INDEX

S.NO	EXPERIMENT	DATE	SIGNATURE
1	WAP to print the given pattern.		
2	WAP to implement a class 'box', three instance variable: height, width and depth. Show the functionality of 'this' keyword		
3	WAP to implement two functions. a. calculates cube of the number b. find out larger number between original no. and its reverse		
4	WAP to implementing the functionality of a stack.		
5	WAP to convert given no. to words.		
6	WAP which will read a text and count all occurrences of a particular character.		
7	WAP which read a string and print its reverse and string in alphabetical order.		
8	WAP to create try block and handle at least 3 exceptions.		
9	WAP to define exception called 'NoMatchException' that throw exception if string not equal to 'INDIA'.		
10	WAP to implement overloading and overriding.		
11	WAP of an applet that receives two numerical values from user and displays sum of these no.		
12	WAP for displaying product list along with their prices and then allow user to buy any one item with required quantity.		
13	WAP to implement multithreading.		

S.NO	EXPERIMENT	DATE	SIGNATURE
14	WAP to implement calculator using awt controls.		
15	WAP to implement the URL.		
16	WAP to implement the InetAddress.		
17	WAP for sending e-mail in Java.		
18	WAP to implement Single Client-Server Communication.		
19	WAP to create a student form web application and store data in derby database.		
20	WAP to implement the login_Id Form using JDBC.		
21	WAP to implement the SQL commands using JDBC.		
22	WAP to implement the List.		
23	WAP to implement the JTrees.		
24	WAP to implement the JTable.		
25	WAP to create the table using JDBC.		
26	WAP to implement Remote Method Invocation.		

Experiment - 1

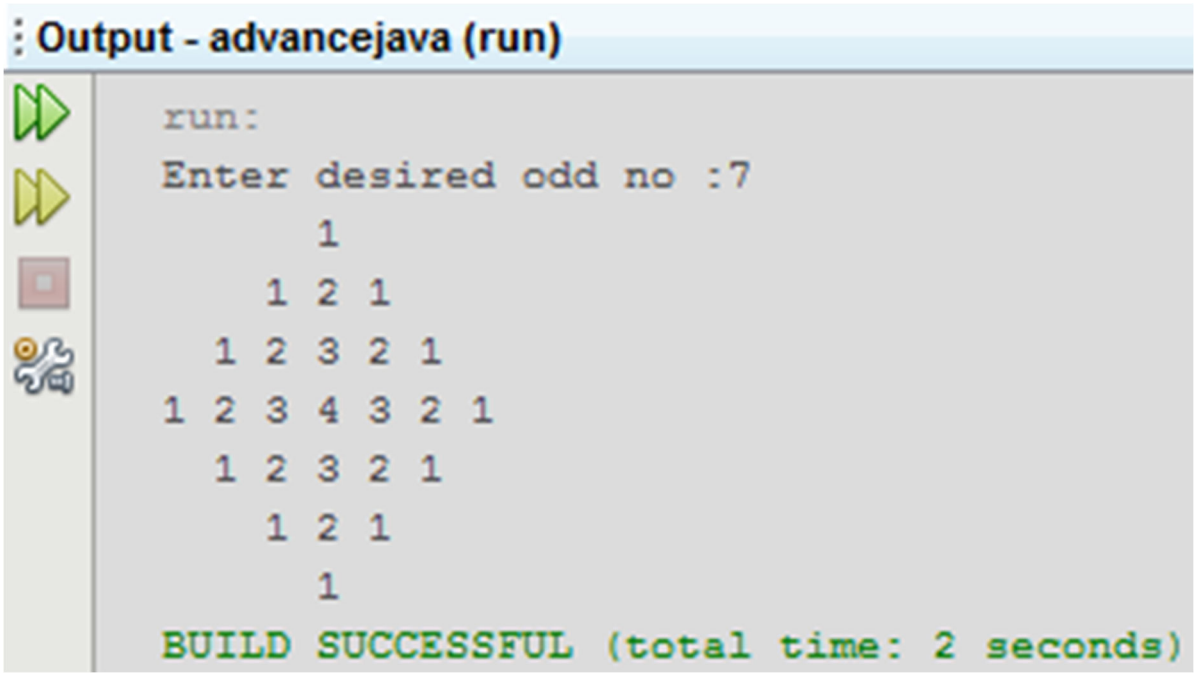
Aim: - Write a program to print the following pattern.

```
        1
    1   2   1
1   2   3   2   1
    1   2   1
        1
```

```
package advancejava;
import java.util.Scanner;
public class exp1
{
   public static void main(String []args)
   {
      System.out.print("Enter desired odd no :");
      Scanner data=new Scanner(System.in);
      int i,j,k,midline,lineno=data.nextInt();
      if(lineno%2==0)
      {   System.out.print("No is not an odd no"); return; }
      midline=lineno/2+1;
      for(i=1;i<=midline;i++)
      {
         for(k=midline-i;k>0;--k)
            System.out.print("  ");
         for(k=1;k<=i;++k)
            System.out.print(k+" ");
         for(k=i-1;k>=1;--k)
            System.out.print(k+" ");
```

```
            System.out.print("\n");
        }
        for(j=midline+1;j<=lineno;++j)
        {
            for(k=1;k<=j-midline;++k)
                System.out.print("  ");
            for(k=1;k<=lineno-j+1;++k)
                System.out.print(k+" ");
            for(k=lineno-j;k>=1;--k)
                System.out.print(k+" ");
            System.out.print("\n");
        }
    }
}
```

Output

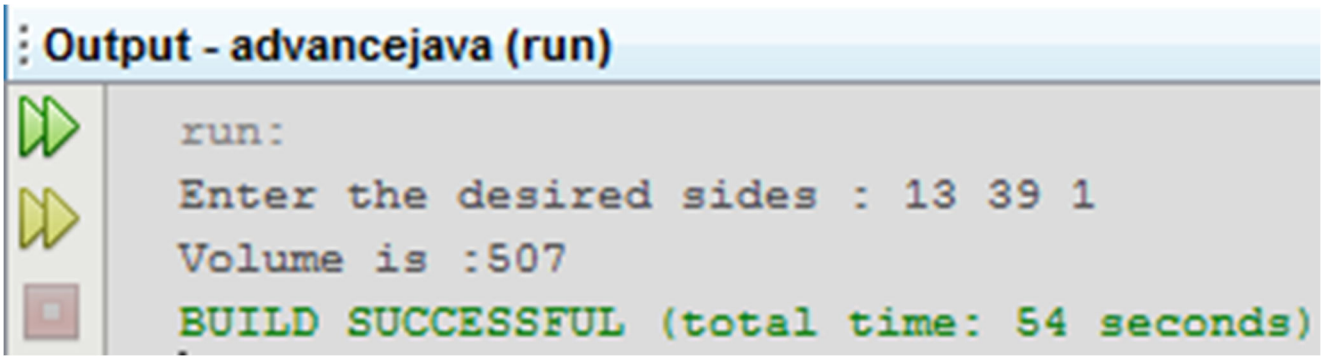


Experiment - 2

Aim: - Write a program to implement a class 'Box'. Make three instance variables of the class namely: height, width and depth. Implement a function to calculate volume. Show the functionality of 'this' keyword in the program. Also make use of overloaded constructors.

```
package advancejava;
import java.util.Scanner;
public class exp2{
   int height, width, depth;
   exp2(){}
   exp2(int height, int width, int depth) {
      this.height = height;
      this.width = width;
      this. depth = depth;
   }int boxvolume(){
      int volume=height*width*depth;
      return volume;
   }public static void main(String []args){
      Scanner data=new Scanner(System.in);
      System.out.print("Enter the desired sides : ");
      exp2 b=new exp2(data.nextInt(),data.nextInt(),data.nextInt());
      System.out.println("Volume is :"+b.boxvolume());
   }
}
```

Output



Experiment - 3

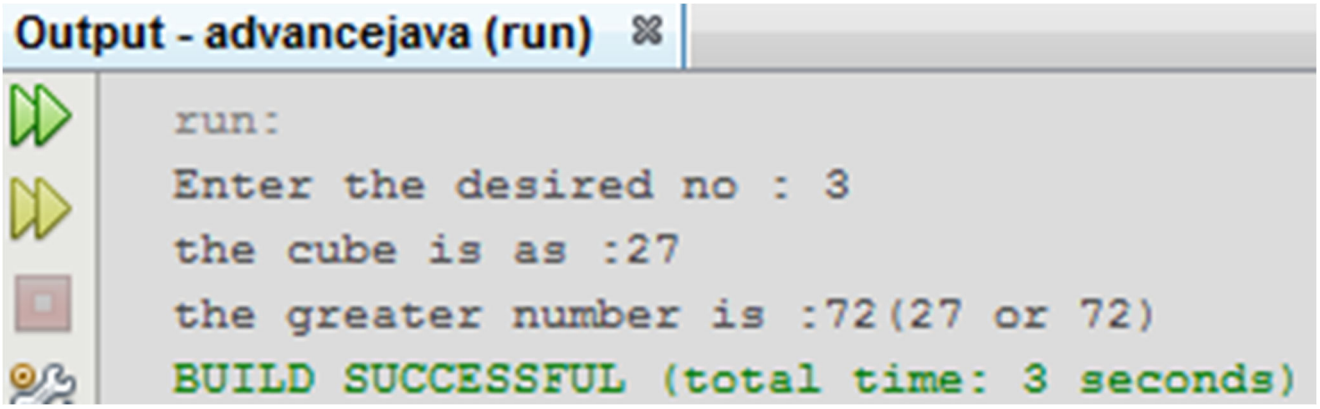
Aim: - Write a program to implement two functions. One function will calculate the cube of a number. Another function will find out the larger number between the original number and its reverse. Both the functions must return an object of class type.

```
package advancejava;
import java.util.Scanner;
public class exp3
{
   int myno, cubeno,secondno=0;
   String str;
   exp3 nocube(exp3 obj)
   {
      obj.cubeno = obj.myno*obj.myno*obj.myno;
      return obj;
   }
   exp3 greatereverse(exp3 obj)
   {
      int temp=obj.cubeno;

      while(temp>0)
      {
         secondno=secondno*10+(temp%10);
         temp/=10;
      }
      if(obj.cubeno>secondno)
         obj.str = "the greater number is :" + obj.cubeno;
      else
         obj.str= "the greater number is :" + secondno;
      return obj;
   }
   public static void main(String []args)
```

```
    {
        exp3 obj=new exp3();
        System.out.print("Enter the desired no : ");
        Scanner data=new Scanner(System.in);
        obj.myno=data.nextInt();
        exp3 obj1 = obj.nocube(obj);
        exp3 obj2 =obj.greatereverse(obj);
        System.out.println("the cube is as :"+obj1.cubeno);
        System.out.println(obj2.str+"("+obj1.cubeno+" or "+obj.secondno+")");
    }
}
```

