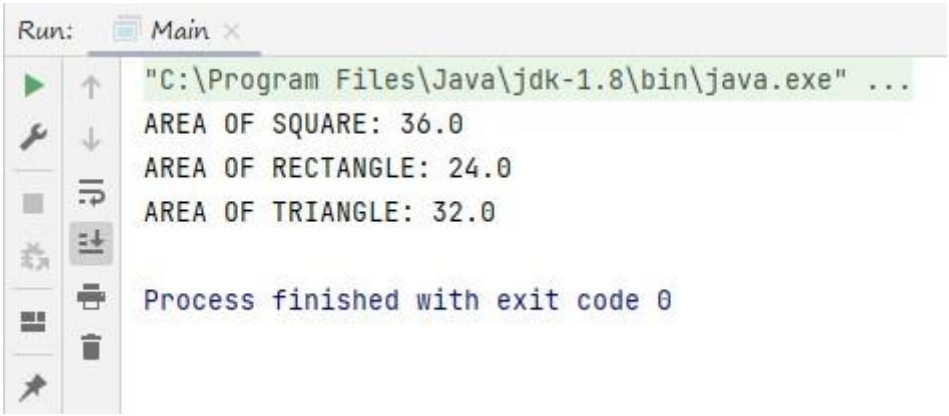


Q2) Write a program to demonstrate use of method overloading to calculate area of square, rectangle and triangle.

INPUT

```
class Area{
   public double ar(double side){
return side * side;
   }
   public double ar(float length, float breadth){
return  length * breadth;
   }
   public double ar(double height, double base){
return  height*base/2;
```

```
    }
}
public class Main {
    public static void main(String[] args){
Area a = new Area();        double
sqArea = a.ar(6);
        System.out.println("AREA OF SQUARE: "+sqArea);
double reArea = a.ar(6,4);
        System.out.println("AREA OF RECTANGLE: "+reArea);
double triArea = a.ar(8,4);
        System.out.println("AREA OF TRIANGLE: "+triArea);
    }
}
```

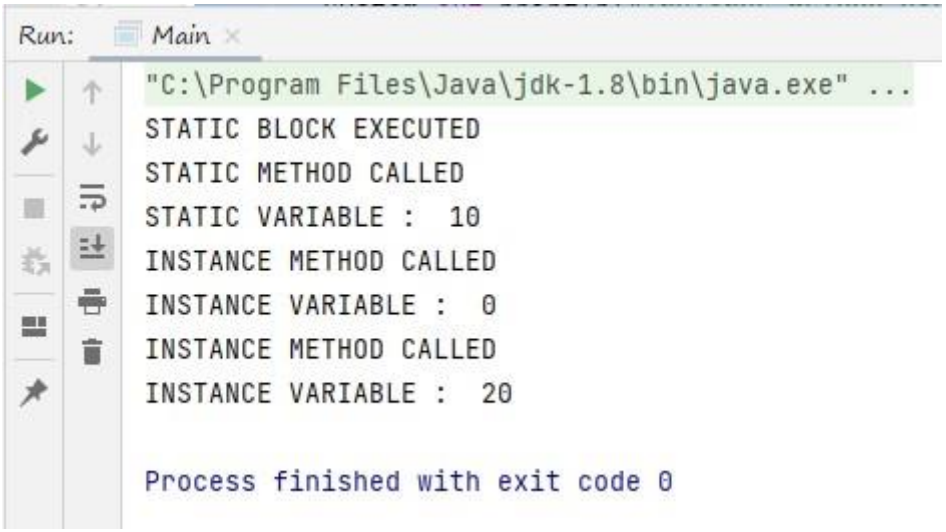


.Q3) Write a program to demonstrate the use of static variable, static method and static block

INPUT

```
class Static{    static
int staticvar;    int
instancevar ;
    static{
        System.out.println("STATIC BLOCK EXECUTED");
staticvar=10;
    }
    static void staticmethod(){
        System.out.println("STATIC METHOD CALLED ");
        System.out.println("STATIC VARIABLE :  "+staticvar);
```

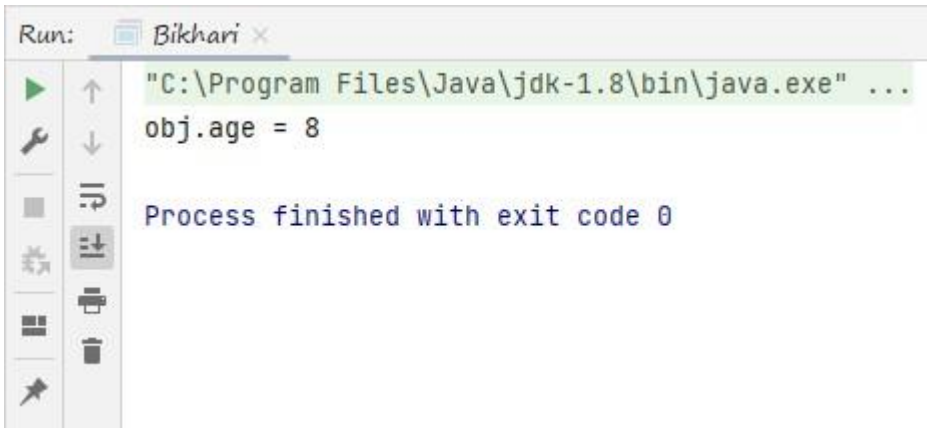
```
    }
     void instancemethod(){
        System.out.println("INSTANCE METHOD CALLED ");
        System.out.println("INSTANCE VARIABLE :  "+instancevar);
    }
}
public class Main {     public static void
main(String[] args) {
        Static.staticmethod();
Static ob1 = new Static();
ob1.instancemethod();         Static
ob2 = new Static();
ob2.instancevar=20;
        ob2.instancemethod();


    }
}
```



Q4) Write a program to demonstrate concept of "this". **INPUT**

```
class Bikhari {

    int age;
Bikhari(int age){
        this.age = age;
    }
    public static void main(String[] args) {
```

```
        Bikhari obj = new Bikhari(8);
        System.out.println("obj.age = " + obj.age);
    }
}
```

