

JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
Experiment: 01	

PROGRAM :

```
import java.util.Scanner;
public class AmstrongTest
{
    public static void main(String[] args)
    {
        int num;
        Scanner sc=new Scanner(System.in);
        num=sc.nextInt();
        int n = num;
        //use to check at last time
        int check=0,remainder;
        while(num > 0)
        {
            remainder = num % 10;
            check = check +(remainder*remainder*remainder) ;
            num = num / 10;
        }
        if(check == n)
            System.out.println(n+" is an Armstrong Number");
        else
            System.out.println(n+" is not a Armstrong Number");
    }
}
```

JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

```
C:\ Command Prompt
C:\Users\pc\Lab>javac AmstrongTest.java
C:\Users\pc\Lab>java AmstrongTest
153
153 is an Armstrong Number
C:\Users\pc\Lab>java AmstrongTest
123
123 is not a Armstrong Number
C:\Users\pc\Lab>
```

JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
Experiment: 02	

PROGRAM :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
//<applet code="Calculator" height=500 width=500></applet>
public class Calculator extends JApplet
{
    public void init()
    {
        CalculatorPanel calc=new CalculatorPanel();
        getContentPane().add(calc);
    }
}
class CalculatorPanel extends JPanel implements ActionListener
{
    JButton n1,n2,n3,n4,n5,n6,n7,n8,n9,n0,plus,minus,mul,div,dot,equal;
    static JTextField result=new JTextField("0",45);
    static String lastCommand=null;
    JOptionPane p=new JOptionPane();
    double preRes=0,
```

JNTUHCEJ



Roll no:20JJ1A1248

```
secVal=0,res;  
  
private static void assign(String no)  
{  
if((result.getText()).equals("0"))  
result.setText(no);  
else if(lastCommand=="=")  
{  
result.setText(no);  
lastCommand=null;  
}  
else  
result.setText(result.getText()+no);  
}  
  
public CalculatorPanel()  
{  
setLayout(new BorderLayout());  
result.setEditable(false);  
result.setSize(300,200);  
add(result,BorderLayout.NORTH);  
JPanel panel=new JPanel();  
panel.setLayout(new GridLayout(4,4));  
n7=new JButton("7");  
panel.add(n7);  
n7.addActionListener(this);  
n8=new JButton("8");  
panel.add(n8);
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
n8.addActionListener(this);  
n9=new JButton("9");  
panel.add(n9);  
n9.addActionListener(this);  
div=new JButton("/");  
panel.add(div);  
div.addActionListener(this);  
n4=new JButton("4");  
panel.add(n4);  
n4.addActionListener(this);  
n5=new JButton("5");  
panel.add(n5);  
n5.addActionListener(this);  
n6=new JButton("6");  
panel.add(n6);  
n6.addActionListener(this);  
mul=new JButton("*");  
panel.add(mul);  
mul.addActionListener(this);  
n1=new JButton("1");  
panel.add(n1);  
n1.addActionListener(this);  
n2=new JButton("2");  
panel.add(n2);  
n2.addActionListener(this);  
n3=new JButton("3");
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
panel.add(n3);n3.addActionListener(this);
minus=new JButton("-");panel.add(minus);
minus.addActionListener(this);
dot=new JButton(".");
panel.add(dot);
dot.addActionListener(this);
n0=new JButton("0");
panel.add(n0);
n0.addActionListener(this);
equal=new JButton "=";
panel.add(equal);
equal.addActionListener(this);
plus=new JButton "+";
panel.add(plus);
plus.addActionListener(this);
add(panel,BorderLayout.CENTER);
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==n1) assign("1");
else if(ae.getSource()==n2) assign("2");
else if(ae.getSource()==n3) assign("3");
else if(ae.getSource()==n4) assign("4");
else if(ae.getSource()==n5) assign("5");
else if(ae.getSource()==n6) assign("6");
else if(ae.getSource()==n7) assign("7");
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
else if(ae.getSource()==n8) assign("8");
else if(ae.getSource()==n9) assign("9");
else if(ae.getSource()==n0) assign("0");
else if(ae.getSource()==dot)
{
if(((result.getText()).indexOf("."))==-1)
result.setText(result.getText()+".");
}
else if(ae.getSource()==minus)
{
preRes=Double.parseDouble(result.getText());
lastCommand="-";
result.setText("0");
}
else if(ae.getSource()==div)
{
preRes=Double.parseDouble(result.getText());
lastCommand="/";
result.setText("0");
}
else if(ae.getSource()==equal)
{
secVal=Double.parseDouble(result.getText());
if(lastCommand.equals("/"))
res=preRes/secVal;else if(lastCommand.equals("*"))
res=preRes*secVal;else if(lastCommand.equals("-"))
res=preRes-secVal;else if(lastCommand.equals("+"))
res=preRes+secVal;
}
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
res=preRes*secVal;  
else if(lastCommand.equals("-"))  
res=preRes-secVal;  
else if(lastCommand.equals("+"))  
res=preRes+secVal;  
  