Output



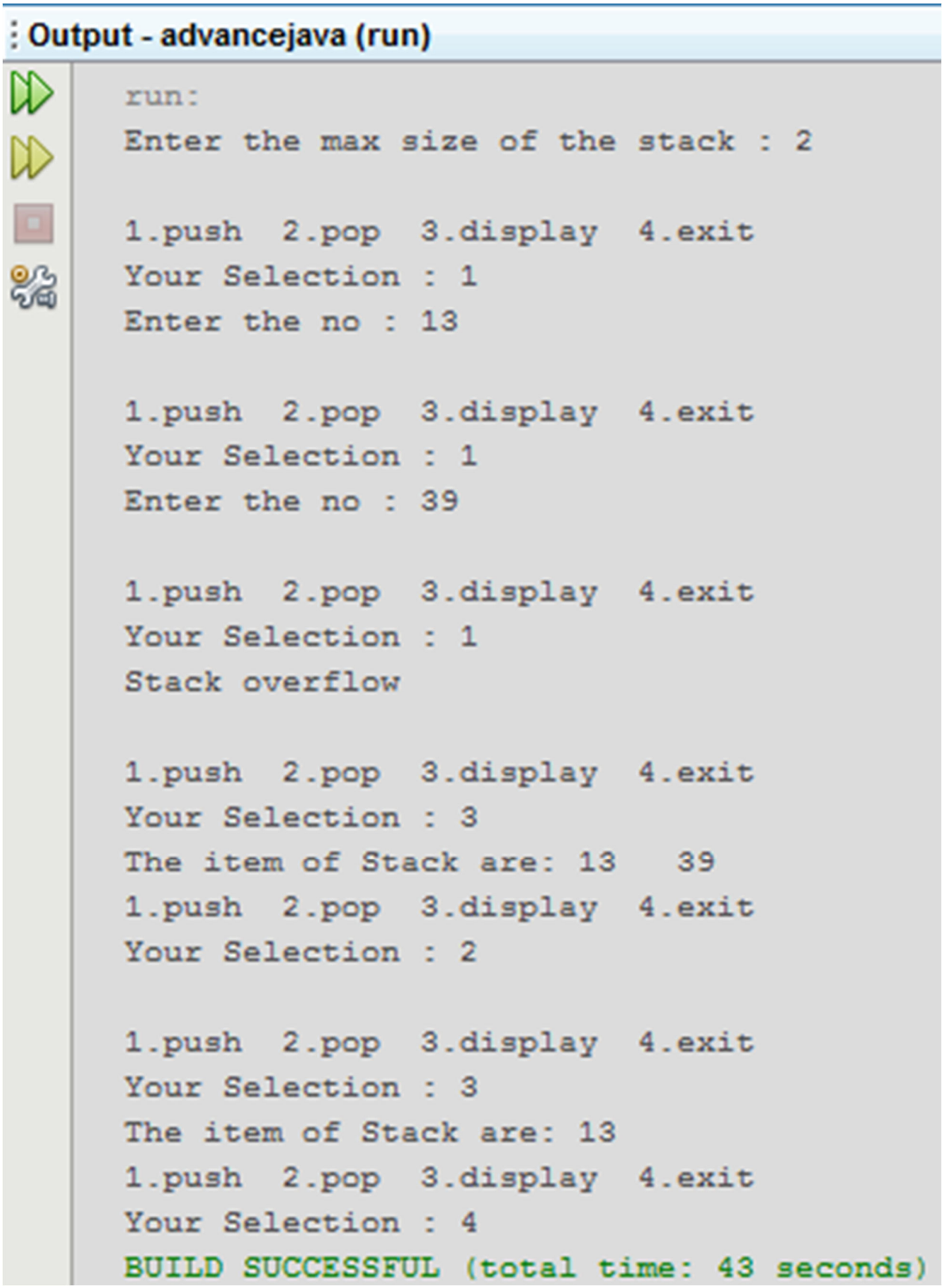
Experiment - 4

Aim: - Write a program to implement the functionality of a stack. The stack size has to be entered from command-line.

```
package advancejava;
import java.util.*;
public class exp4{
  static int maxSize,top;
  static long[] stackArray;
  public exp4(int s){
    maxSize = s;
    stackArray = new long[maxSize];
    top = -1;
  }
  public void push(long j){
     stackArray[++top] = j;
  }
  public long pop(){
     return stackArray[top--];
  }
  public static void main(String[] args){
     System.out.print("Enter the max size of the stack : ");
     Scanner data=new Scanner(System.in);
     int stackSize=data.nextInt();
     exp4 theStack = new exp4(stackSize);
     int ch,no,i;
     while(true){
        System.out.print("\n1.push  2.pop  3.display  4.exit\nYour Selection : ");
        ch=data.nextInt();
        switch(ch){
           case 1:
              if(top<maxSize-1){
                 System.out.print("Enter the no : ");
```

```
                no=data.nextInt();
                theStack.push(no);
              }
              else
                System.out.println("Stack overflow");
              break;
            case 2:
              if(top<0)
                System.out.print("Stack underflow");
              else
                theStack.pop();
              break;
            case 3:
              System.out.print("The item of Stack are: ");
              for(i=0;i<=top;i++)
                System.out.print(stackArray[i]+"   ");
              break;
            case 4:
              System.exit(0);
          }
        }
    }
}
```

Output


Output - advancejava (run)
run:
Enter the max size of the stack : 2
1.push 2.pop 3.display 4.exit
Your Selection : 1
Enter the no : 13
1.push 2.pop 3.display 4.exit
Your Selection : 1
Enter the no : 39
1.push 2.pop 3.display 4.exit
Your Selection : 1
Stack overflow
1.push 2.pop 3.display 4.exit
Your Selection : 3
The item of Stack are: 13 39
1.push 2.pop 3.display 4.exit
Your Selection : 2
1.push 2.pop 3.display 4.exit
Your Selection : 3
The item of Stack are: 13
1.push 2.pop 3.display 4.exit
Your Selection : 4
BUILD SUCCESSFUL (total time: 43 seconds)

Experiment - 5

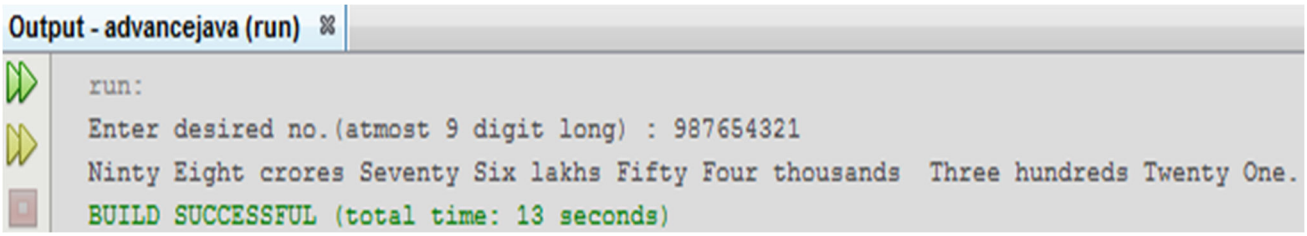
Aim: - Write a program to convert given number to words. Number should be at least 5 digits long.

```
package advancejava;
import java.util.Scanner;
public class exp5{
   static String inttostring(int tempno){
      int second=tempno%10;   int first=tempno/10;    String mystring="";
      switch(first){
         case 0: mystring="";        break;
         case 2: mystring="Twenty";  break;
         case 3: mystring="Thirty";  break;
         case 4: mystring="Forty";   break;
         case 5: mystring="Fifty";   break;
         case 6: mystring="Sixty";   break;
         case 7: mystring="Seventy"; break;
         case 8: mystring="Eighty";  break;
         case 9: mystring="Ninty";   break;
         default:
            switch(second){
               case 0:mystring="Ten";      break;
               case 1:mystring="Eleven";   break;
               case 2:mystring="Twelve";   break;
               case 3:mystring="Thirteen"; break;
               case 4:mystring="Forteen";  break;
               case 5:mystring="Fifteen";  break;
               case 6:mystring="Sixteen";  break;
               case 7:mystring="Seventeen";break;
               case 8:mystring="Eighteen"; break;
               case 9:mystring="Ninteen";  break;
            }
      }
```

```
        if(first!=1){
            switch(second){
                case 0:mystring+="";        break;
                case 1:mystring+=" One";     break;
                case 2:mystring+=" Two";     break;
                case 3:mystring+=" Three";   break;
                case 4:mystring+=" Four";    break;
                case 5:mystring+=" Five";    break;
                case 6:mystring+=" Six";     break;
                case 7:mystring+=" Seven";   break;
                case 8:mystring+=" Eight";   break;
                case 9:mystring+=" Nine";    break;
            }
        }
        return mystring;
    }
    public static void main(String []args){
        Scanner data=new Scanner(System.in);    String stringtoprint="";
        System.out.print("Enter desired no.(atmost 9 digit long) : ");
        long myno=data.nextLong();
        int tens=(int) (myno%100);      myno/=100;
        int hundreds=(int) (myno%10);   myno/=10;
        int thousands=(int) (myno%100); myno/=100;
        int lakhs=(int) (myno%100);     myno/=100;
        int crores=(int) (myno);
        if(crores!=0){
            stringtoprint+=inttostring(crores);
            stringtoprint+=" crores ";
        }if(lakhs!=0){
            stringtoprint+=inttostring(lakhs);
            stringtoprint+=" lakhs ";
        }if(thousands!=0){
```

```
            stringtoprint+=inttostring(thousands);
            stringtoprint+=" thousands ";
        }if(hundreds!=0){
            stringtoprint+=inttostring(hundreds);
            stringtoprint+=" hundreds ";
        }if(tens!=0){
            stringtoprint+=inttostring(tens);
            stringtoprint+=".";
        }System.out.println(stringtoprint);
    }
}
```

Output

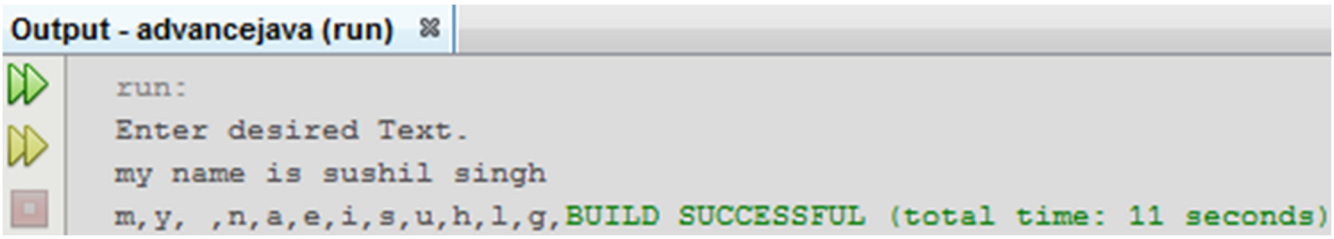


Experiment - 6

Aim: - Write a program, which will read a text and count all occurrences of a particular character.

```
package advancejava;
import java.util.Scanner;
public class exp6{
   public static void main(String []args){
      Scanner data=new Scanner(System.in);
      System.out.print("Enter desired Text.\n");
      String texttotest=data.nextLine();
      String newstring=""; int flag=1;
      for(int i=0;i<texttotest.length();i++){
         for(int j=0;j<newstring.length();j++){
            Character  t1=texttotest.charAt(i),t2=newstring.charAt(j);
            int res=t1.compareTo(t2);
            if(res==0){
               flag=0; break;
            }else
               flag=1;
         }if(flag==1)
            newstring+=texttotest.charAt(i);
      }for(int j=0;j<newstring.length();j++)
         System.out.print(newstring.charAt(j)+",");
   }
}
```

Output



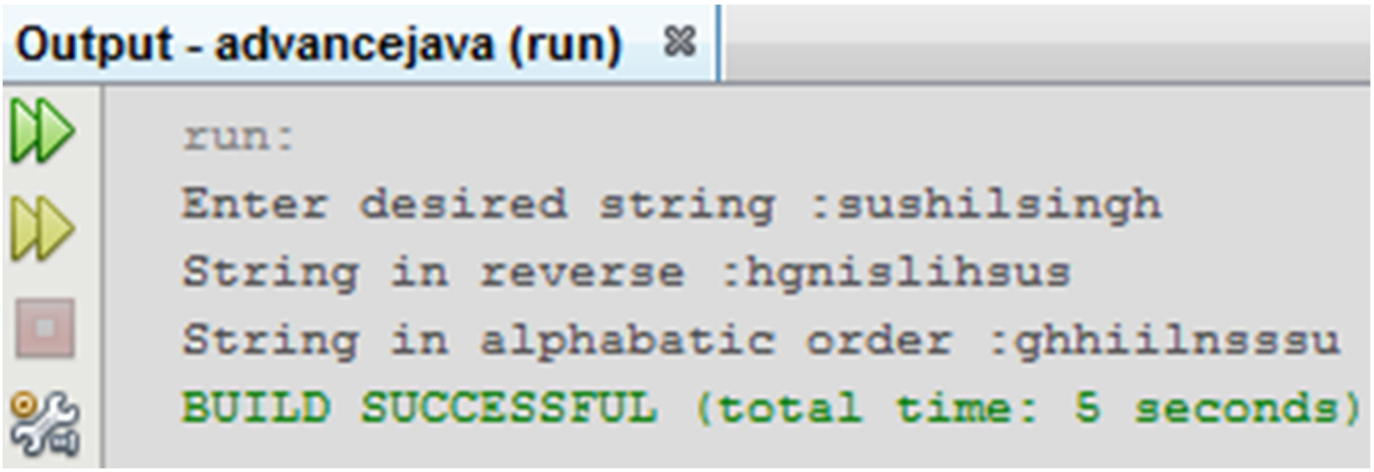
Experiment - 7

Aim: - Write a program, which will read a string and print its reverse. The program should also rewrite the string in alphabetical order.

```
package advancejava;
import java.util.Arrays;
import java.util.Scanner;
public class exp7
{
   public static void main(String []args)
   {
      Scanner data=new Scanner(System.in);
      System.out.print("Enter desired string :");
      String instring=data.next();
      StringBuilder mystring1=new StringBuilder();
      mystring1.append(instring);
      mystring1=mystring1.reverse();
      System.out.print("String in reverse :");
      for(int i=0;i<mystring1.length();i++)
         System.out.print(mystring1.charAt(i));
      Character[] chars=new Character[instring.length()];
      for(int i=0;i<chars.length;i++)
         chars[i]=instring.charAt(i);
      Arrays.sort(chars, (Character c1, Character c2) -> {
         int cmp=Character.compare(
               Character.toLowerCase(c1.charValue()),
               Character.toLowerCase(c2.charValue())
         );
         if(cmp!=0)
            return cmp;
         return Character.compare(c1, c2);
      });
      StringBuilder mystring2=new StringBuilder(chars.length);
```

```
        for(char c:chars)
            mystring2.append(c);
        System.out.print("\nString in alphabatic order :");
        for(int i=0;i<mystring1.length();i++)
            System.out.print(mystring2.charAt(i));
    }
}
```