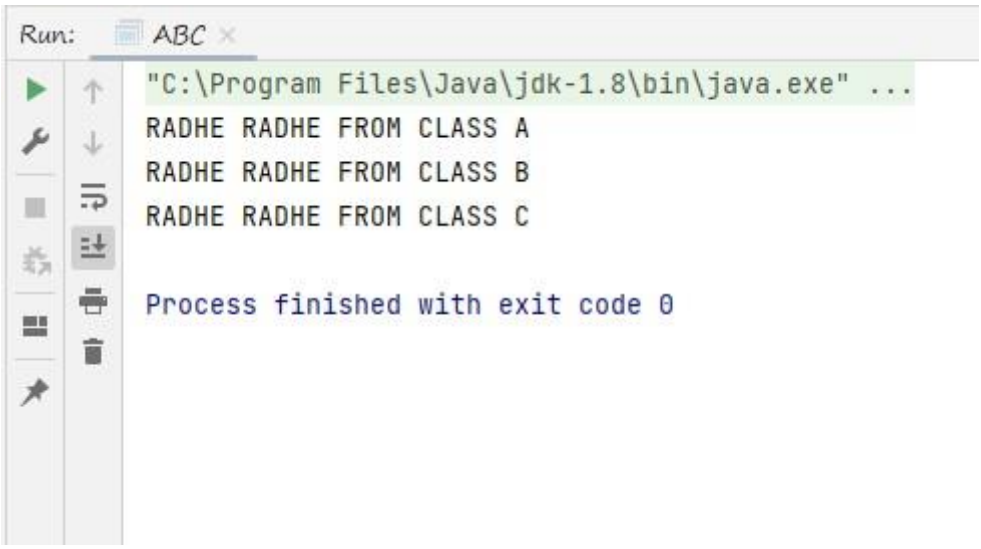


Q5) Write a program to demonstrate multi-level and hierarchical inheritance

INPUT

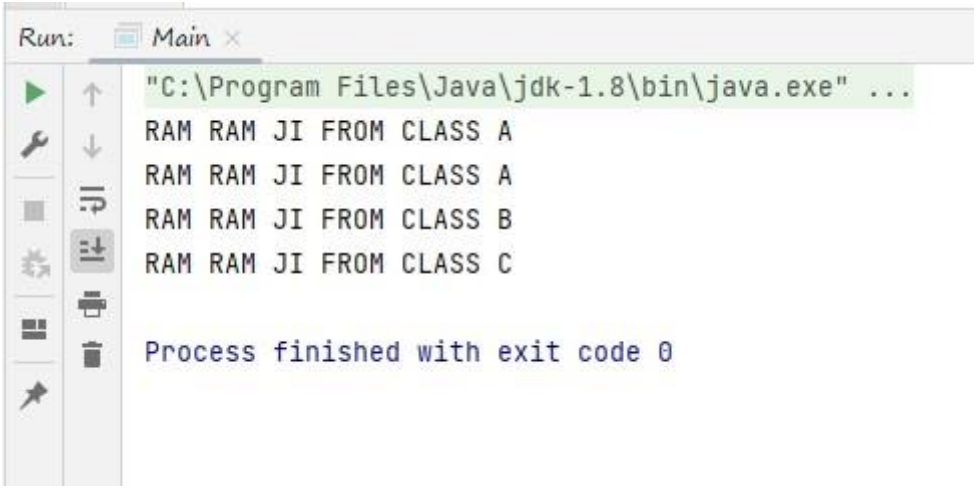
```
/multilevel class
A1{    void
show_A1(){
        System.out.println("RADHE RADHE FROM CLASS A");
    }
}
class B1 extends A1{
void show_B1(){
        System.out.println("RADHE RADHE FROM CLASS B");
    }
}
class C1 extends B1{
void show_C1(){
        System.out.println("RADHE RADHE FROM CLASS C");
    }
}
class ABC{
    public static void main(String[] args) {
        C1 ob = new C1();
ob.show_A1();
ob.show_B1();        ob.show_C1();
```

```
    }
}
```




```
//hierarchial
class A{    void
show_A(){
        System.out.println("RAM RAM JI FROM CLASS A");
    }
}
class B extends A{
void show_B(){
        System.out.println("RAM RAM JI FROM CLASS B");
    }
}
class C extends A{
void show_C(){
        System.out.println("RAM RAM JI FROM CLASS C");
    }
}
public class Main {    public static void
main(String[] args) {
        C ob = new C();
B ob1 = new B();
ob.show_A();
ob1.show_A();
ob1.show_B();
        ob.show_C();
```

```
    }
}
```

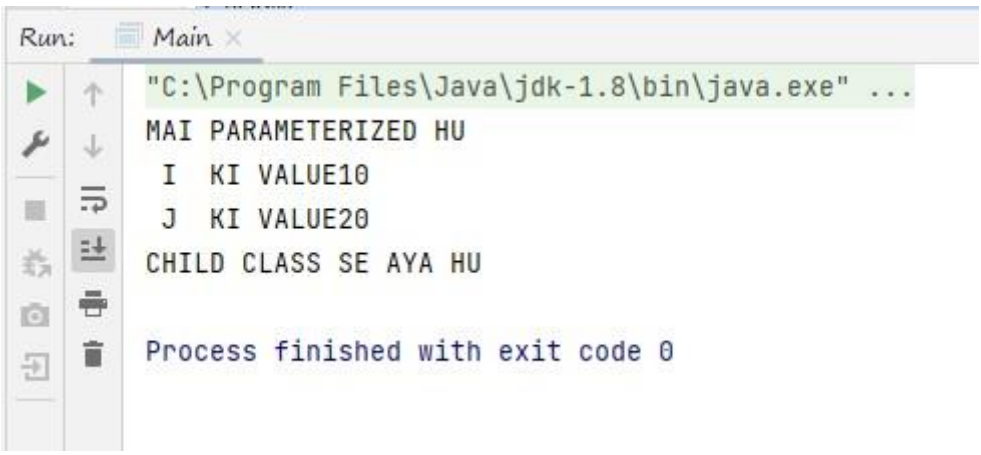


Q6) Write a program to use super() to invoke base class constructor.