result.setText(" "+res);  
lastCommand="-";  
}  
else if(ae.getSource()==mul)  
{  
preRes=Double.parseDouble(result.getText());  
lastCommand="*";  
result.setText("0");  
}  
else if(ae.getSource()==plus)  
{  
preRes=Double.parseDouble(result.getText());  
lastCommand="+";  
result.setText("0");  
}  
}  
}
```



OUTPUT :

```
ca] Command Prompt
Microsoft Windows [Version 10.0.19043.1151]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pc>cd Lab
C:\Users\pc\Lab>javac Calculator.java
C:\Users\pc\Lab>appletviewer Calculator.html
```

A screenshot of the Java Applet Viewer window. The title bar says "Applet Viewer: Si...". The main area displays a 4x4 grid of buttons labeled with digits 1-9, 0, and operators +, -, *, /, and a decimal point (.). Below the grid, a status bar shows "Applet started.".

JNTUHCEJ



Roll no: 20JJ1A1248

Date:	A) Develop an applet in Java that displays a simple message.
Experiment: 03	B) Develop an applet in Java that receives an integer in one text field, and computes its factorial. Value and returns it in another text field, when the button named "Compute" is clicked.

PROGRAM A :

```
import java.awt.*;
import java.applet.*;
/*<applet code="FirstApplet.class" width=300 height=300></applet>/
public class FirstApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("this is first applet program",100,100);
    }
}
```

PROGRAM B :

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class FactorialApplet extends Applet implements ActionListener
{
    Label L1,L2;
    TextField T1,T2;
```

JNTUHCEJ



Roll no: 20JJ1A1248

Button B1;

public void init()

{

L1=new Label("enter the value");

add(L1);

T1=new TextField(10);

add(T1);

L2=new Label("Factorial of num");

add(L2);

T2=new TextField(10);

add(T2);

B1=new Button("Compute");

add(B1);

B1.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==B1)

{

int value=Integer.parseInt(T1.getText());

int fact=factorial(value);

T2.setText(String.valueOf(fact));

}}

int factorial(int n)

{

if(n==0)

JNTUHCEJ



Roll no: 20JJ1A1248

```
return 1;  
else  
return n*factorial(n-1);  
}  
}  
  
/*<applet code="FactorialApplet" width=300 height=300> </applet>*/
```



JNTUHCEJ

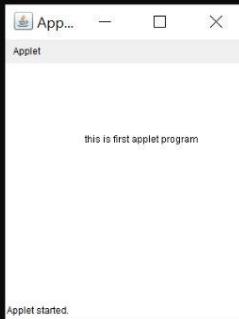


Roll no: 20JJ1A1248

OUTPUT :

```
C:\ Command Prompt - appletviewer FirstApplet.html
Microsoft Windows [Version 10.0.19043.1151]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pc>cd Lab
C:\Users\pc\Lab>javac FirstApplet.java
C:\Users\pc\Lab>appletviewer FirstApplet.html
```



```
C:\ Command Prompt - appletviewer FactorialApplet.html
Microsoft Windows [Version 10.0.19043.1151]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pc>cd Lab
C:\Users\pc\Lab>javac FactorialApplet.java
C:\Users\pc\Lab>appletviewer FactorialApplet.html
```



JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box
Experiment: 04	

PROGRAM :