Output

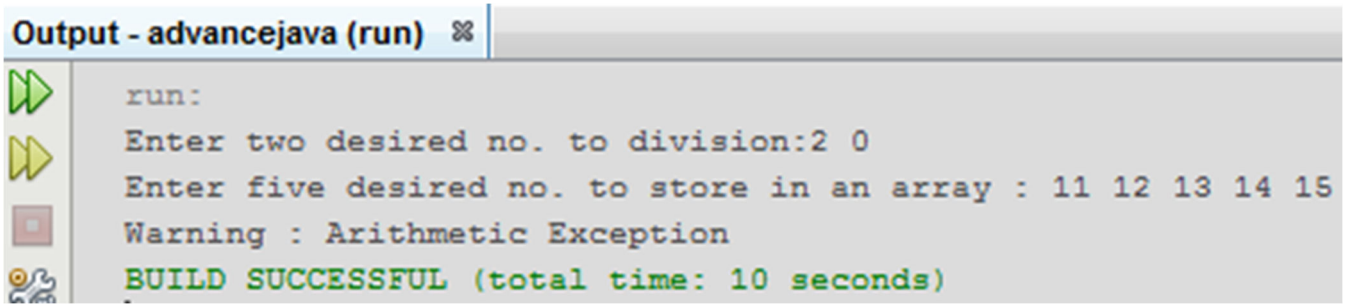


Experiment - 8

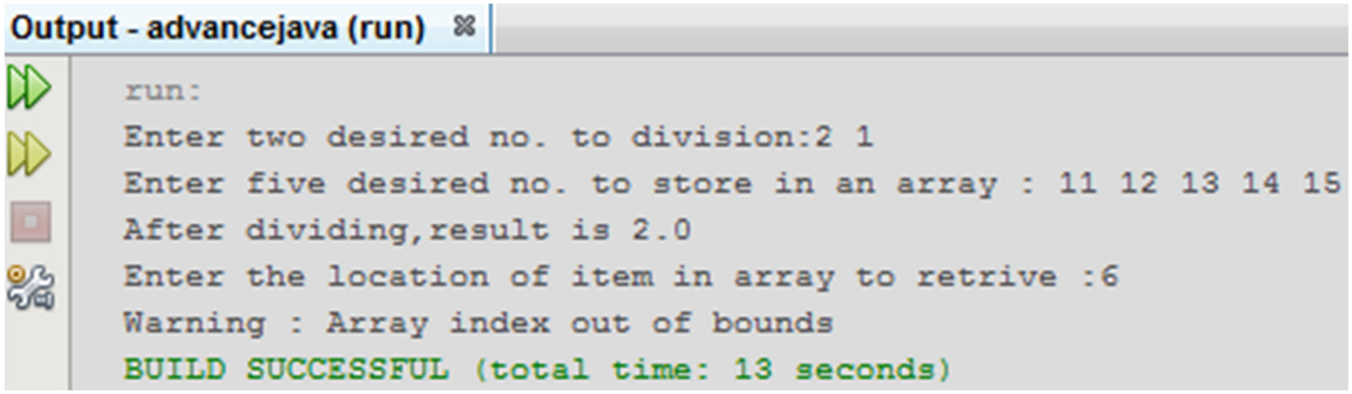
Aim: - Write a program to create a try block that is likely to generate three type of exception and then incorporate necessary catch blocks to catch and handle them appropriately.

```
package advancejava;
import java.io.*;
import java.util.*;
public class exp8{
   public static void main(String []args){
      Scanner data=new Scanner(System.in);
      System.out.print("Please enter two no. to division:");
      int A=data.nextInt();       int B=data.nextInt();
      int[] myarray=new int[5];
      System.out.print("Please enter five no to store in an array : ");
      for(int i=0;i<5;i++)
         myarray[i]=data.nextInt();
      try{
         float C=(float)A/B;
         System.out.print("After dividing,result is "+C);
         System.out.print("\nEnter the location of item in array to retrive :");
         int D=data.nextInt();
         System.out.println("The item is : "+myarray[D-1]);
         File file=new File("D:\\Important document\\file.txt");
         FileReader fr=new FileReader(file);
      }catch(ArithmeticException ex){
         System.out.print("Warning : Arithmetic Exception");
      }catch(ArrayIndexOutOfBoundsException ex){
         System.out.print("Warning : Array index out of bounds");
      }catch (FileNotFoundException ex){
         System.out.print("Warning : File not found");
      }
   }
}
```

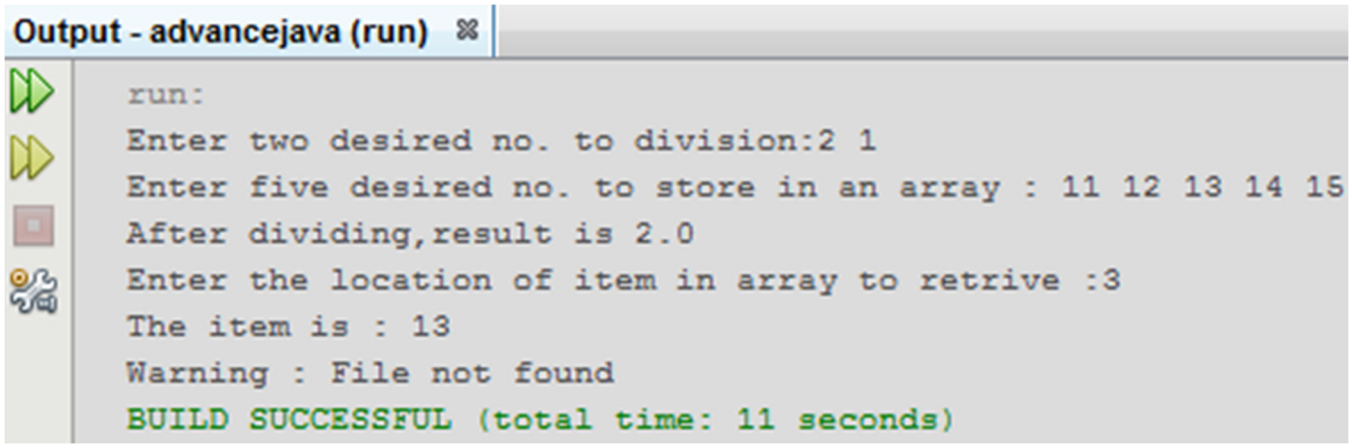
Output



Pic 8.1 :-Arithmetic exception handling.



Pic 8.2 :-Array index out of bound exception handling.



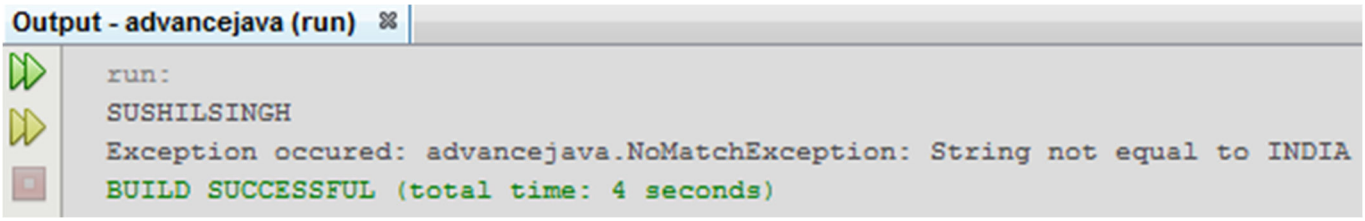
Pic 8.3 :-File exception handling.

Experiment - 9

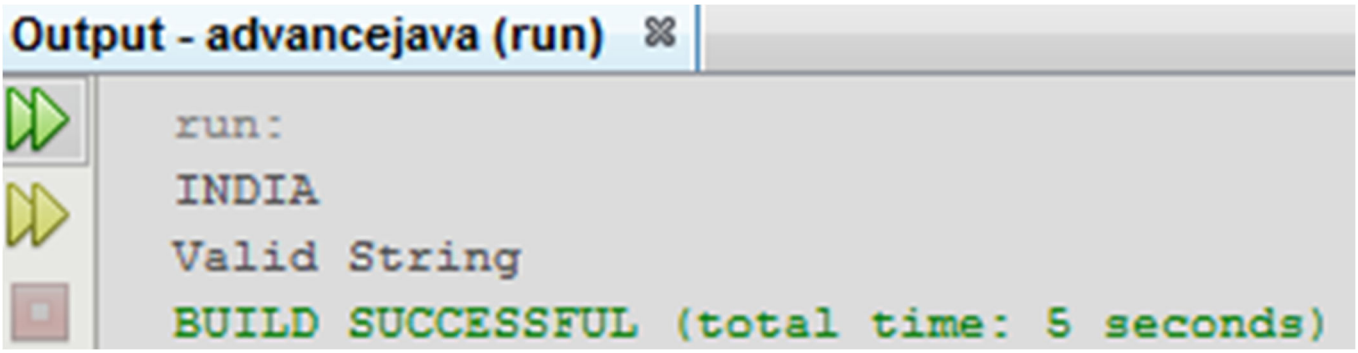
Aim: - Define an exception called 'NoMatchException' that is thrown when a string is not equal to 'INDIA'. Write a program that uses this exception.

```
package advancejava;
import java.util.Scanner;
class NoMatchException extends Exception
{
   NoMatchException(String s)
   {
      super(s);
   }
}
public class exp9
{
   static void validate(String instring)throws NoMatchException
   {
      boolean equals = instring.equals("INDIA");
      if(equals==false)
         throw new NoMatchException("String not equal to INDIA");
      else
         System.out.println("Valid String");
   }
   public static void main(String []args)
   {
      Scanner data=new Scanner(System.in);
      try{
        validate(data.next());
      }catch(Exception ex){
         System.out.println("Exception occured: "+ex);
      }
   }
}
```

Output



Pic 9.1 :-User define exception thrown.



Pic 9.2 :- User define exception not thrown.

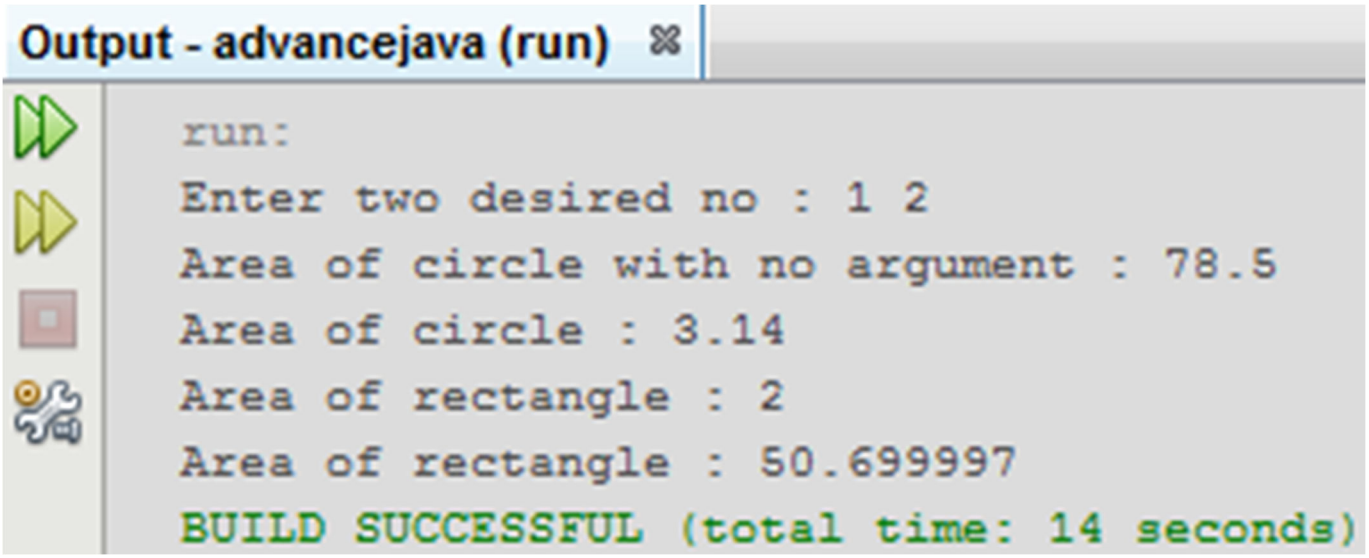
Experiment – 10A

Aim: -WAP to implement the concept of function overloading.

```
package advancejava;
import java.util.Scanner;
class areacalculator
{
   void area()
   {
      int r=5;
      System.out.println("Area of circle with no argument : "+3.14*r*r);
   }
   void area(int r)
   {
      System.out.println("Area of circle : "+3.14*r*r);
   }
   void area(int a,int b)
   {
      System.out.println("Area of rectangle : "+(a*b));
   }
   void area(float a,float b)
   {
      System.out.println("Area of rectangle : "+(a*b));
   }
}
public class exp10a
{
   public static void main(String []args)
   {
      areacalculator sushil=new areacalculator();
      Scanner data=new Scanner(System.in);
      System.out.print("Enter two desired no : ");
```

```
        int a=data.nextInt();
        int b=data.nextInt();
        sushil.area();
        sushil.area(a);
        sushil.area(a,b);
        sushil.area(1.3f,39f);
    }
}
```

Output

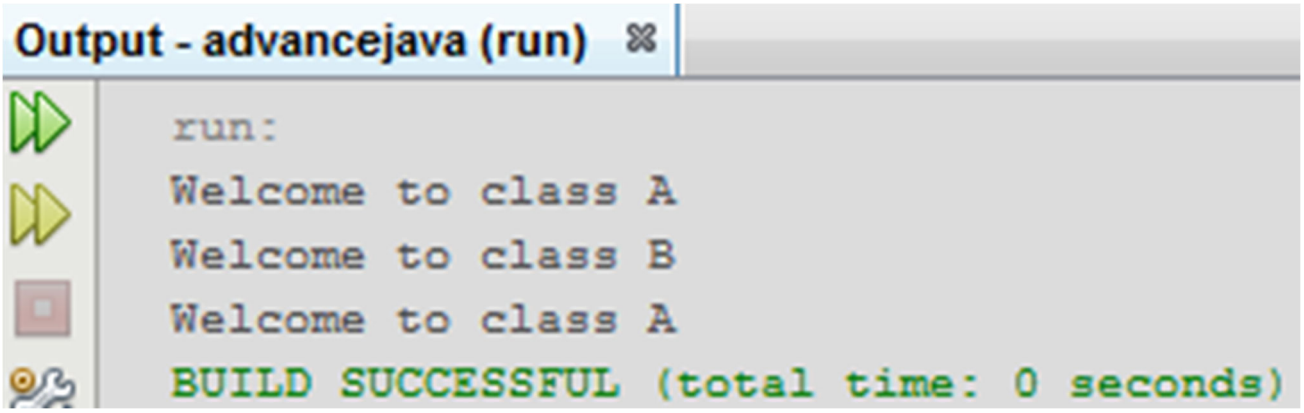


Experiment – 10B

Aim: -WAP to implement the concept of function overriding.

```
package advancejava;
class A{
    void show(){
        System.out.println("Welcome to class A");
    }
}
class B extends A{
    @Override
    void show(){
        super.show();
        System.out.println("Welcome to class B");
        super.show();
    }
}
public class exp10b{
    public static void main(String []args){
        B b=new B();
        b.show();
    }
}
```