INPUT

```
// Base Class
class ABC {

    // Creating Constructor for
// class Programming.
    public ABC()
    {
        System.out.println("BASE CLASS SE AYA HU");
    }

    // Parameterized Constructor
    public ABC(int i, int j)
    {
        System.out.println("MAI PARAMETERIZED HU");
        System.out.println(" I  KI VALUE"+i);
        System.out.println(" J  KI VALUE"+j);
    }
}

// Child Class
class DP extends ABC{

    public DP()
    {
```

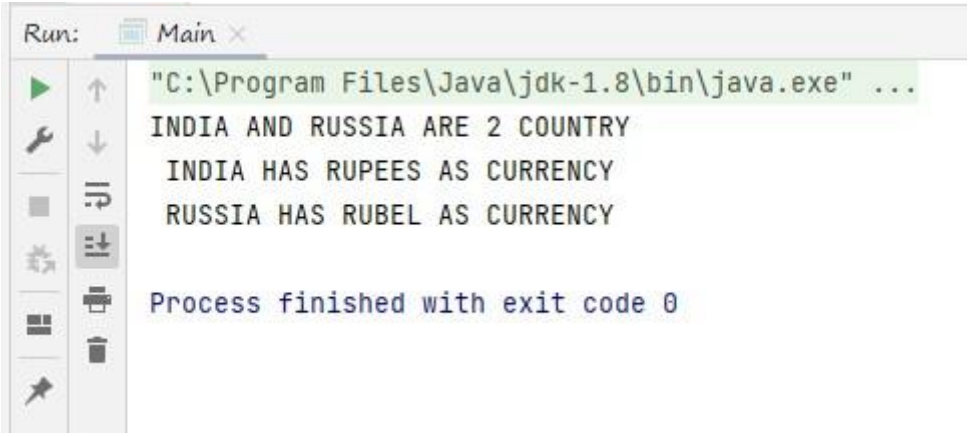
```
        super(10, 20);
        System.out.println("CHILD CLASS SE AYA HU");
    }
}

public class Main {     public static void
main(String[] args)
    {
        DP obj = new DP();
    }
}
```



Q7) Write a program to demonstrate run-time polymorphism **INPUT**

```
class Country{
void show(){
        System.out.println("INDIA AND RUSSIA ARE 2 COUNTRY");
    }
}
class India extends Country{
    void show(){
        System.out.println(" INDIA HAS RUPEES AS CURRENCY ");
    }
}
class Russia extends Country{
    void show(){
        System.out.println(" RUSSIA HAS RUBEL AS CURRENCY ");
    }
}
```

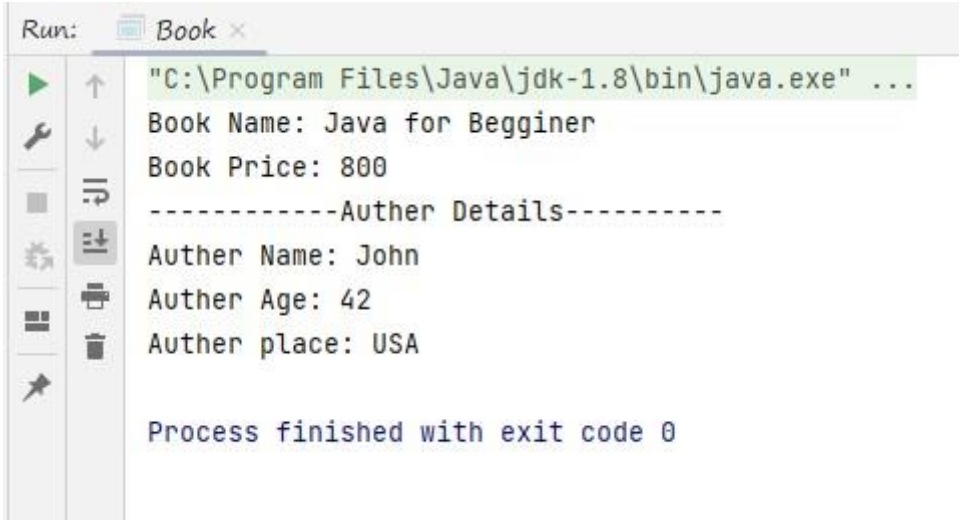
```
public class Main {     public static void
main(String[] args) {
        Country a= new Country();
        Country b = new India();
Country c = new Russia();
a.show();
        b.show();
        c.show();
    }
}
```



Q8) Write a program to demonstrate the concept of aggregation.

```
class Author
{
    String authorName;
int age;
    String place;

    // Author class constructor
    Author(String name, int age, String place)
    {
        this.authorName = name;
        this.age = age;
        this.place = place;
    }
}

class  Book
{
```

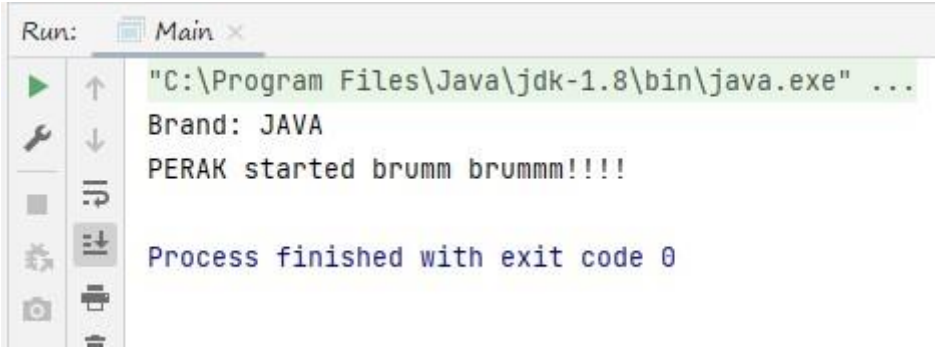
```
    String name;
int price;     //
author details
    Author auther;
    Book(String n, int p, Author auther)
    {
        this.name = n;
this.price = p;
        this.auther = auther;
    }
    public static void main(String[] args) {
        Author auther = new Author("John", 42, "USA");
        Book b = new Book("Java for Begginer", 800, auther);
        System.out.println("Book Name: "+b.name);
        System.out.println("Book Price: "+b.price);
        System.out.println("------------Auther Details----------");
        System.out.println("Auther Name: "+b.auther.authorName);
        System.out.println("Auther Age: "+b.auther.age);
        System.out.println("Auther place: "+b.auther.place);
    }

}
```