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
import javax.swing.*;
public class DivisionApplet extends Applet implements ActionListener
{
    Label L1,L2,L3;
    TextField T1,T2,Result;
    Button B1;
    public void init()
    {
        L1=new Label("enter first num");
        add(L1);
        T1=new TextField(10);
        add(T1);
        L2=new Label("enter second num");
        add(L2);
        T2=new TextField(10);
        add(T2);
        L3=new Label("result");
        add(L3);
        Result=new TextField(10);
        add(Result);
        B1=new Button("Divide");
        add(B1);
        B1.addActionListener(this);
    }
}
```

JAVA PROGRAMMING

JNTUHCEJ



Roll no: 20JJ1A1248

```
}

public void actionPerformed(ActionEvent e)
{
if(e.getSource()==B1)
{
try
{
int value1=Integer.parseInt(T1.getText());
int value2=Integer.parseInt(T2.getText());
int result=value1/value2;
Result.setText(String.valueOf(result));
}
catch(NumberFormatException nfe)
{
JOptionPane.showMessageDialog(this,"not a number");
}
catch(ArithmeticException ae)
{
JOptionPane.showMessageDialog(this,"divided by zero");
}
}
}
}
}

/*
<applet code="DivisionApplet" width=500 height=500> </applet>*/
```

JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

```
C:\Select Command Prompt - appletviewer DivisionApplet.html
C:\Users\pc>cd Lab
C:\Users\pc\Lab>javac DivisionApplet.java
C:\Users\pc\Lab>appletviewer DivisionApplet.html

Applet Viewer: DivisionApplet... — □ ×
Applet
enter first num: 101 enter second num: 10 result:
10 [Divide]
```





JNTUHCEJ

Roll no: 20JJ1A1248

Date:	Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number
Experiment: 05	

PROGRAM :

```

import java.util.*;
class Even implements Runnable
{
public int x;
public Even(int x)
{
this.x = x;
}
public void run()
{
System.out.println("New Thread "+ x +" is EVEN and Square of " + x + " is: " + x * x);
}
}
class Odd implements Runnable
{
public int x;
public Odd(int x)
{
this.x = x;
}
public void run()
{
System.out.println("New Thread "+ x +" is ODD and Cube of " + x + " is: " + x * x * x);
}
}
class RandomNumberGenerating extends Thread
{
public void run()
{
int num = 0;

```

JNTUHCEJ



Roll no: 20JJ1A1248

```
Random r = new Random();
try
{
for (int i = 0; i < 5; i++)
{
num = r.nextInt(20);
System.out.println("Generated Number is " + num);
if (num % 2 == 0)
Thread t1 = new Thread(new Even(num));
t1.start();
}
else
{
Thread t2 = new Thread(new Odd(num));
t2.start();
}
Thread.sleep(1000);
System.out.println(".....");
}
}
catch (Exception ex)
{
System.out.println(ex.getMessage());
}
}
}

public class MultiOddEvenThread
{
public static void main(String[] args)
{
RandomNumberGenerating a = new RandomNumberGenerating();
a.start();
}
}
```

JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

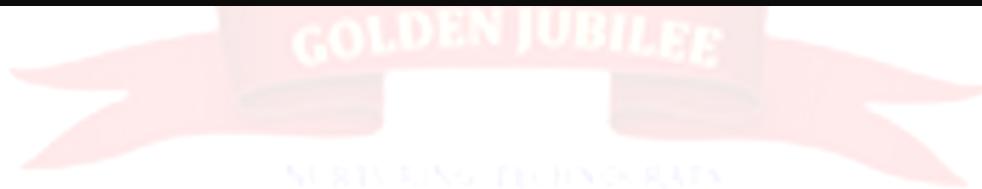
```
[cmd] Command Prompt
Microsoft Windows [Version 10.0.19043.1151]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pc>cd Lab

C:\Users\pc\Lab>javac MultiOddEvenThread.java

C:\Users\pc\Lab>java MultiOddEvenThread
Generated Number is 6
New Thread 6 is EVEN and Square of 6 is: 36
-----
Generated Number is 9
New Thread 9 is ODD and Cube of 9 is: 729
-----
Generated Number is 9
New Thread 9 is ODD and Cube of 9 is: 729
-----
Generated Number is 14
New Thread 14 is EVEN and Square of 14 is: 196
-----
Generated Number is 18
New Thread 18 is EVEN and Square of 18 is: 324
-----