Output



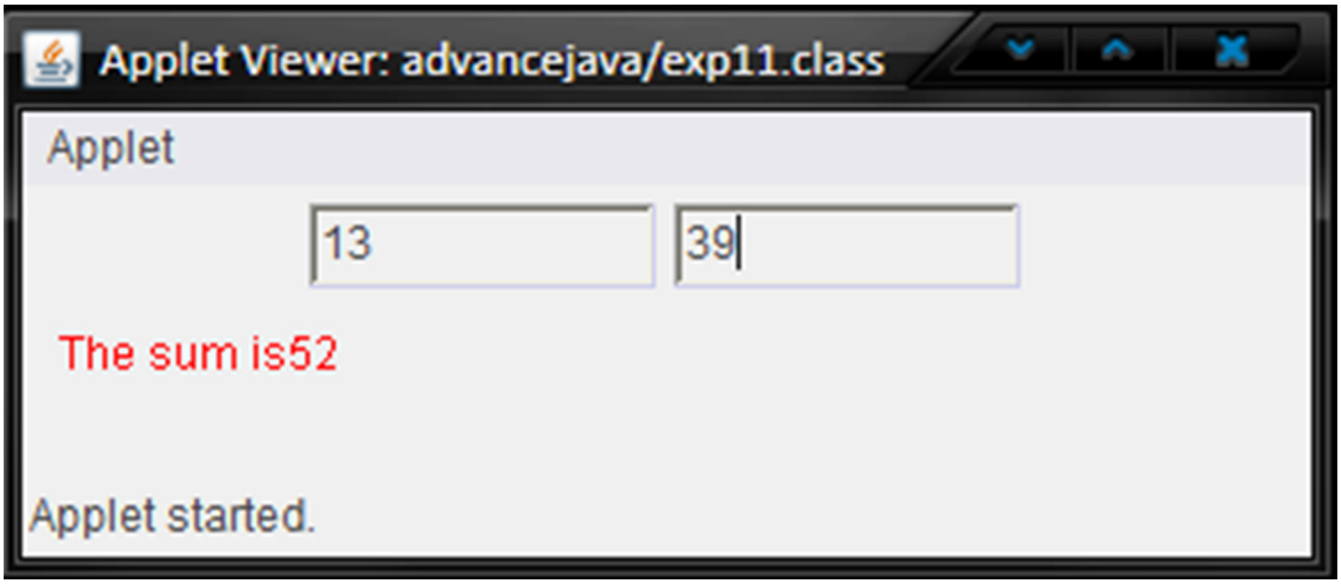
Experiment – 11

Aim: -WAP of an applet that receives two numerical values as the input from user and displays the sum of these two numbers.

```
package advancejava;
import java.applet.*;
import java.awt.*;
public class exp11 extends Applet
{
   TextField t1,t2;
   @Override
   public void init()
   {
      t1=new TextField(10);
      t2=new TextField(10);
      add(t1);
      add(t2);
      t1.setText("0");
      t2.setText("0");
   }
   @Override
   public void paint(Graphics g)
   {
      int a,b;
      g.setColor(Color.red);
      a=Integer.parseInt(t1.getText());
      b=Integer.parseInt(t2.getText());
      int c=a+b;
      String s=Integer.toString(c);
      g.drawString("The sum is"+s,10,50);
   }
   @Override
   public boolean action(Event e,Object o)
```

```
    {
        repaint();
        return true;
    }
}
```

Output



Experiment – 12

Aim: - WAP for displaying product list along with their prices and then allow user to buy any one item from them with required quantity.(Menu Driven)

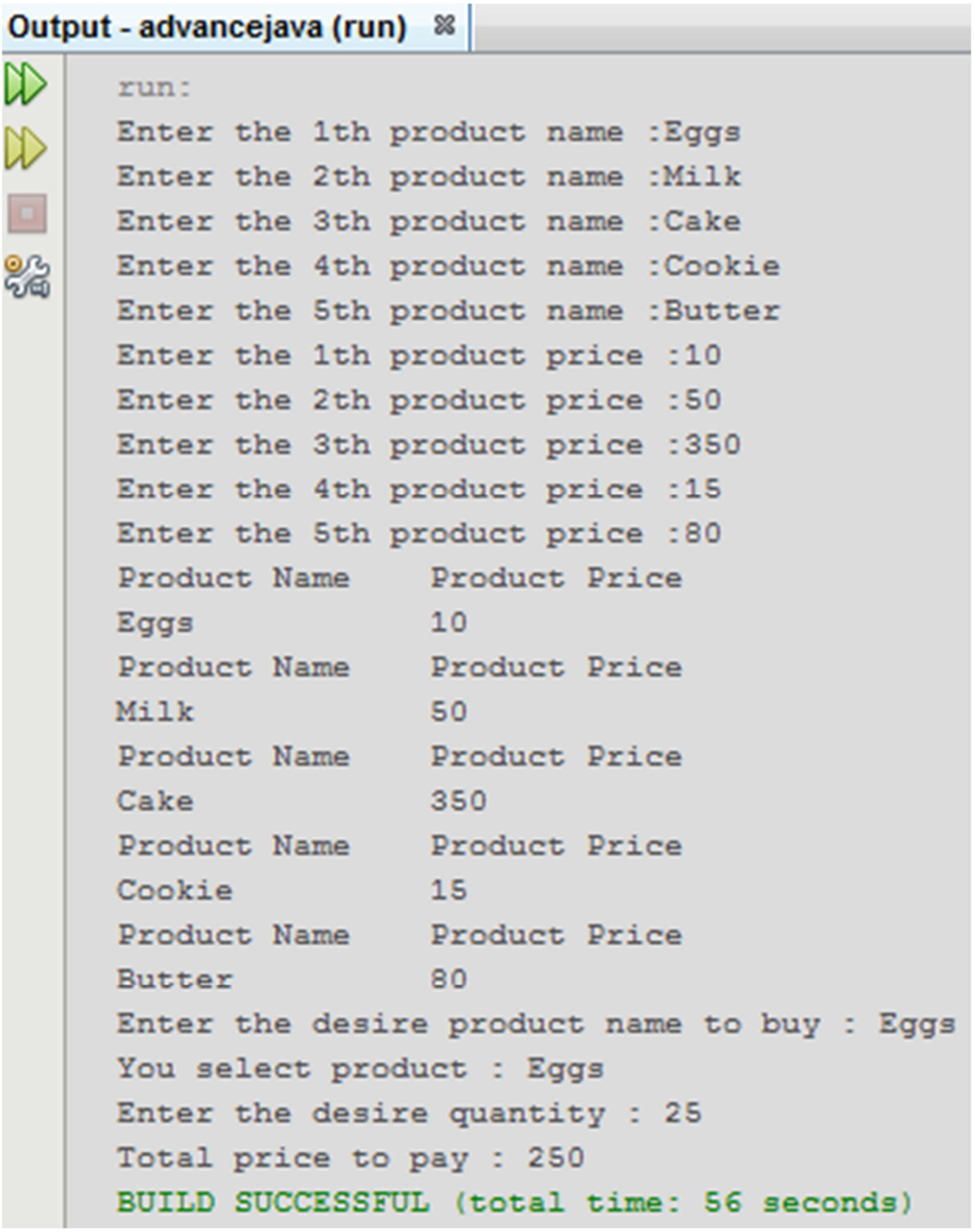
```
package advancejava;
import java.util.*;
public class exp12
{
   int[] price=new int[5];
   String[] name=new String[5];
   Scanner data=new Scanner(System.in);
   String selectitem;
   int selectprice,selectquantity,totalprice;
   public void items()
   {
      for(int i=0;i<5;i++)
      {

         System.out.print("Enter the "+(i+1)+"th product name :");
         name[i]=data.next();
      }
   }
   public void prices()
   {
      for(int i=0;i<5;i++)
      {
         System.out.print("Enter the "+(i+1)+"th product price :");
         price[i]=data.nextInt();
      }
   }
   public void display()
   {
      for(int i=0;i<5;i++)
```

```
    {
        System.out.println("Product Name\tProduct Price");
        System.out.println(name[i]+"\t\t"+price[i]);
    }
}
public void choice()
{
    System.out.print("Enter the desire product name to buy : ");
    selectitem=data.next();
    for(int i=0;i<5;i++)
    {
        if(selectitem.equals(name[i]))
        {
            System.out.println("You select product : "+name[i]);
            selectprice=price[i];
            break;
        }
    }
}
void totalprice()
{
    System.out.print("Enter the desire quantity : ");
    selectquantity=data.nextInt();
    totalprice=selectquantity*selectprice;
    System.out.println("Total price to pay : "+totalprice);
}
public static void main(String []args)
{
    exp12 sushil=new exp12();
    sushil.items();
    sushil.prices();
    sushil.display();
```

```
        sushil.choice();
        sushil.totalprice();
    }
}
```

Output



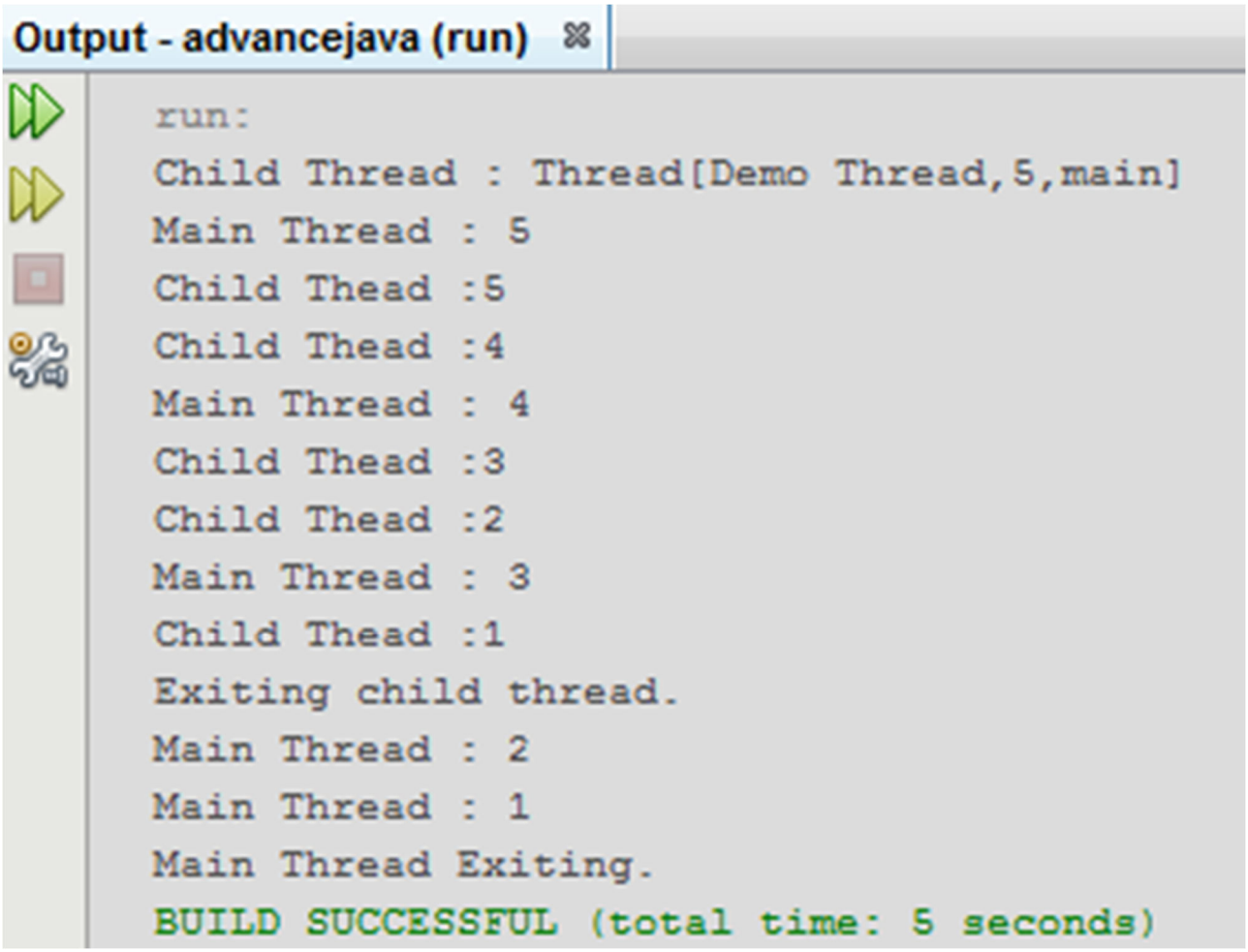
Experiment – 13

Aim: - WAP to implement multithreading(three threads using single run method).

```
package advancejava;
class newThread extends Thread
{
   newThread()
   {
      super("Demo Thread");
      System.out.println("Child Thread : "+this);
      start();
   }
   @Override
   public void run()
   {
      try
      {
         for(int i=5;i>0;i--)
         {
            System.out.println("Child Thead :"+i);
            Thread.sleep(500);
         }
      }catch(InterruptedException ex){
         System.out.println("Child Thread Interrupted.");
      }System.out.println("Exiting child thread.");
   }
}
public class exp13
{
   public static void main(String []args)
   {
      new newThread();
      try
```

```
        {
            for(int i=5;i>0;i--)
            {
                System.out.println("Main Thread : "+i);
                Thread.sleep(1000);
            }
        }catch(InterruptedException ex){
            System.out.println("Main Thread Interrupted.");
        }System.out.println("Main Thread Exiting.");
    }
}
```

Output



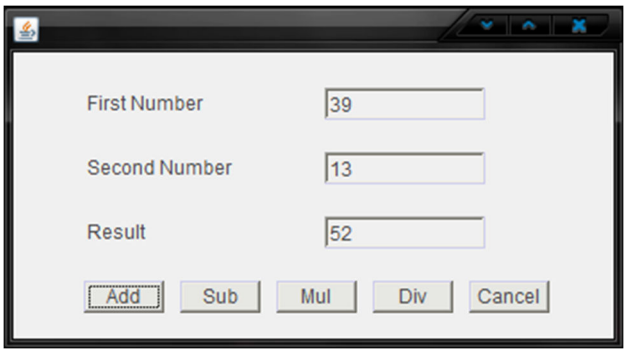
Experiment – 14

Aim: - WAP to implement the calculator using awt controls.