9) Write a program to demonstrate the concept of abstract class with constructor and "final" method.

```
 abstract class Vehicle {
private String brand;     public
```

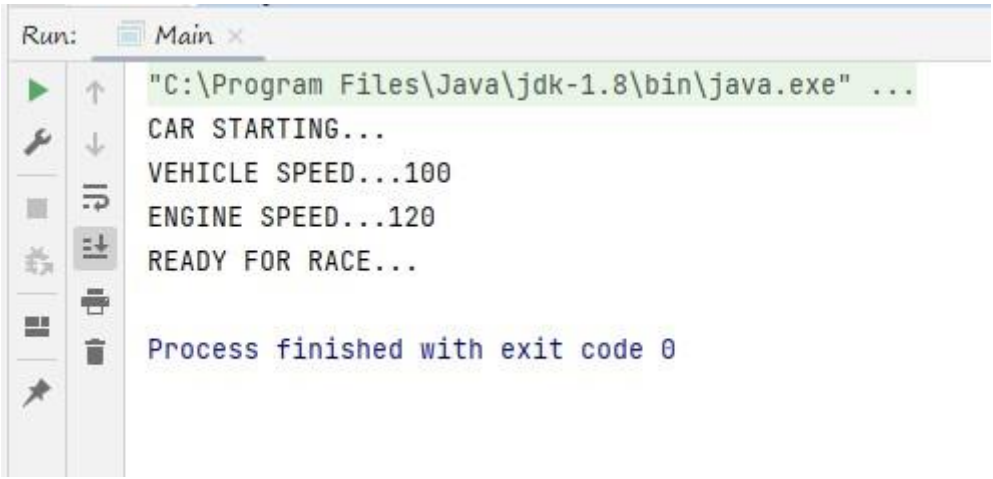
```
Vehicle(String brand) {
this.brand = brand;
   }
   public final void display() {
      System.out.println("Brand: " + brand);
   }
   public abstract void start();
}
class Bike extends Vehicle {
public Bike(String brand) {
super(brand);
   }
   public void start() {
      System.out.println("PERAK started brumm brummm!!!!");
   }
}
public class Main {
   public static void main(String[] args) {
Bike a = new Bike("JAVA");        a.display();
      a.start();
   } ]
```



Q10) Write a program to demonstrate the concept of interface when two interfaces have unique methods and same data members. **INPUT**

```
interface Vehicle{
int speed=100;
void start();
}
```

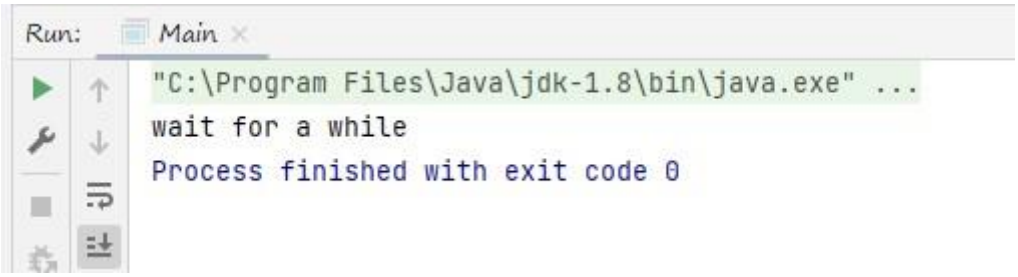
```
interface Engine{
int speed = 120;
void go();
}
class car implements Vehicle,Engine{
    public void start(){
        System.out.println("CAR STARTING...");
        System.out.println("VEHICLE SPEED..."+Vehicle.speed);
        System.out.println("ENGINE SPEED..."+Engine.speed);
    }
    public void go() {
        System.out.println("READY FOR RACE...");
    }
}
public class Main {     public static void
main(String[] args) {         car a= new
car();         a.start();
        a.go();
    }
}
```

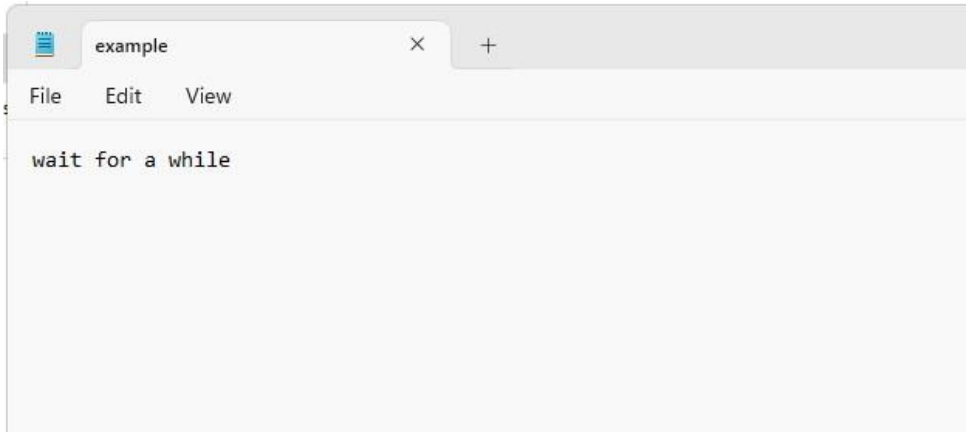


Q11) Write a program to demonstrate checked exception during file handling.

```
import java.io.File; import
java.io.FileReader; import
java.io.FileNotFoundException; import
java.io.IOException; public class Main {
public static void main(String[] args) {
```

```
        File file = new File("example.txt");
FileReader reader = null;        try {
            reader = new FileReader(file);
int data = reader.read();            while
(data != -1) {
                System.out.print((char) data);
                data = reader.read();
            }
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + file.getName());
        } catch (IOException e) {
            System.out.println("Error reading file: " + file.getName());
        } finally {            if
(reader != null) {
try {
                    reader.close();
                } catch (IOException e) {
                    System.out.println("Error closing file:+ file.getName());
                }
            }
        }
    }
}
```