C:\Users\pc\Lab>
```



JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Write a Java program for the following:
Experiment: 06	Create a doubly linked list of elements. Delete a given element from the above list. Display the contents of the list after deletion.

PROGRAM :

```

import java.util.*;
public class DoublyLinkListDemo {
    public static void main(String[] args) {
        int i,choice,ele,pos;
        LinkedList<Integer> dobList = new LinkedList<Integer>();
        System.out.println("1.Insert at begining");
        System.out.println("2.Insert at end");
        System.out.println("3.Insert at position");
        System.out.println("4.Enter element to delete");
        System.out.println("5.Exit");
        Scanner sc=new Scanner(System.in);
        do
        {
            System.out.println("enter your choice");
            choice=sc.nextInt();
            switch(choice)
            {
                case 1:System.out.println("enter element");
                ele=sc.nextInt();
                dobList.addFirst(ele);
                break;
                case 2:System.out.println("enter element");
                ele=sc.nextInt();
                dobList.addLast(ele);
                break;
                case 3:System.out.println("enter position");
                pos=sc.nextInt();
                if(pos<=dobList.size())
                {
                    System.out.println("enter element");
                    ele=sc.nextInt();

```

JNTUHCEJ



Roll no: 20JJ1A1248

```
dobList.add(pos,ele);
}
else
{
System.out.println("enter the size between 0to"+dobList.size());
}
break;
case 4:System.out.println("enter element to be deleted");
Integer rm;
rm=sc.nextInt();
if (dobList.contains(rm))
{
dobList.remove(rm);
Iterator itr=dobList.iterator();
System.out.println("Elements after deleting :"+rm);
while(itr.hasNext())
{
System.out.print(itr.next()+"<->");
}
System.out.println("NULL");
}
else
{
System.out.println("element not found");
}
break;
case 5:System.out.println("program terminated");
break;
default:System.out.println("enter correct choice");
}
}
}
while(choice!=5);
}
}
```

JNTUHCEJ

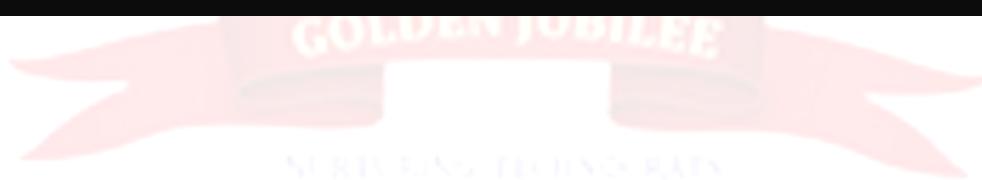


Roll no: 20JJ1A1248

OUTPUT :

```
C:\ Command Prompt
C:\Users\pc>cd Lab
C:\Users\pc\Lab>javac DoublyLinkedListDemo.java
C:\Users\pc\Lab>java DoublyLinkedListDemo
1.Insert at begining
2.Insert at end
3.Insert at position
4.Enter element to delete
5.Exit
enter your choice
1
enter element
10
enter your choice
2
enter element
20
enter your choice
4
enter element to be deleted
10
Elements after deleting :10
20<->NULL
enter your choice
5
program terminated

C:\Users\pc\Lab>
```





JNTUHCEJ

Roll no: 20JJ1A1248

Date:	Write a Java program that simulates a traffic light.
Experiment: 07	The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially, there is no message shown

PROGRAM :

```

import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
class TrafficSimulator extends JFrame implements ItemListener
{
JLabel l1, l2,l3,l4,l5;
 JPanel nPanel, cPanel;
 CheckboxGroup cbg;
 public TrafficSimulator()
 {
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(400,400);
setLayout(new GridLayout(2, 1));
nPanel = new JPanel(new FlowLayout());
cPanel = new JPanel(new FlowLayout());
l1 = new JLabel();
Font f = new Font("Times New Roman", Font.ITALIC, 72);
l1.setFont(f);
nPanel.add(l1);
add(nPanel);
l2 = new JLabel("Select Lights");
cPanel.add(l2);
cbg = new CheckboxGroup();
Checkbox r1 = new Checkbox("Red Light", cbg, true);
r1.setBackground(Color.red);
cPanel.add(r1);
r1.addItemListener(this);
Checkbox r2 = new Checkbox("Yellow Light", cbg, true);

```

JNTUHCEJ



Roll no: 20JJ1A1248