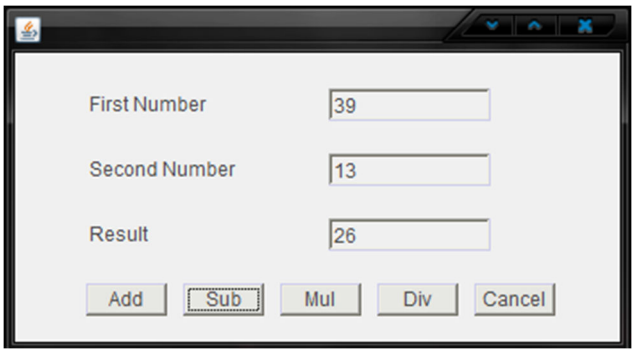
```
package advancejava;
import java.awt.event.*;
import java.awt.*;
public class exp14 implements ActionListener
{
   Frame f=new Frame();
   Label l1=new Label("First Number");
   Label l2=new Label("Second Number");
   Label l3=new Label("Result");
   TextField t1=new TextField();
   TextField t2=new TextField();
   TextField t3=new TextField();
   Button b1=new Button("Add");
   Button b2=new Button("Sub");
   Button b3=new Button("Mul");
   Button b4=new Button("Div");
   Button b5=new Button("Cancel");
   exp14()
   {
      l1.setBounds(50,100,100,20);    f.add(l1);
      l2.setBounds(50,140,100,20);    f.add(l2);
      l3.setBounds(50,180,100,20);    f.add(l3);
      t1.setBounds(200,100,100,20);   f.add(t1);
      t2.setBounds(200,140,100,20);   f.add(t2);
      t3.setBounds(200,180,100,20);   f.add(t3);
      b1.setBounds(50,250,50,20);     f.add(b1);
      b2.setBounds(110,250,50,20);    f.add(b2);
      b3.setBounds(170,250,50,20);    f.add(b3);
      b4.setBounds(230,250,50,20);    f.add(b4);
      b5.setBounds(290,250,50,20);    f.add(b5);
```

```
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        f.setLayout(null);
        f.setVisible(true);
        f.setSize(400,350);
    }
    @Override
    public void actionPerformed(ActionEvent e)
    {
        int n1=Integer.parseInt(t1.getText());
        int n2=Integer.parseInt(t2.getText());
        if(e.getSource()==b1)
            t3.setText(String.valueOf(n1+n2));
        if(e.getSource()==b2)
            t3.setText(String.valueOf(n1-n2));
        if(e.getSource()==b3)
            t3.setText(String.valueOf(n1*n2));
        if(e.getSource()==b4)
            t3.setText(String.valueOf(n1/n2));
        if(e.getSource()==b5)
            System.exit(0);
    }
    public static void main(String []args)
    {
            exp14 sushil = new exp14();
    }
}
```

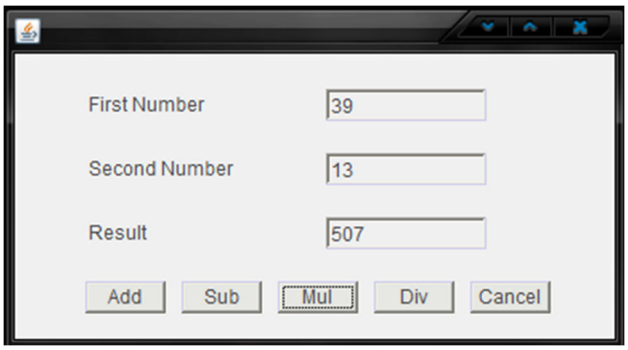
Output



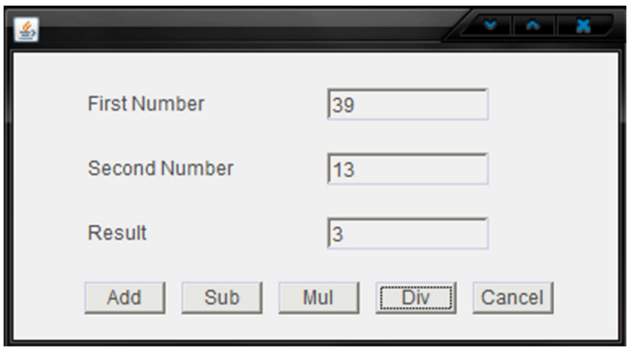
Pic 14.1 :-Addition.



Pic 14.2 :-Subtraction.



Pic 14.3 :-Multiplication.



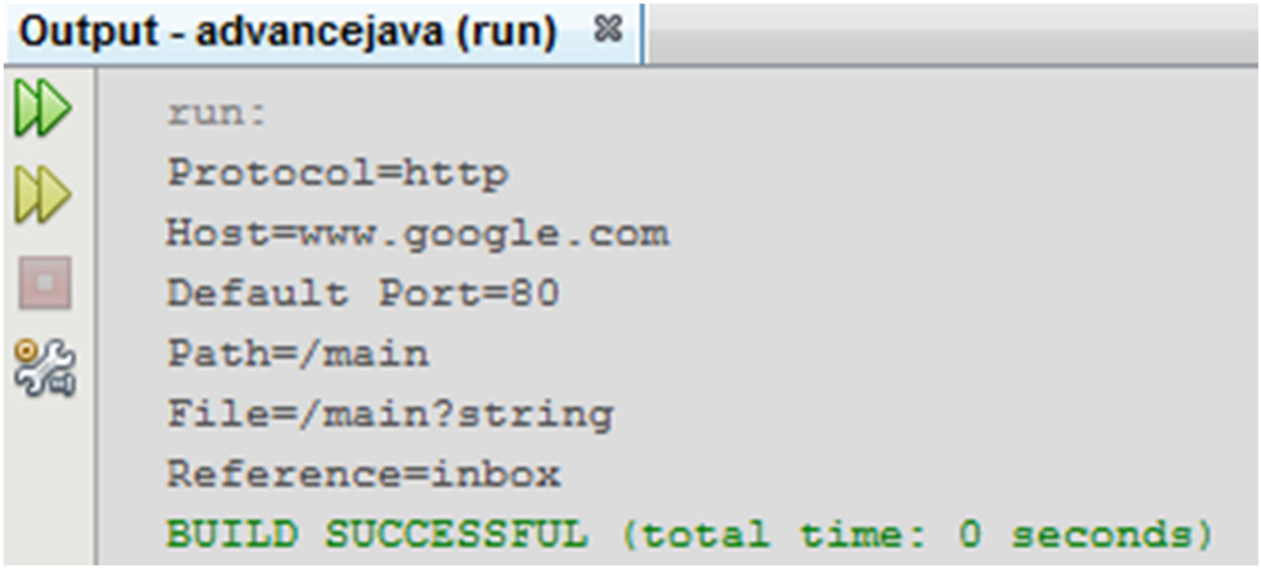
Pic 14.4 :-Division.

Experiment – 15

Aim: - WAP to implement the URL.

```
package advancejava;
import java.net.*;
public class exp15{
   public static void main(String []args){
      try{
         String str="http://www.google.com/main?string#inbox";
         URL ur=new URL(str);
         System.out.println("Protocol="+ur.getProtocol());
         System.out.println("Host="+ur.getHost());
         System.out.println("Default Port="+ur.getDefaultPort());
         System.out.println("Path="+ur.getPath());
         System.out.println("File="+ur.getFile());
         System.out.println("Reference="+ur.getRef());
      }catch(MalformedURLException e){
         System.out.println(e);
      }
   }
}
```

Output

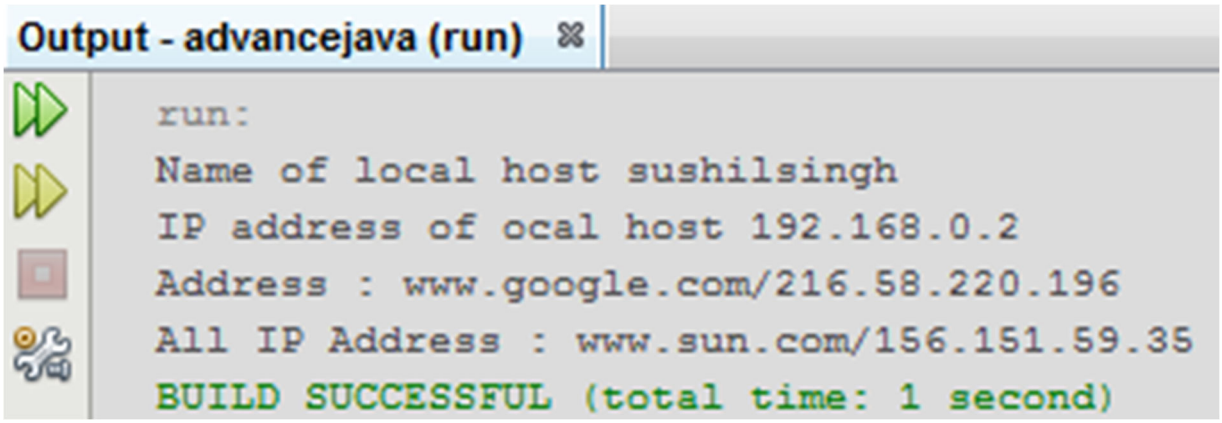


Experiment – 16

Aim: - WAP to implement the InetAddress.

```
package advancejava;
import java.net.*;
public class exp16{
   public static void main(String []args){
      try{
         InetAddress add=InetAddress.getLocalHost();
         String name=add.getHostName();
         System.out.println("Name of local host "+name);
         String add1=add.getHostAddress();
         System.out.println("IP address of ocal host "+add1);
         InetAddress address=InetAddress.getByName("www.google.com");
         System.out.println("Address : "+address);
         InetAddress add2[]=InetAddress.getAllByName("www.sun.com");
         for(int i=0;i<add2.length;i++){
            System.out.println("All IP Address : "+add2[i]);
         }
      }catch(Exception ex){
         System.out.println(ex);
      }
   }
}
```

Output



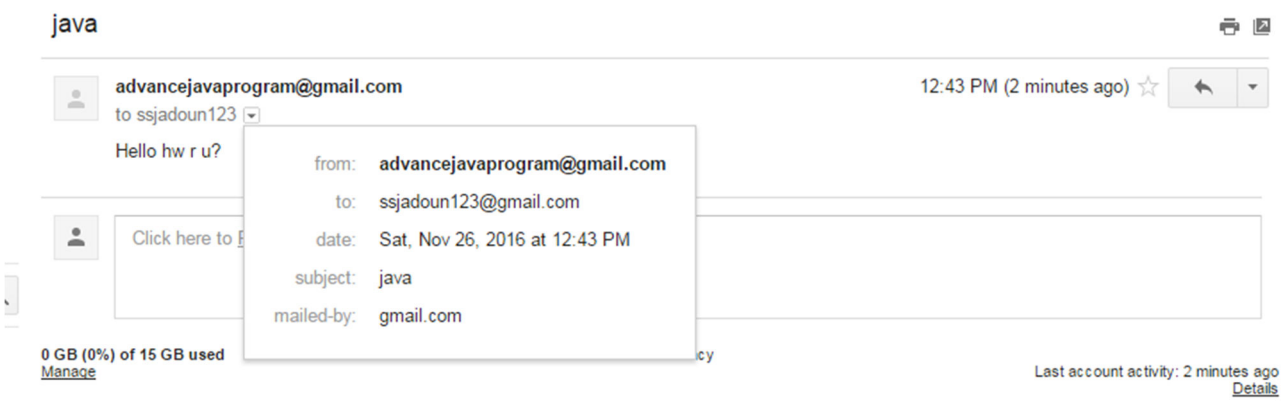
Experiment – 17

Aim: - WAP for Sending e-mail in Java.

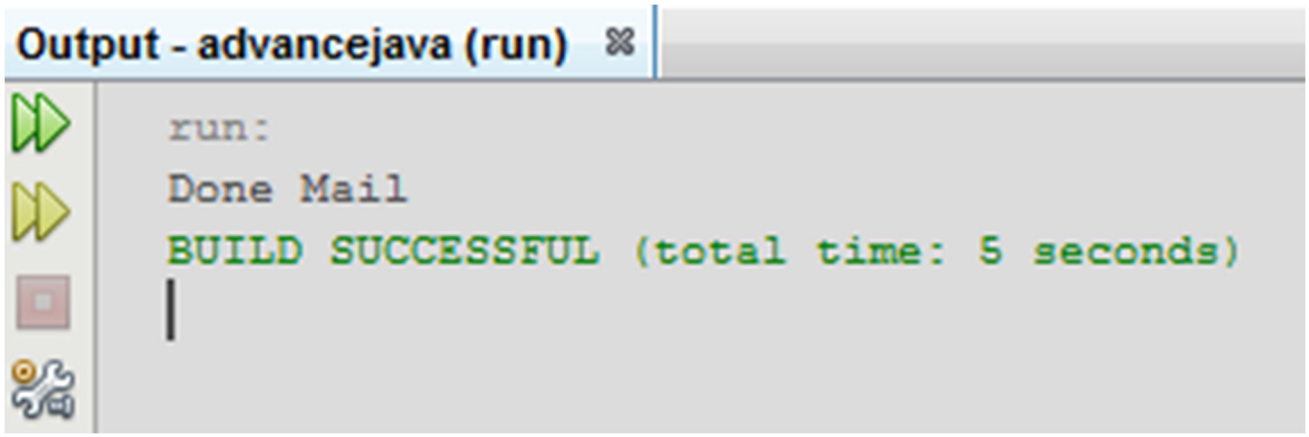
```
package advancejava;
import javax.mail.*;
import java.util.Properties;
import javax.mail.internet.*;
public class exp17{
   public static void main(String[] args){
          String host = "smtp.gmail.com";
          String user = "advancejavaprogram@gmail.com";
          String pass = "hello123123";
          String SSL_FACTORY = "javax.net.ssl.SSLSocketFactory";
          String from1 = "advancejavaprogram@gmail.com";
          String to1 = "ssjadoun123@gmail.com";
          String subject1 = "java";
          String messageText = "Hello hw r u?";
          Properties props = System.getProperties();
      props.put("mail.host", host);
          props.put("mail.transport.protocol.", "smtp");
          props.put("mail.smtp.port", "465");
          props.put("mail.smtp.socketFactory.class", SSL_FACTORY);
          Session mailSession = Session.getDefaultInstance(props, null);
          Message msg = new MimeMessage(mailSession);
      try{
         InternetAddress  fr = new InternetAddress(from1);
         msg.setFrom(fr);
         InternetAddress[] address = {new InternetAddress(to1)};
         msg.setRecipients(Message.RecipientType.TO, address);
         msg.setSubject(subject1);
         msg.setContent(messageText, "text/html");
         Transport transport = mailSession.getTransport("smtp");
         transport.connect(host, user, pass);
```

```
            transport.sendMessage(msg, msg.getAllRecipients());
            transport.close();
            System.out.println("Done Mail");
        }catch (Exception err){
            System.out.println("not done mail");
            System.out.println(err);
        }
    }
}
```

Output



Pic 17.1 :-Mail in inbox.