Q12) Write a program to demonstrate unchecked exception

INPUT

```
import java.util.Scanner;

class RAUNAKDON{
   public static void main(String[] args){
Scanner sc=new Scanner(System.in);
System.out.println("ENTER A: ");        int
a =sc.nextInt();
       System.out.println("ENTER B: ");
       int b =sc.nextInt();
       int c=a/b;
       System.out.println("DIVISION = "+c);

   }
}
```

Q13)Write a program to demonstrate creation of multiple child threads.

INPUT

```
class MultipleThreadsDemo {

 public static void main(String[] args) {
 // Create and start multiple threads
 Thread thread1 = new Thread(new MyRunnable("Thread 1"));
 Thread thread2 = new Thread(new MyRunnable("Thread 2"));
 Thread thread3 = new Thread(new MyRunnable("Thread 3"));

 thread1.start();
thread2.start();
thread3.start();
```

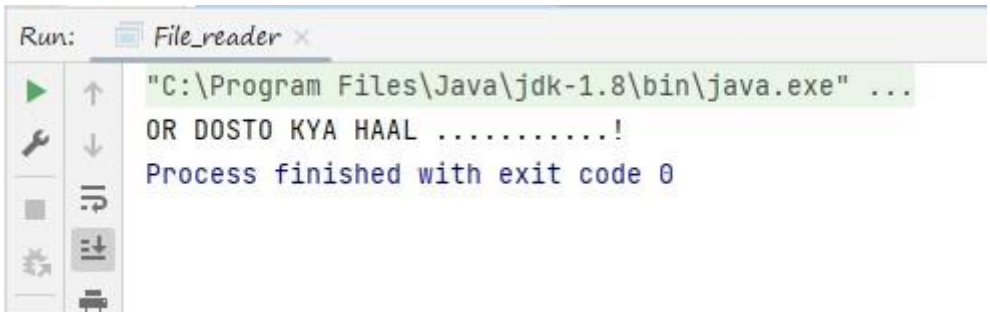
```
 }
}

class MyRunnable implements Runnable {
private String threadName;

 public MyRunnable(String threadName) {
this.threadName = threadName;
 }

 public void run() {
 System.out.println("Thread " + threadName + " is running.");

 try {
 // Simulate some work
 Thread.sleep(2000);
 } catch (InterruptedException e) {
e.printStackTrace();
 }

 System.out.println("Thread " + threadName + " finished.");
 }
}
```

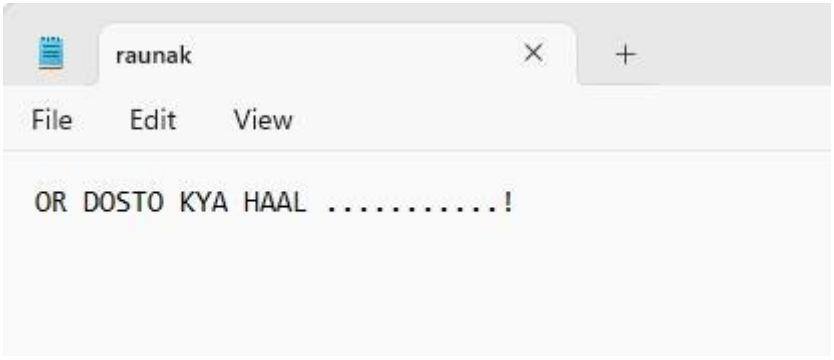
Q14) Write a program to use Byte stream class to read from a text file and display the content on the output screen.

INPUT

```
import java.io.FileReader;
```

```
import java.io.IOException;

class File_reader{
   public static void main(String[] args){
try {
        FileReader r= new
FileReader("C:\\Users\\Raunak\\IdeaProjects\\file\\raunak.txt");
try{
            int i;
            while((i=r.read())!=-1){
               System.out.print((char) i);
            }
}          finally
{
            r.close();
         }
      }
      catch (IOException e){
         System.out.print("EXCEPTION HANDELED..!");

      }
   }
}
```

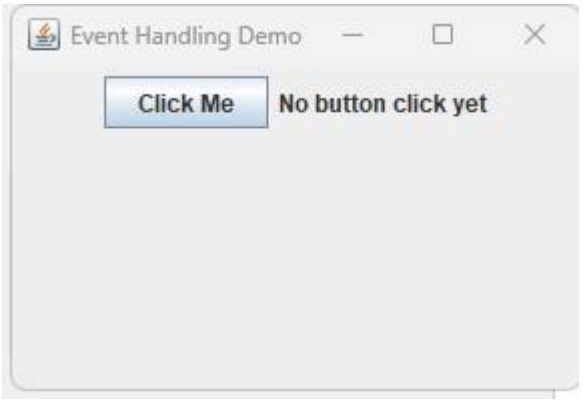


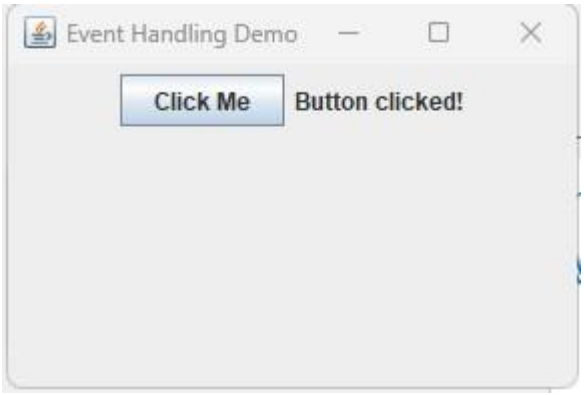


Q 15) Write a program to demonstrate any event handling. **INPUT**

```
import javax.swing.*; import
java.awt.*; import
java.awt.event.ActionEvent;
import
java.awt.event.ActionListen
er;

class EventHandlingDemo {

 private JFrame frame;
private JButton button;
private JLabel label;

 public EventHandlingDemo() {  frame = new
JFrame("Event Handling Demo");  button =
new JButton("Click Me");  label = new
JLabel("No button click yet");

 // Register an ActionListener to the button
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
label.setText("Button clicked!");
 }
 });

 // Set up the frame layout  frame.setLayout(new
FlowLayout());
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(300, 200);  frame.add(button);
frame.add(label);  frame.setVisible(true);
 }

 public static void main(String[] args) {
SwingUtilities.invokeLater(new Runnable() {
public void run() {  new
EventHandlingDemo();
 }
 });
```

```
 }
}
```