```
r2.setBackground(Color.YELLOW);
cPanel.add(r2);
r2.addItemListener(this);
Checkbox r3 = new Checkbox("Green Light", cbg, true);
r3.setBackground(Color.GREEN);
cPanel.add(r3);
r3.addItemListener(this);
add(cPanel);
setVisible(true);
}
public void itemStateChanged(ItemEvent i)
{
Checkbox chk = cbg.getSelectedCheckbox();
String str=chk.getLabel();
char choice=str.charAt(0);
switch (choice)
{
case 'R':
l1.setText("STOP");
l1.setForeground(Color.red);
break;
case 'Y':
l1.setText("Ready");
l1.setForeground(Color.YELLOW);
break;
case 'G':
l1.setText("GO");
l1.setForeground(Color.GREEN);
break;
}
}
public static void main(String[] args)
{
TrafficSimulator a = new TrafficSimulator();
}
```

JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

```
C:\ Command Prompt - java TrafficSimulator
C:\Users\pc\Lab>javac TrafficSimulator.java
C:\Users\pc\Lab>java TrafficSimulator
```

The screenshot shows a Java application window titled "STOP". Inside the window, there is a label "Select Lights" followed by three radio buttons: "Red Light" (red), "Yellow Light" (yellow), and "Green Light" (green). The "Red Light" button is selected. The window has standard operating system window controls (minimize, maximize, close) at the top.



JNTUHCEJ

Roll no: 20JJ1A1248

Date:	Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape
Experiment: 08	

PROGRAM :

```

import java.util.*;
abstract class Shape {
public int x,y;
public abstract void printArea();
}
class Rectangle1 extends Shape {
public void printArea() {
float area;
area= x * y;
System.out.println("Area of Rectangle is " +area);
}}
class Triangle extends Shape {
public void printArea()
{
float area;
area= (x * y) / 2.0f;
System.out.println("Area of Triangle is " + area);
}}
class Circle extends Shape {
public void printArea()
{
float area;
area=(22 * x * x) / 7.0f;
System.out.println("Area of Circle is " + area);
}}
public class AreaOfShapes {

```

JNTUHCEJ



Roll no:20JJ1A1248

```
public static void main(String[] args)
{
    int choice;
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter your choice \n 1.Area of Rectangle \n 2.Area of Traingle \n
3.Area of
Circle ");
    choice=sc.nextInt();
    switch(choice)
    {
        case 1:System.out.println("Enter length and breadth for area of rectangle : ");
        Rectangle1 r = new Rectangle1();
        r.x=sc.nextInt();
        r.y=sc.nextInt();
        r.printArea();
        break;
        case 2:System.out.println("Enter breadth and height for area of traingle : ");
        Triangle t = new Triangle();
        t.x=sc.nextInt();
        t.y=sc.nextInt();
        t.printArea();
        break;
        case 3:System.out.println("Enter radius for area of circle : ");
        Circle c = new Circle();
        c.x = sc.nextInt();
        c.printArea();
        break;
        default:System.out.println("Enter correct choice");
    }
}
```

JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

```
C:\ Command Prompt  
C:\Users\pc\Lab>javac AreaOfShapes.java  
C:\Users\pc\Lab>java AreaOfShapes  
Enter your choice  
1.Area of Rectangle  
2.Area of Traingle  
3.Area of Circle  
3  
Enter radius for area of circle :  
5  
Area of Circle is 78.57143  
C:\Users\pc\Lab>
```

JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.
Experiment: 09	

PROGRAM :

```
import java.io.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
class Text_To_Table extends JFrame
{
    public void convertTexttotable()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(700,800);
        GridLayout g = new GridLayout(0, 4);
        setLayout(g);
        try
        {
            FileInputStream fis = new FileInputStream("./Table.txt");
            Scanner sc = new Scanner(fis);
            String[] arrayList;
            String str;
            while (sc.hasNextLine())
            {
                str = sc.nextLine();
                arrayList = str.split(",");
                for (String i : arrayList)
                {
                    add(new Label(i));
                }
            }
        } catch (Exception ex) {
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
ex.printStackTrace();
}
setVisible(true);
}
}

public class TableText
{
public static void main(String[] args)
{
Text_To_Table tt = new Text_To_Table();
tt.convertTexttotable();
}
}