Pic 17.2 :-Mail has been send.

Experiment – 18

Aim: - WAP to implement Single-Server Communication.

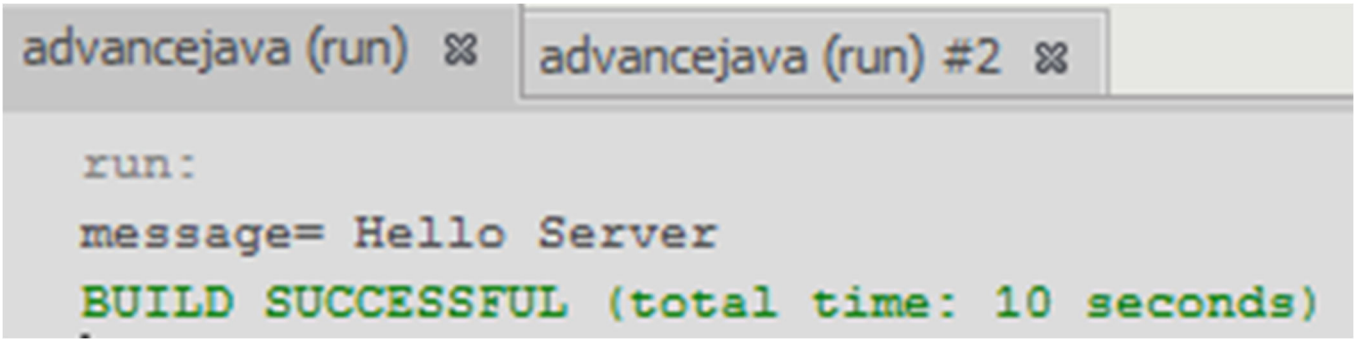
File : exp18client.java

```
package advancejava;
import java.io.*;
import java.net.*;
public class exp18client{
   public static void main(String[] args){
      try{
         Socket s=new Socket("localhost",6666);
         DataOutputStream dout=new DataOutputStream(s.getOutputStream());
         dout.writeUTF("Hello Server");
         dout.flush();          dout.close();          s.close();
      }catch(Exception e){
         System.out.println(e);
} } }
```

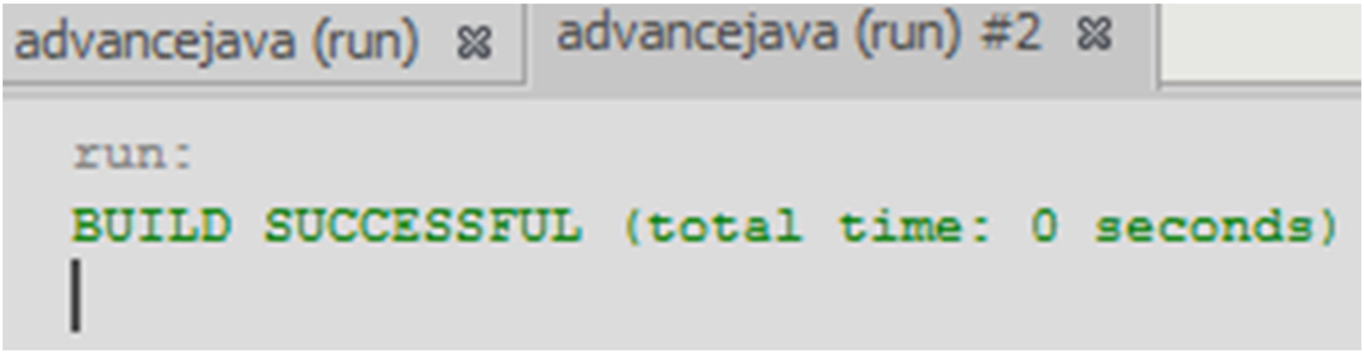
File : exp18server.java

```
package advancejava;
import java.io.*;
import java.net.*;
public class exp18server{
   public static void main(String[] args){
      try{
         ServerSocket ss=new ServerSocket(6666);
         Socket s=ss.accept();
         DataInputStream dis=new DataInputStream(s.getInputStream());
         String  str=(String)dis.readUTF();
         System.out.println("message= "+str);      ss.close();
      }catch(Exception e){
         System.out.println(e);
} } }
```

Output



Pic 18.1 :- File exp18server.java.



Pic 18.2 :- File exp18client.java.

Experiment – 19

Aim: - WAP to create a student form web application and store the data in derby database (make use of java beans).

File :index.jsp

```
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
   </head>
   <body>
      <h1 align="center">Student Form</h1>
      <form action="hello.jsp" method="POST">
         <table border="1" align="center">
            <tbody>
               <tr>
                  <td>Student Name</td>
                  <td><input type="text" name="name" value="" /></td>
               </tr>
               <tr>
                  <td>Mail ID</td>
                  <td><input type="text" name="mail" value="" /></td>
               </tr>
               <tr>
                  <td colspan="2" align="center"><input type="submit" value="submit" /></td>
               </tr>
            </tbody>
         </table>
      </form>
   </body>
</html>
```

File :hello.jsp

```
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
   </head>
   <body>
      <h1></h1>
      <jsp:useBean id="myBean" scope="session" class="org.geeks.Second" />
      <jsp:setProperty name="myBean" property="name" />
      <jsp:setProperty name="myBean" property="mail" />
      <table border="1" align="center">
         <thead>
            <tr>
               <td>Form Values</td>
            </tr>
         </thead>
         <tbody>
            <tr>
               <td><jsp:getProperty name="myBean" property="name" /></td>
            </tr>
            <tr>
               <td><jsp:getProperty name="myBean" property="mail" /></td>
            </tr>
         </tbody>
      </table>
            <%
               int a= myBean.store();
               if(a==1){
                  out.print("Elements Stored");
               }else{
```

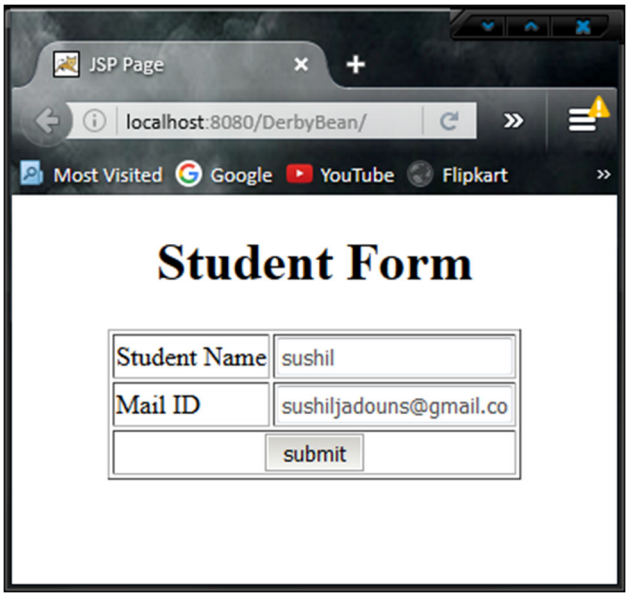
```
                    out.print("Elements Not Stored");
                }
            %>
    </body>
</html>
```

File :Second.java

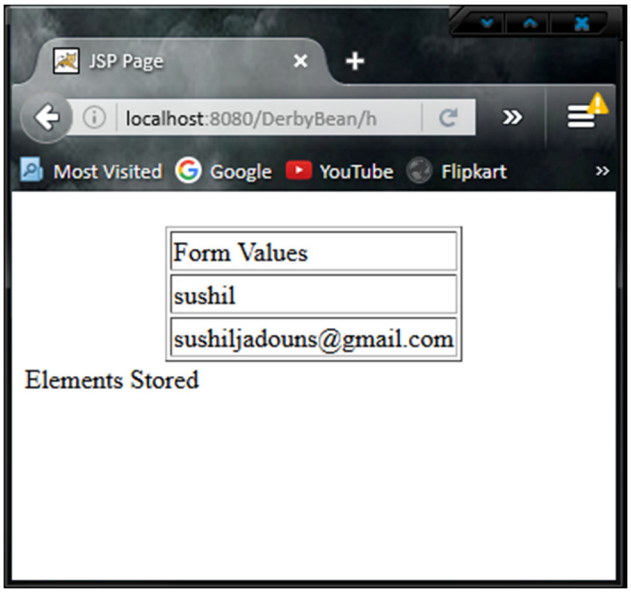
```
package org.geeks;
import java.sql.*;
public class Second {
    private String name;
    private String mail;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getMail() {
        return mail;
    }
    public void setMail(String mail) {
        this.mail = mail;
    }
    public int store() throws ClassNotFoundException, SQLException
    {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        String url="jdbc:derby://localhost:1527/sample;create=true;user=app;password=app";
        Connection con=DriverManager.getConnection(url);
        PreparedStatement ps=con.prepareStatement("insert into studentdata(name,mail) values(?,?)");
        ps.setString(1,name);
        ps.setString(2,mail);
        int a = ps.executeUpdate();
```

```
        if(a==1)
            return a;
        else
            return a;
    }
}
```

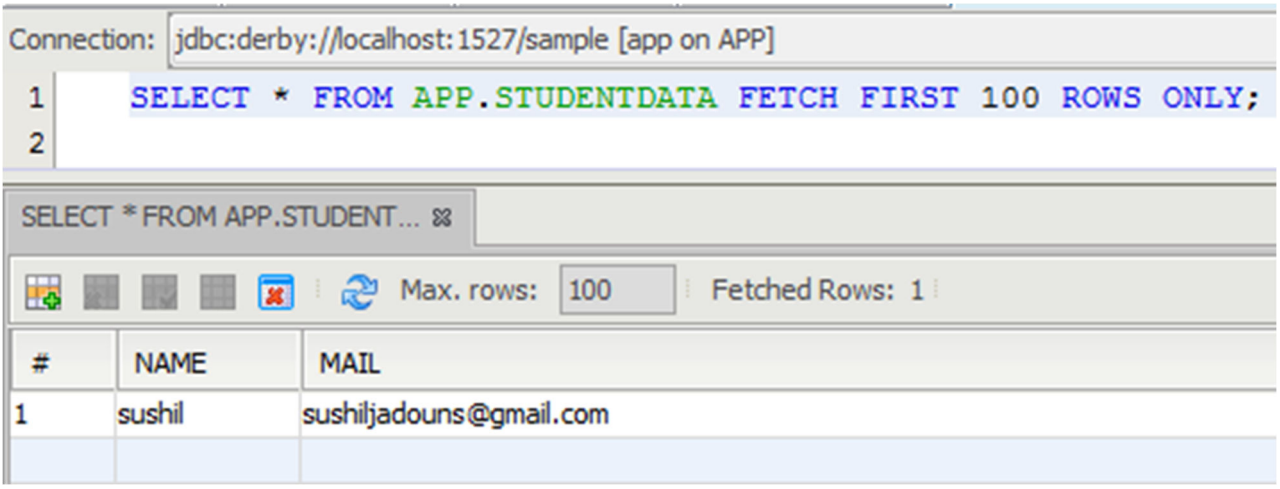
Output



Pic 19.1: -Submitting data.



Pic 19.2:-Displaying data.



Pic 19.3: -Data Stored in Derby Database.