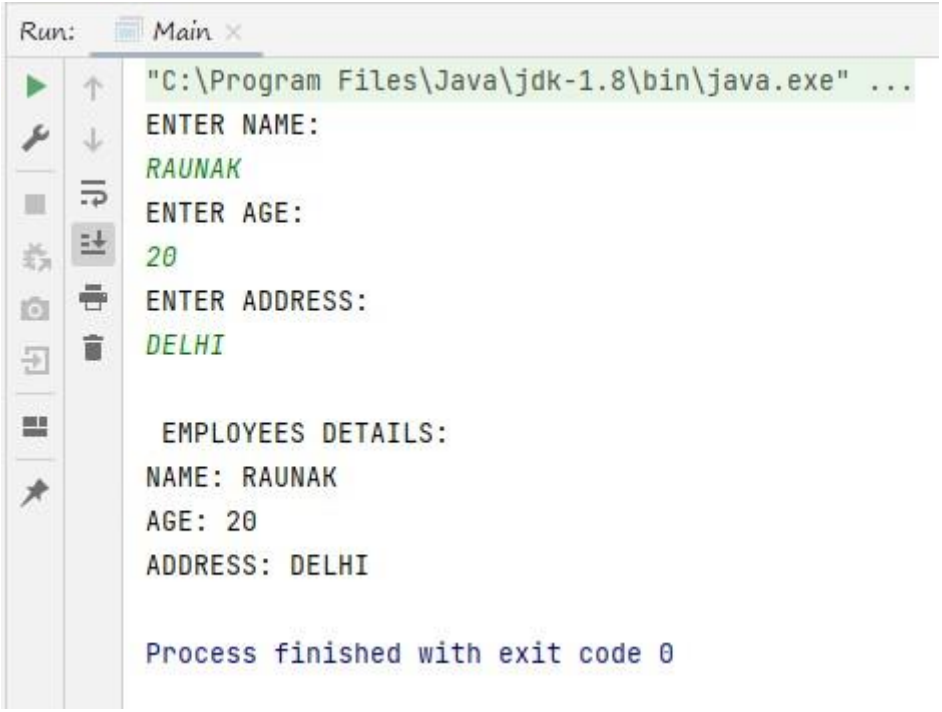


Q16) Create a class employee which have name, age and address of employee, include method:; getdata() and showdata(), getdata() takes the input from the user, showdata() display the data in following format: Name: Age: Address:

INPUT

```
import java.util.Scanner;

class Employees{
String name;     int
age;
   String address;
public void getdata(){
       Scanner a=new Scanner(System.in);
System.out.println("ENTER NAME: ");
name=a.nextLine();
       System.out.println("ENTER AGE: ");
age=a.nextInt();        a.nextLine();
       System.out.println("ENTER ADDRESS: ");
       address=a.nextLine();
```

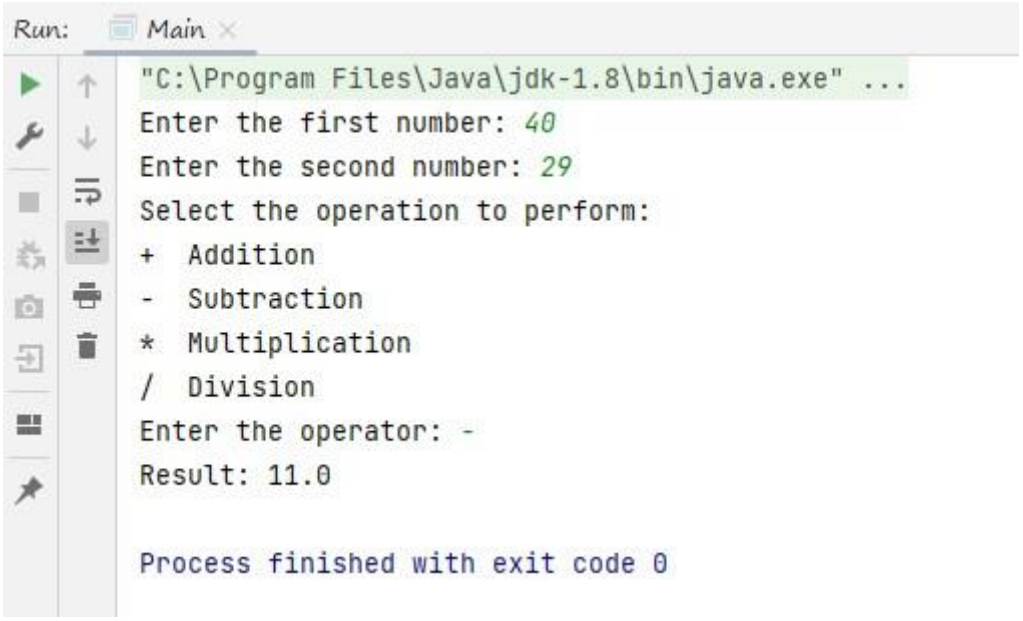
```
    }
    public void showdata(){
        System.out.println("NAME: "+name);
        System.out.println("AGE: "+age);
        System.out.println("ADDRESS: "+address);
    }
}
public class Main{     public static void
main(String[] args){         Employees e=
new Employees();         e.getdata();
        System.out.println("\n EMPLOYEES DETAILS: ");
e.showdata();
    }
}
```