Table.txt
name,htnum,marks,result
SWATHI,157Z1A0501,544,PASS
RAMESH,157Z1A0503,344,PASS
GAURAV,167Z1A0523,344,PASS
ANOK,167Z1A0501,344,PASS
KISHORE,167Z1A05,344,PASS
```

JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

```
C:\ Command Prompt - java TableText
C:\Users\pc\Lab>javac TableText.java
C:\Users\pc\Lab>java TableText


| name    | htnum      | marks | result |
|---------|------------|-------|--------|
| SWATHI  | 15721A0501 | 544   | PASS   |
| RAMESH  | 15721A0503 | 344   | PASS   |
| GAURAV  | 15721A0523 | 344   | PASS   |
| ANOK    | 15721A0501 | 344   | PASS   |
| KISHORE | 15721A0505 | 344   | PASS   |


```

JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
Experiment: 10	

PROGRAM :

```
import javax.swing.*;
import java.awt.*;
import javax.swing.event.*;
import java.awt.event.*;
class MouseEventPerformer extends JFrame implements MouseListener
{
JLabel l1;
public MouseEventPerformer()
{
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(500,500);
setLayout(new FlowLayout(FlowLayout.CENTER));
l1 = new JLabel();
Font f = new Font("Verdana", Font.BOLD, 20);
l1.setFont(f);
l1.setForeground(Color.BLUE);
add(l1);
addMouseListener(this);
setVisible(true);
}
public void mouseExited(MouseEvent m)
{
l1.setText("Mouse Exited");
}
public void mouseEntered(MouseEvent m)
{
l1.setText("Mouse Entered");
}
public void mouseReleased(MouseEvent m)
{
l1.setText("Mouse Released");
}
public void mousePressed(MouseEvent m)
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
{  
l1.setText("Mouse Pressed");  
}  
public void mouseClicked(MouseEvent m)  
{  
l1.setText("Mouse Clicked");  
}  
public static void main(String[] args) {  
MouseEventPerformer mep = new MouseEventPerformer();  
}  
}
```



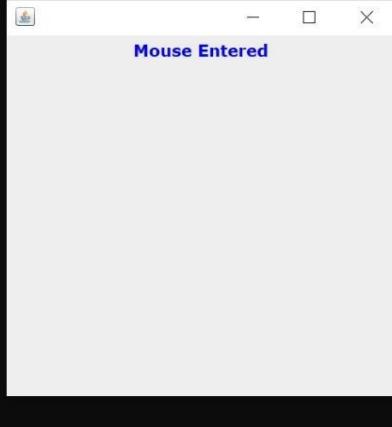
JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

```
C:\ Command Prompt - java MouseEventPerformer
C:\Users\pc\Lab>javac MouseEventPerformer.java
C:\Users\pc\Lab>java MouseEventPerformer


```

JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
Experiment: 11	

PROGRAM :

```

import java.io.*;
import java.util.*;
public class Phonebook
{
    public static void main(String args[])
    {
        try
        {
            FileInputStream fis=new FileInputStream("C:/Users/pc/lab");
            Scanner sc=new Scanner(fis).useDelimiter("\t");
            Hashtable<String,String> ht=new Hashtable<String,String> ();
            String[] strarray;
            String a,str;
            while(sc.hasNext())
            {
                a=sc.nextLine();
                strarray=a.split("\t");
                ht.put(strarray[0],strarray[1]);
                System.out.println("hash table values are \t"+strarray[0]+":"+strarray[1]);
            }
            Scanner s=new Scanner(System.in);
            System.out.println("Enter the name as given in the phone book");
            str=s.next();
            if(ht.containsKey(str))
            {
                System.out.println("phone no is"+ht.get(str));
            }
            else

```

JNTUHCEJ



Roll no: 20JJ1A1248