Experiment – 20

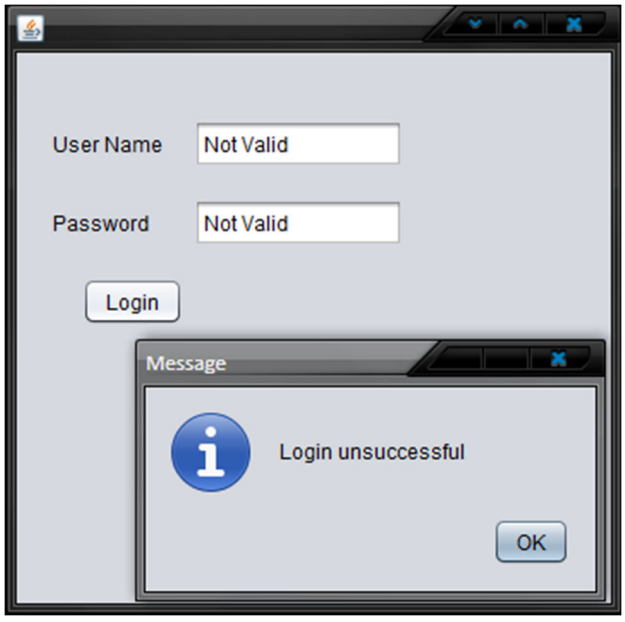
Aim: - WAP to implement the Login_Id Form using JDBC.

```
package advancejava;
import java.sql.*;
import java.util.*;
import javax.swing.JOptionPane;
public class exp20new extends javax.swing.JFrame {
   public exp20() {
      initComponents();
   }
   private void initComponents() {
                  //Autogenerated code
   }
   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
   try{
      Class.forName("java.sql.DriverManager");
      Connection con = (Connection)
      DriverManager.getConnection("jdbc:mysql://localhost:3306/advancejava","root","1234");
      Statement stmt = con.createStatement();
      ResultSet rs;
      String query = "select*from exp20";
      rs= stmt.executeQuery(query);
      while(rs.next()){

if(((jTextField1.getText()).equals(rs.getString(1)))&&((jTextField2.getText()).equals(rs.getString(2)))) {
            JOptionPane.showMessageDialog(this,"login successful");
            System.exit(0);
         }
      }JOptionPane.showMessageDialog(this,"Login unsuccessful");
      System.exit(0);
      con.close();
   }catch (ClassNotFoundException | SQLException ex) {
```

```
        Logger.getLogger(exp20new.class.getName()).log(Level.SEVERE, null, ex);
    }
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new exp20new().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration
}
```

Output



Pic 20.1: -Login unsuccessful.

Pic 20.1:-Login successful.

Experiment – 21

Aim: - WAP to implement the SQL commands using JDBC.

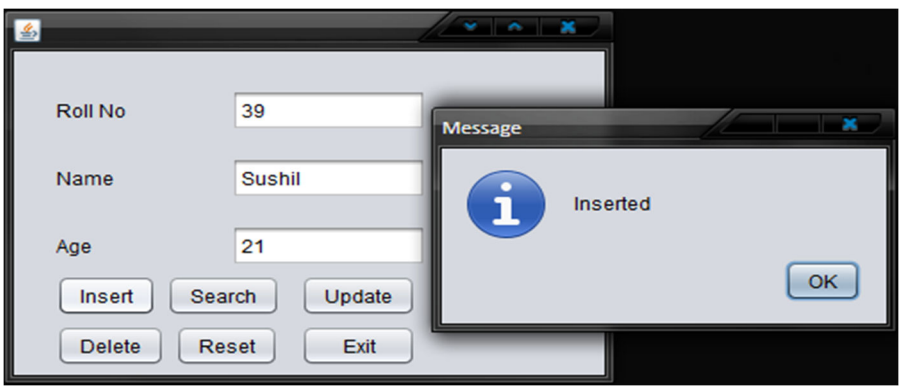
```
package advancejava;
import java.sql.*;
import java.util.logging.*;
import javax.swing.JOptionPane;
public class exp21new extends javax.swing.JFrame {
    public exp21new() {
        initComponents();
    }
            //Autogenerated code
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String rollno=rn.getText();
        String name=n.getText();
        String age=a.getText();
        String query="Insert into exp21 values("+rollno+",'"+name+"',"+age+");";
        try{
            Class.forName("java.sql.DriverManager");
            Connection con=(Connection)
                    DriverManager.getConnection("jdbc:mysql://localhost:3306/advancejava","root","1234");
            Statement stmt=con.createStatement();
            int rowsEffected = stmt.executeUpdate(query);
            if(rowsEffected==0)
              JOptionPane.showMessageDialog(this, "Please Try Again");
            else
              JOptionPane.showMessageDialog(this,"Inserted");
        } catch (SQLException | ClassNotFoundException ex) {
            Logger.getLogger(exp21new.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    String rollno=rn.getText();
```

```
String query="Select*from exp21 where rollno="+rollno+";";
try{
    Class.forName("java.sql.DriverManager");
    Connection con=(Connection)
        DriverManager.getConnection("jdbc:mysql://localhost:3306/advancejava","root","1234");
    Statement stmt=con.createStatement();
    ResultSet rs =  stmt.executeQuery(query);
      if(rs.next())
      {
        String name=rs.getString("name");
        String age=rs.getString("age");
        n.setText(name);
        a.setText(age);
        JOptionPane.showMessageDialog(this," Record Found ");
      }else
        JOptionPane.showMessageDialog(this," No Record Found ");
  } catch (SQLException | ClassNotFoundException ex) {
    Logger.getLogger(exp21new.class.getName()).log(Level.SEVERE, null, ex);
  }
}
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
  String rollno=rn.getText();
  String name=n.getText();
  String age=a.getText();
  String query="Update exp21 set name='"+(name)+"',age="+(age)+" where rollno="+(rollno)+";";
  try{
    Class.forName("java.sql.DriverManager");
    Connection con=(Connection)
        DriverManager.getConnection("jdbc:mysql://localhost:3306/advancejava","root","1234");
    Statement stmt=con.createStatement();
    int rowsEffected = stmt.executeUpdate(query);
    if(rowsEffected==0)
```

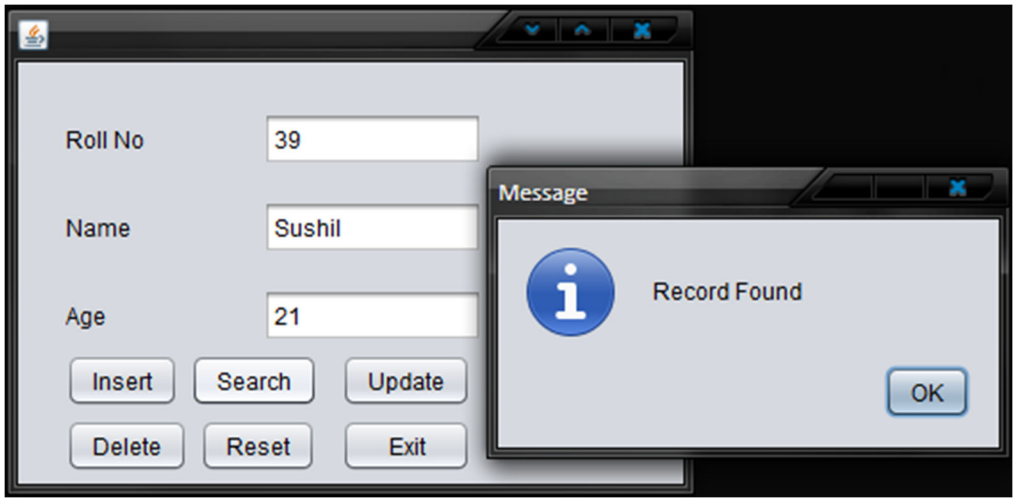
```
            JOptionPane.showMessageDialog(this, "Please Try Again");
        else
            JOptionPane.showMessageDialog(this,"Inserted");
    } catch (SQLException | ClassNotFoundException ex) {
        Logger.getLogger(exp21new.class.getName()).log(Level.SEVERE, null, ex);
    }
}
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    String rollno=rn.getText();
    String query="Delete from exp21 where rollno ="+(rollno)+";";
    try{
        Class.forName("java.sql.DriverManager");
        Connection con=(Connection)
                DriverManager.getConnection("jdbc:mysql://localhost:3306/advancejava","root","1234");
        Statement stmt=con.createStatement();
        int rowsEffected = stmt.executeUpdate(query);
        if(rowsEffected==0)
            JOptionPane.showMessageDialog(this, "Please Try Again");
        else
            JOptionPane.showMessageDialog(this,"Deleted");
    } catch (SQLException | ClassNotFoundException ex) {
        Logger.getLogger(exp21new.class.getName()).log(Level.SEVERE, null, ex);
    }
}
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    n.setText("");
    rn.setText("");
    a.setText("");
}
private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0);
}
```

```
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new exp21new().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JTextField a;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JButton jButton7;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JTextField n;
    private javax.swing.JTextField rn;
    // End of variables declaration
}
```

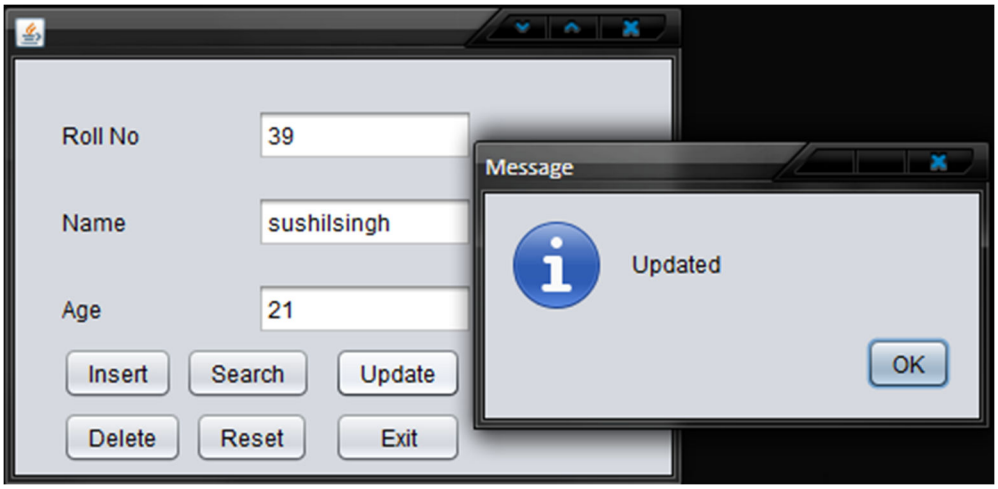
Output



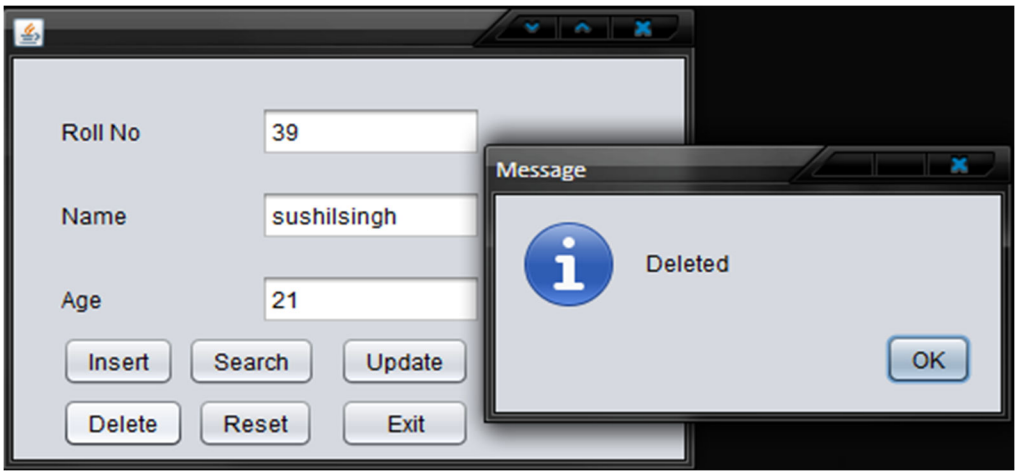
Pic 21.1 :-Insertion.



Pic 21.2 :-Searching.



Pic 21.3 :-Updation.



Pic 21.4 :-Deletion.

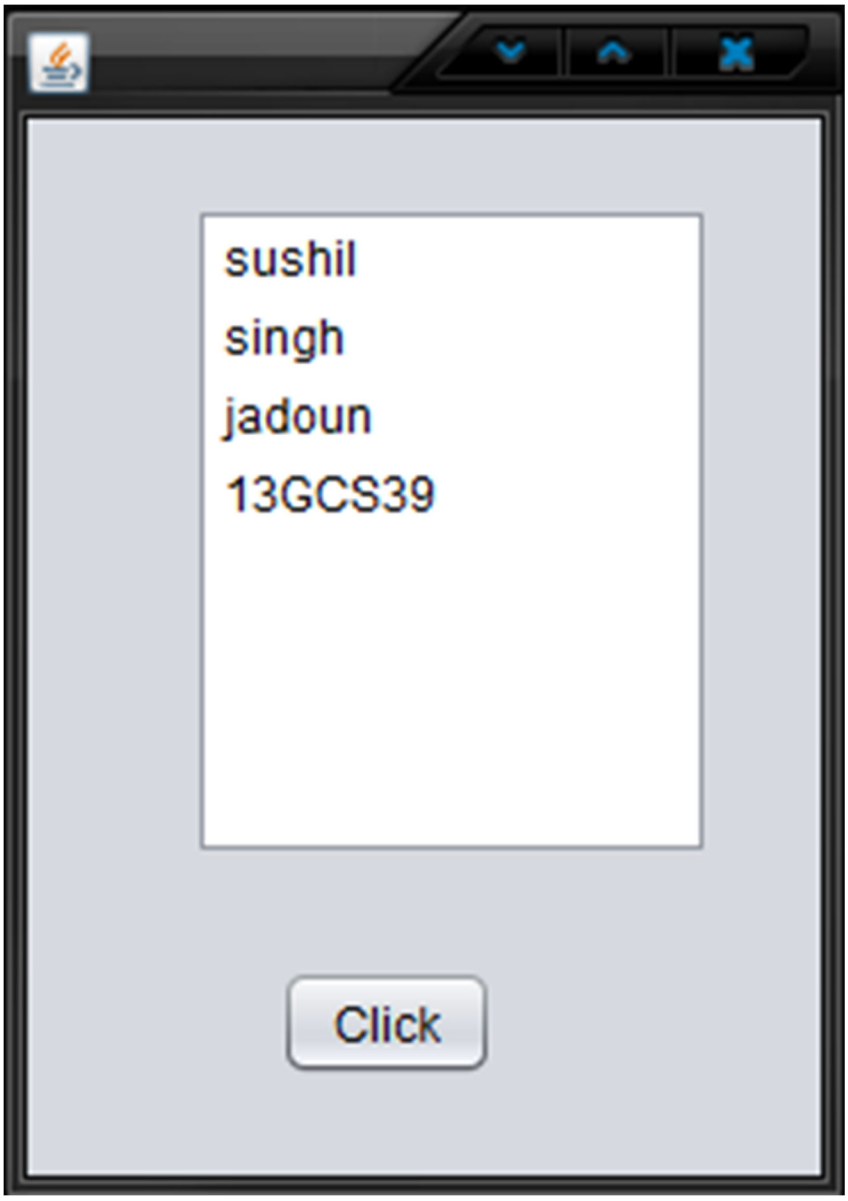
Experiment – 22

Aim: - WAP to implement the list.

```
package advancejava;
import java.sql.*;
import java.util.logging.*;
import javax.swing.DefaultListModel;
public class exp22 extends javax.swing.JFrame {
   public exp22() {
      initComponents();
   }
         //Autogenerated code
            private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      String query="Select name from exp21;";
      DefaultListModel model=new DefaultListModel();
      try{
         Class.forName("java.sql.DriverManager");
         Connection con=(Connection)
               DriverManager.getConnection("jdbc:mysql://localhost:3306/advancejava","root","1234");
         Statement stmt=con.createStatement();
         ResultSet rs=stmt.executeQuery(query);
         while(rs.next()){
            model.addElement(rs.getString(1));
         }jList1.setModel(model);
         } catch (SQLException | ClassNotFoundException ex) {
         Logger.getLogger(exp21.class.getName()).log(Level.SEVERE, null, ex);
      }
   }
   public static void main(String args[]) {
                  //Autogenerated code
                        java.awt.EventQueue.invokeLater(new Runnable() {
         public void run() {
            new exp22().setVisible(true);
```

```
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JList jList1;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration
}
```