Q17) Write a Java program to perform basic Calculator operations. Make a menu driven program to select operation to perform (+ - * /). Take 2 integers and perform operation as chosen by user. **INPUT**

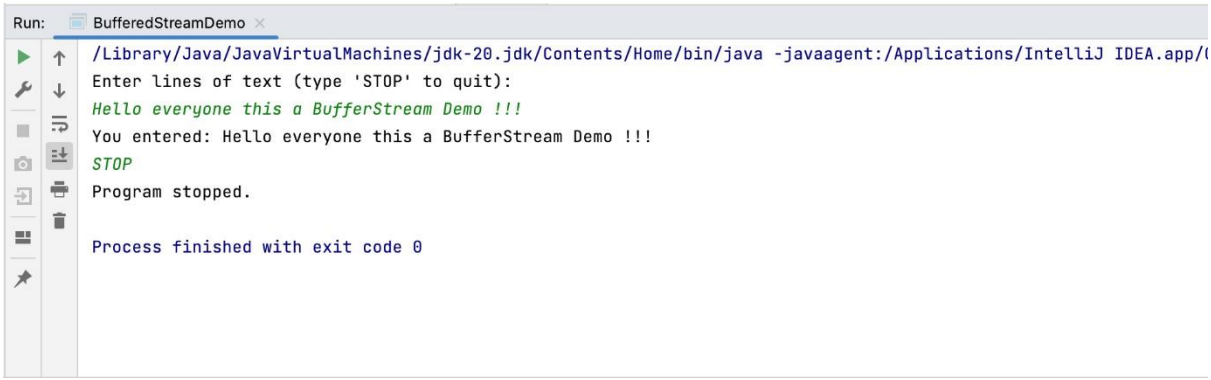
```
import java.util.Scanner; public
class Main {
    public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
```

```
int num1, num2;        char operator;        double
result;
        System.out.print("Enter the first number: ");
num1 = scanner.nextInt();
        System.out.print("Enter the second number: ");
num2 = scanner.nextInt();
        System.out.println("Select the operation to perform:");
        System.out.println("+  Addition");
        System.out.println("-  Subtraction");
        System.out.println("*  Multiplication");
        System.out.println("/              Division");
System.out.print("Enter   the   operator:   ");
operator = scanner.next().charAt(0);
        switch (operator) {
case '+':
                result = num1 + num2;
                System.out.println("Result: " + result);
break;          case '-':
                result = num1 - num2;
                System.out.println("Result: " + result);
break;          case '*':
                result = num1 * num2;
                System.out.println("Result: " + result);
break;          case '/':
                if (num2 != 0) {
                   result = (double) num1 / num2;
                   System.out.println("Result: " + result);
                } else {
                   System.out.println("Error: Division by zero is not allowed.");
                }
break;
default:
                System.out.println("Error: Invalid operator.");
        }
    }
}
```



Q18) Write a program to make use of BufferedStream to read lines from the keyboard until 'STOP' is Typed. –

```
import java.io.BufferedReader;
import java.io.IOException; import
java.io.InputStreamReader;

class BufferedStreamDemo {

 public static void main(String[] args) {
 try (BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in))) {
 String line;

 System.out.println("Enter lines of text (type 'STOP' to quit):");
while (!(line = reader.readLine()).equals("STOP")) {
 System.out.println("You entered: " + line);
 }

 System.out.println("Program stopped.");
} catch (IOException e) {
e.printStackTrace();
 }
 }
}
```

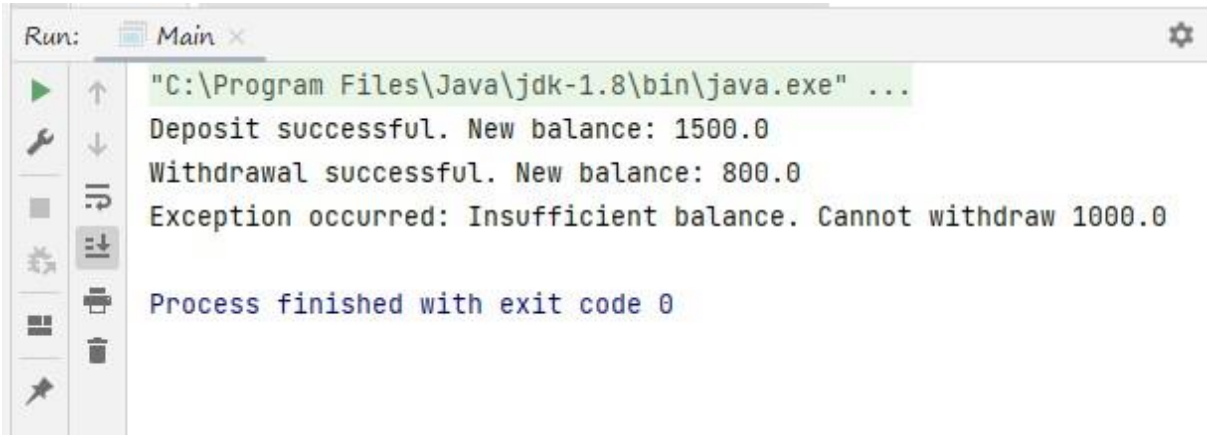


Q19) Write a program declaring a Java class called SavingsAccount with members "accountNumber" and "Balance". Provide member functions as "depositAmount ()" and ''withdraw Amount O''. If user tries to withdraw an amount greater than their balance then throw a user-defined exception.