```
{  
System.out.println("Name is not matched");  
}  
}  
}  
catch(Exception e)  
{  
System.out.println(e);  
}  
}  
}  
}  
}  
Myfile.txt
```

Surya	567
Ravi	456



JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

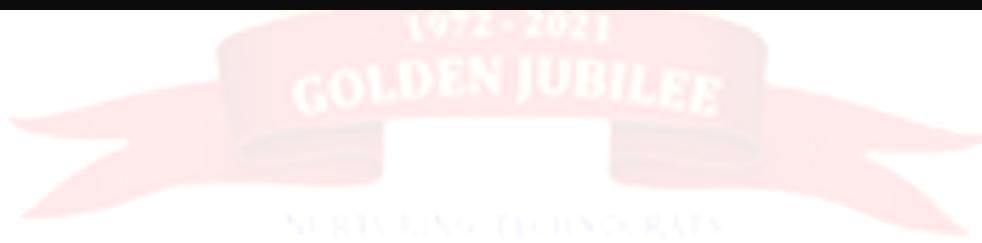
```
C:\ Command Prompt
Microsoft Windows [Version 10.0.19043.1151]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pc>cd Lab

C:\Users\pc\Lab>javac Phonebook.java

C:\Users\pc\Lab>java Phonebook
hash table values are   Surya:567
hash table values are   Ravi:456
Enter the name as given in the phone book
Ravi
phone no is456

C:\Users\pc\Lab>
```





JNTUHCEJ

Roll no: 20JJ1A1248

Date:

Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication

Experiment: 12

PROGRAM :

```
import java.io.*;
class Thread1
{
int n;
boolean valueset=false;
synchronized int get()
{
if (!valueset)
try
{
wait();
}
catch (Exception e)
{
System.out.println("Exception occur at :" +e);
}
System.out.println("get" +n);
try
{
Thread.sleep(1000);
}
catch (Exception e)
{
System.out.println("Exception occur at :" +e);
}
valueset=false;
notify();
return n;
}
synchronized int put(int n)
{
if (valueset)
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
try
{
wait();
}
catch (Exception e)
{
System.out.println("Exception occur at : "+e);
}
this.n=n;
valueset=true;
System.out.println("put"+n);
try
{
Thread.sleep(1000);
}
catch (Exception e)
{
System.out.println("Exception occur at : "+e);
}
notify();
return n;
}
}
class Producer implements Runnable
{
Thread1 t;
Producer(Thread1 t)
{
this.t=t;
new Thread(this,"Producer").start();
}
public void run()
{
int i=0;
while (true)
{
t.put(i++);
}
}
}
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
class Consumer implements Runnable
{
    Thread1 t;
    Consumer(Thread1 t)
    {
        this.t=t;
        new Thread(this,"Consumer").start();
    }
    public void run()
    {
        int i=0;
        while (true)
        {
            t.get();
        }
    }
}
class ProducerConsumer
{
    public static void main(String[] args) throws IOException
    {
        Thread1 t=new Thread1();
        new Producer(t);
        new Consumer(t);
        System.out.println("Press Control+c to exit");
    }
}
```

JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

```
C:\ Command Prompt  
C:\Users\pc\Lab>javac ProducerConsumer.java  
C:\Users\pc\Lab>java ProducerConsumer  
put0  
Press Control+c to exit  
get0  
put1  
get1  
put2  
get2  
put3  
get3  
put4  
get4  
put5  
get5  
C:\Users\pc\Lab>
```

JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Write a Java program to list all the files in a directory including the files present in all its subdirectories
Experiment: 13	

PROGRAM :

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

// creates a file object
File file = new File("C:/Users/pc/lab");

// returns an array of all files
String[] fileList = file.list();

for(String str : fileList) {
System.out.println(str);
}
}
}
```



OUTPUT :

```
C:\ Command Prompt
C:\Users\pc\Lab>javac Main.java
C:\Users\pc\Lab>java Main
AmstrongTest.class
AmstrongTest.java
AreaOfShapes.class
AreaOfShapes.java
Calculator.class
Calculator.html
Calculator.java
CalculatorPanel.class
Circle.class
Consumer.class
DivisionApplet.class
DivisionApplet.html
DivisionApplet.java
DoublyLinkedListDemo.class
DoublyLinkedListDemo.java
Even.class
FactorialApplet.class
FactorialApplet.html
FactorialApplet.java
FirstApplet.class
FirstApplet.html
FirstApplet.java
Main.class
Main.java
MouseEventPerformer.class
MouseEventPerformer.java
MultiOddEvenThread.class
MultiOddEvenThread.java
myfile.txt
Odd.class
Phonebook.class
Phonebook.java
Producer.class
ProducerConsumer.class
ProducerConsumer.java
RandomNumberGenerating.class
Rectanglei.class
```



JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order
Experiment: 14	

PROGRAM :

```

public class QuickSortOnStrings {

    String names[];
    int length;

    public static void main(String[] args) {
        QuickSortOnStrings obj = new QuickSortOnStrings();
        String stringsList[] = {"raja", "gouthu", "rani", "gouthami",
honey", "heyaansh", "hello"};
        obj.sort(stringsList);

        for (String i : stringsList) {
            System.out.print(i);
            System.out.print(" ");
        }
    }

    void sort(String array[]) {
        if (array == null || array.length == 0) {
            return;
        }
        this.names = array;
        this.length = array.length;
        quickSort(0, length - 1);
    }

    void quickSort(int lowerIndex, int higherIndex) {
        int i = lowerIndex;
        int j = higherIndex;
        String pivot = this.names[lowerIndex + (higherIndex -
lowerIndex) / 2];

        while (i <= j) {
            while (this.names[i].compareTo(pivot) > 0)
                i++;
            while (this.names[j].compareTo(pivot) < 0)
                j--;
            if (i <= j) {
                String temp = this.names[i];
                this.names[i] = this.names[j];
                this.names[j] = temp;
                i++;
                j--;
            }
        }
    }
}
  
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
i++;
}

while (this.names[j].compareToIgnoreCase(pivot) > 0) {
    j--;
}

if (i <= j) {
    exchangeNames(i, j);
    i++;
    j--;
}
}

if (lowerIndex < j) {
    quickSort(lowerIndex, j);
}
if (i < higherIndex) {
    quickSort(i, higherIndex);
}

void exchangeNames(int i, int j) {
    String temp = this.names[i];
    this.names[i] = this.names[j];
    this.names[j] = temp;
}
```



OUTPUT :Gouthami
Gouthu
Hello
Heyaansh
Honey
Raja
Rani

JNTUHCEJ



Roll no: 20JJ1A1248

Date:	Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.
Experiment: 15	

PROGRAM :

```
public class BubbleSortDescendingOrder {  
  
    public static void main(String[] args) {  
  
        //create an int array we want to sort using bubble sort algorithm  
        int intArray[] = new int[]{5,90,35,45,150,3};  
  
        //print array before sorting using bubble sort algorithm  
        System.out.println("Array Before Bubble Sort");  
        for(int i=0; i < intArray.length; i++){  
            System.out.print(intArray[i] + " ");  
        }  
  
        //sort an array in descending order using bubble sort algorithm  
        bubbleSort(intArray);  
        System.out.println("");  
  
        //print array after sorting using bubble sort algorithm  
        System.out.println("Array After Bubble Sort");  
        for(int i=0; i < intArray.length; i++){  
            System.out.print(intArray[i] + " ");  
        }  
    }  
  
    private static void bubbleSort(int[] intArray) {  
        int n = intArray.length;  
        int temp = 0;  
  
        for(int i=0; i < n; i++){  
            for(int j=1; j < (n-i); j++){  
                if(intArray[j] > intArray[j+1]) {  
                    temp = intArray[j];  
                    intArray[j] = intArray[j+1];  
                    intArray[j+1] = temp;  
                    count++;  
                }  
            }  
        }  
    }  
}
```

JNTUHCEJ



Roll no: 20JJ1A1248

```
if(intArray[j-1] < intArray[j]){
//swap the elements!
temp = intArray[j-1];
intArray[j-1] = intArray[j];
intArray[j] = temp;
}
}
}
}
}
}
```



JNTUHCEJ



Roll no: 20JJ1A1248

OUTPUT :

```
C:\ Command Prompt  
C:\Users\pc\Lab>javac BubbleSortDescendingOrder.java  
C:\Users\pc\Lab>java BubbleSortDescendingOrder  
Array Before Bubble Sort  
5 90 35 45 150 3  
Array After Bubble Sort  
150 90 45 35 5 3  
C:\Users\pc\Lab>
```