Output

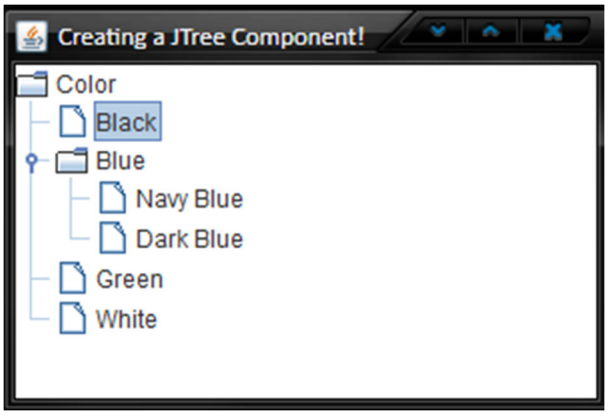


Experiment – 23

Aim: - WAP to implement the JTrees.

```
package advancejava;
import javax.swing.*;
import javax.swing.tree.*;
public class exp23{
  public static void main(String[] args) {
    JFrame frame = new JFrame("Creating a JTree Component!");
    DefaultMutableTreeNode parent = new DefaultMutableTreeNode("Color", true);
    DefaultMutableTreeNode black = new DefaultMutableTreeNode("Black");
    DefaultMutableTreeNode blue = new DefaultMutableTreeNode("Blue");
    DefaultMutableTreeNode nBlue = new DefaultMutableTreeNode("Navy Blue");
    DefaultMutableTreeNode dBlue = new DefaultMutableTreeNode("Dark Blue");
    DefaultMutableTreeNode green = new DefaultMutableTreeNode("Green");
    DefaultMutableTreeNode white = new DefaultMutableTreeNode("White");
    parent.add(black);          parent.add(blue);          blue.add(nBlue);
    blue.add(dBlue);            parent.add(green );        parent.add(white);
    JTree tree = new JTree(parent);
    frame.add(tree);
    frame.setSize(200,200);
    frame.setVisible(true);
  }
}
```

Output

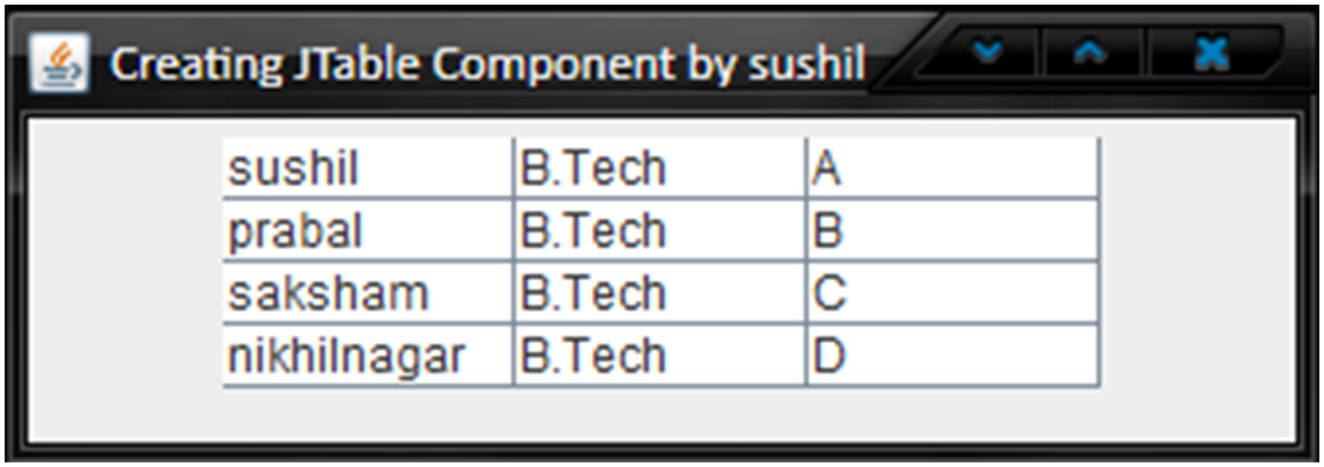


Experiment – 24

Aim: - WAP to implement the JTable.

```
package advancejava;
import javax.swing.*;
import java.awt.*;
import javax.swing.table.*;
public class exp24{
  public static void main(String[] args){
      exp24 exp24 = new exp24();
  }public exp24(){
    JFrame frame = new JFrame("Creating JTable Component by sushil");
    JPanel panel = new JPanel();
    String data[][] = {{"sushil","B.Tech","A"},{"prabal","B.Tech","B"},
     {"saksham","B.Tech","C"},{"nikhilnagar","B.Tech","D"}};
      String col[] = {"Name","Course","Grade"};
    JTable table = new JTable(data,col);
    panel.add(table,BorderLayout.CENTER);
    frame.add(panel);
    frame.setSize(300,200);
    frame.setVisible(true);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}}
```

Output



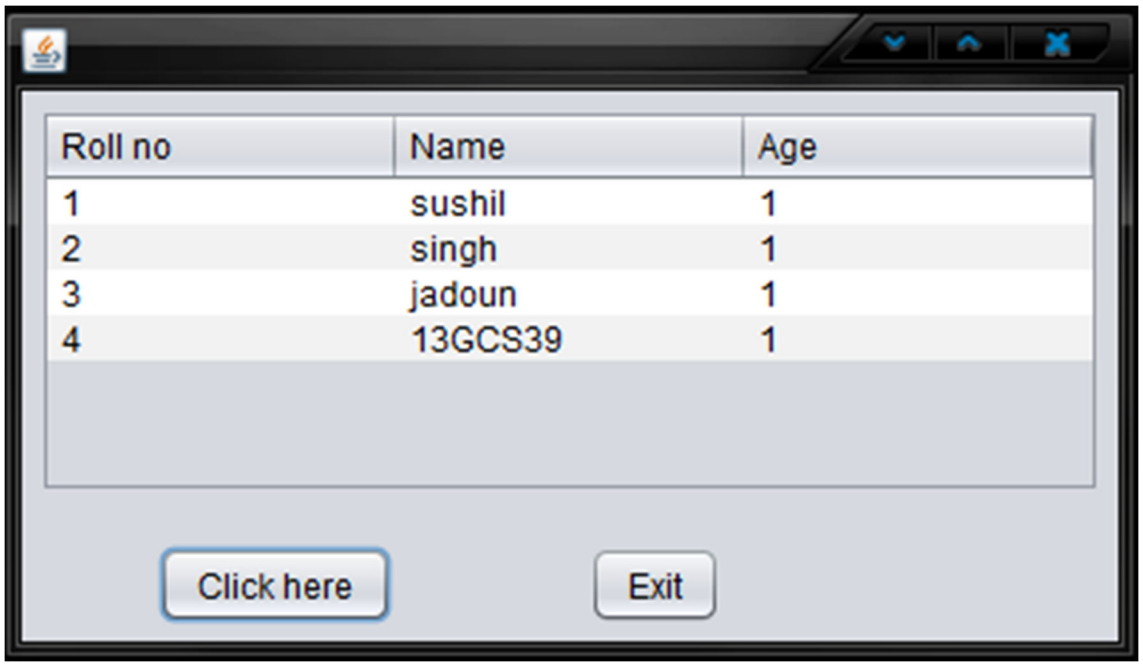
Experiment – 25

Aim: - WAP to create the table using JDBC.

```
package advancejava;
import java.sql.*;
import java.util.logging.*;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class exp25 extends javax.swing.JFrame {
   public exp25() {
      initComponents();
   }
         //autogenerated code
            private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
      System.exit(0);
   }
   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
   String query="Select*from exp21;";
   DefaultTableModel model= (DefaultTableModel)jTable1.getModel();
      int colno=3;
   try{
      Class.forName("java.sql.DriverManager");
      Connection con=(Connection)
            DriverManager.getConnection("jdbc:mysql://localhost:3306/advancejava","root","1234");
      Statement stmt=con.createStatement();
      ResultSet rs =  stmt.executeQuery(query);
      while(rs.next()){
         Object[] cell= new Object[colno];
         for(int i=0;i<colno;i++){
            cell[i]=rs.getObject(i+1);
         }model.addRow(cell);
      }
      jTable1.setModel(model);
```

```
        } catch (SQLException | ClassNotFoundException ex) {
            Logger.getLogger(exp21.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    public static void main(String args[]) {
                        //autogenerated code
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new exp25().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    // End of variables declaration
}
```

Output



Experiment – 26

Aim: - WAP to implement the Remote Method Invocation.

File : AddI.java

```
import java.rmi.Remote;
public interface AddI extends Remote{
        public int add(int x,int y) throws Exception;
}
```

File : AddC.java

```
import java.rmi.server.*;
public class AddC extends UnicastRemoteObject implements AddI{
        public AddC() throws Exception{
                super();
        }public int add(int x,int y) {
                return x+y;
} }
```

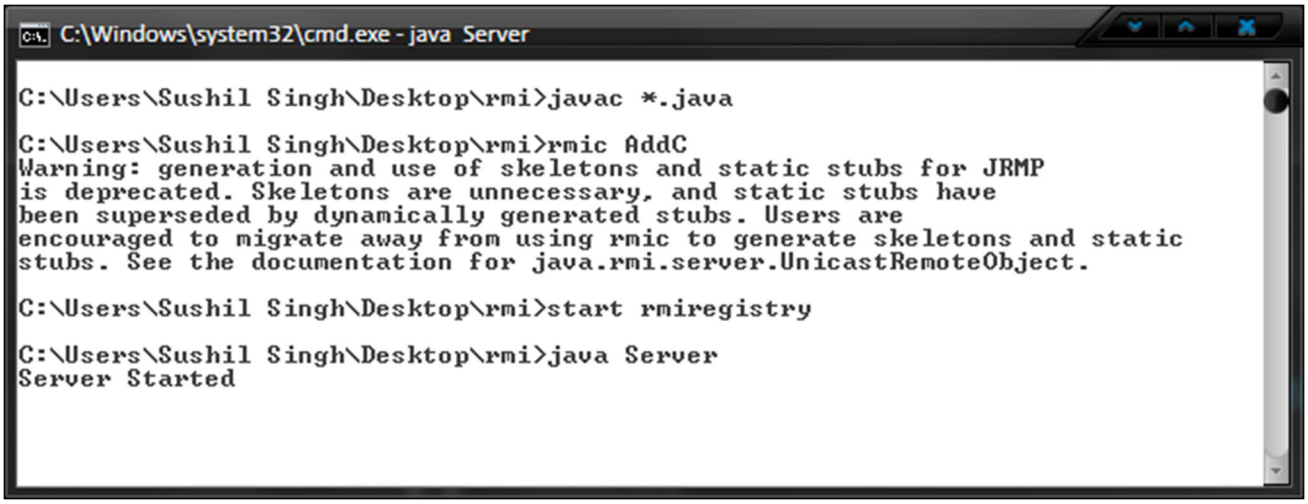
File : Client.java

```
import java.rmi.*;
public class Client{
        public static void main(String []args) throws Exception{
                AddI obj =(AddI) Naming.lookup("Add");
                int n = obj.add(5,4);
                System.out.println("Additon is : " + n);
} }
```

File : Server.java

```
import java.rmi.*;
public class Server{
        public static void main(String []args) throws Exception{
                AddC obj = new AddC();
                Naming.rebind("Add",obj);
                System.out.println("Server Started");
} }
```

Output

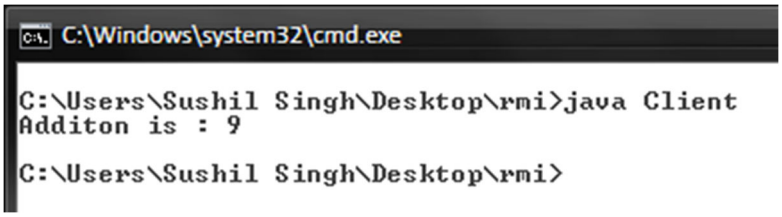


1.Compile all .java files.

2.Generate skeletons and stubs.

3.Start the registry.

4.Start server.



Client file execution.