INPUT

```
class Insufficientbalance_exception extends Exception{
public Insufficientbalance_exception(String message){
super(message);
   }
}
class SavingAccount{
int accno;     double
balance;
   public SavingAccount(int accno,double balance){
      this.accno = accno;
      this.balance = balance;

   }
   public void depositeAcc(double amount){
      balance += amount;
      System.out.println("Deposit successful. New balance: " + balance);

   }
   public void withdrawAmount(double amount)
throws Insufficientbalance_exception {
      if (amount > balance) {          throw new
Insufficientbalance_exception("Insufficient balance. Cannot withdraw " +
amount);
```

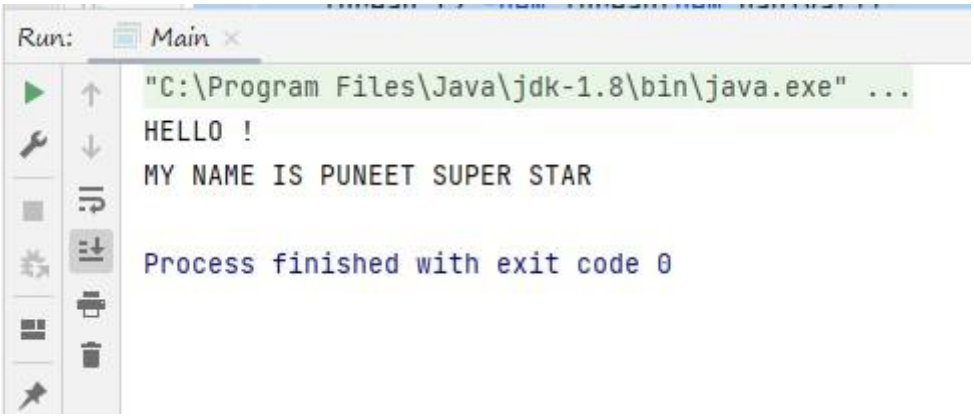
```
        }

        balance -= amount;
        System.out.println("Withdrawal successful. New balance: " + balance);
    }
}


public class Main {
    public static void main(String[] args) {
        SavingAccount account = new SavingAccount(123456, 1000.0);
         try
{
            account.depositeAcc(500.0);
account.withdrawAmount(700.0);
account.withdrawAmount(1000.0);
        } catch (Insufficientbalance_exception e) {
            System.out.println("Exception occurred: " + e.getMessage());
        }
    }
}
```



Q20) Write a program creating 2 threads using Runnable interface. Print your name in "run()" method of first class and "Hello Java" in "run()" method of second thread. **INPUT**

```
class kya_haal_hai implements Runnable{
public void run(){
        System.out.println("HELLO ! ");
    }
```

```
}
class badiya implements Runnable{
    public void run(){
        System.out.println("MY NAME IS PUNEET SUPER STAR");
    }
}
public class Main {
    public static void main(String[] args) {
        Thread t1 =new Thread(new kya_haal_hai());
        Thread t2 =new Thread(new badiya());

t1.start();
        t2.start();
    }
}
```



Q21) Write a swing application that uses atleast 5 swing controls

INPUT

```
import javax.swing.*; import
java.awt.*; import
java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class Swing_controle extends JFrame{
    JLabel label;
    JTextField textField;
JButton button;
    JCheckBox checkBox;
    JRadioButton radioButton;
    JComboBox<String> comboBox;
```

```
   public Swing_controle(){        setTitle("SWING
CONTROLS");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());
label = new JLabel("Enter your name:");
textField = new JTextField(15);         button
= new JButton("Click Me");
        checkBox = new JCheckBox("BOLO JAI MATA DI");
radioButton = new JRadioButton("Option 1");
        comboBox = new JComboBox<>(new String[]{"Option 1", "Option 2", "Option
3"});

        button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
                String name = textField.getText();
                JOptionPane.showMessageDialog(null, "Hello, " + name + "!");
            }
        });

        add(label);
add(textField);
add(button);
add(checkBox);
add(radioButton);
        add(comboBox);

        pack();
setLocationRelativeTo(null);
        setVisible(true);
    }

    public static void main(String[] args) {
SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new Swing_controle();
            }
        });
    }
}
```

Q22) Write a java program to insert and update details data in the database.

INPUT

```
import java.sql.*;

public class insert1
{
    public static void main(String args[])
    {
        String id = "id1";
        String pwd = "pwd1";
        String fullname = "geeks for geeks";
        String email = "geeks@geeks.org";

        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("
            jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");
            Statement stmt = con.createStatement();

            // Inserting data in database
            String q1 = "insert into userid values('" +id+ "', '" +pwd+
                                "', '" +fullname+ "', '" +email+ "')";
            int x = stmt.executeUpdate(q1);
            if (x > 0)
                System.out.println("Successfully Inserted");
            else
                System.out.println("Insert Failed");

            con.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

```
Output :
Successfully Registered
```

Updating data:

```java
import java.sql.*;

public class update1
{
    public static void main(String args[])
    {
        String id = "id1";
        String pwd = "pwd1";
        String newPwd = "newpwd";
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("
            jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");
            Statement stmt = con.createStatement();

            // Updating database
            String q1 = "UPDATE userid set pwd = '" + newPwd +
                    "' WHERE id = '" +id+ "' AND pwd = '" + pwd + "'";
            int x = stmt.executeUpdate(q1);

            if (x > 0)
                System.out.println("Password Successfully Updated");
            else
                System.out.println("ERROR OCCURRED :(");

            con.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

```
Output :
Password Successfully Updated
```

Q23) Write a java program to retrieve data from database and display it

INPUT

```
import java.sql.*;

public class jdbcResultSet {
   public static void main(String[] args) {
      try {
         Class.forName("org.apache.derby.jdbc.ClientDriver");
      } catch(ClassNotFoundException e) {
         System.out.println("Class not found "+ e);
      }
      try {
         Connection con = DriverManager.getConnection(
            "jdbc:derby://localhost:1527/testDb","username", "password");

         Statement stmt = con.createStatement();
         ResultSet rs = stmt.executeQuery("SELECT * FROM employee");
         System.out.println("id  name    job");

         while (rs.next()) {
            int id = rs.getInt("id");
            String name = rs.getString("name");
            String job = rs.getString("job");
            System.out.println(id+"   "+name+"    "+job);
         }
      } catch(SQLException e) {
         System.out.println("SQL exception occured" + e);
      }
   }
}
```

Output:

```
id  name   job
1   alok   trainee
2   ravi   trainee
```