**Experiment - 1**

The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, 1. Every subsequent value is the sum of the 2 values preceding it. Write a Java Program that uses both recursive and non-recursive functions to print the nth value of the Fibonacci sequence.

**Source Code:**

import java.util.Scanner; class Prog1

{

public static void main(String args[ ])

{

Scanner sc=new Scanner(System.in); Fibonacci obj=new Fibonacci ();

int a,b,c;

System.out.println("Which term you want to print ");

a=sc.nextInt();

b=obj.fibo\_recursion(a);

System.out.println("The "+a+"th Fibonacci term using recursion is: "+b);

c=obj.fibo\_non\_recursion(a);

System.out.println("The "+a+"th Fibonacci term using non recursion is: "+c);

}

}

class Fibonacci

{

int a=1,b=1,c;

public int fibo\_recursion (int n)

{

if(n == 1)

return a;

else if(n == 2)

return b;

else

return fibo\_recursion(n-1) + fibo\_recursion(n-2);

}

public int fibo\_non\_recursion (int n)

{

if(n==1)

return a;

else if(n==2)

return b;

else

{

for(int i=3 ; i<=n ; i++)

{

c=a+b;

a=b;

b=c;

}

return c;

}

}

}

**OUTPUT:**

**Experiment-2:**

**Write a Java Program that prompts the user for an integer and then prints out all the prime numbers up to that Integer.**

Source Code:

import java.util.\*;

class Experiment2

{

public static boolean primeOrNot(int n)

{

int f=0;

for(int i=2;i<=Math.sqrt(n);i++)

{

if(n%i==0)

return false;

}

return true;

}

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

int range;

range=sc.nextInt();

for(int i=2;i<=range;i++)

{

if(primeOrNot(i))

System.out.print(i+" ");

}

}

}

**OUTPUT:**

**Experiment-3**

**Write a Java Program that checks whether a given string is a palindrome or not.**

Ex. MALAYALAM is a palindrome.

**Source Code:**

import java.util.Scanner;

class Experiment3

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

String str = sc.nextLine();

boolean flag = true;

**//Converts the given string into lowercase**

str = str.toLowerCase();

for(int i = 0; i < str.length()/2; i++)

{

if(str.charAt(i) != str.charAt(str.length()-i-1))

{

flag = false;

break;

}

}

if(flag)

System.out.println("Given string "+str+" is palindrome");

else

System.out.println("Given string "+str+" is not a palindrome");

}

}

**OUTPUT:**

**Experiment-4:**

**Write a Java Program for sorting a given list of names in ascending order.**

**Source Code:**

import java.util.Scanner;

class Experiment4

{

Scanner sc = new Scanner(System.in);

public void sortStrings()

{

System.out.print("How many Names ");

int i, j, n;

n = sc.nextInt();

String[] str = new String[n];

System.out.println("Enter " + n + " Strings");

for(i = 0; i < n; i++)

{

str[i] = new String(sc.next());

}

for(i = 0; i < n; i++)

{

for(j = i+1; j < n; j++)

{

if(str[i].compareTo(str[j])>0)

{

String temp = str[i]; str[i] = str[j];

str[j] = temp;

}

}

}

System.out.println("Given Strings in Alphabetical Order");

for(i=0; i<n ; i++)

System.out.println(str[i]);

}

}

class Mainclass

{

public static void main(String args[])

{

Experiment4 obj = new Experiment4();

obj.sortStrings();

}

}

**OUTPUT:**

**Experiment-5:**

**Write a Java Program that illustrates how runtime polymorphism is achieved.**

**Source Code:**

class Vehicle

{

public void speed()

{

System.out.println("Vehicle speed = 60km/h");

}

}

class Bike extends Vehicle

{

public void speed()

{

System.out.println("Bike speed = 40km/h");

}

}

class Experiment5

{

public static void main(String[] args)

{

Vehicle obj = new Bike();

obj.speed();

}

}

**OUTPUT:**

**Experiment-6:**

**Write a Java Program to create and demonstrate packages.**

**Source Code:**

**Program1: - MyMath.java**

package mypack;

public class MyMath

{

public void sum(int a, int b)

{

int c=a+b;

System.out.println("Addition = " + c);

}

}

**Note:** Compile package program with -d (it represents destination directory) and . (it represents the current folder/Path also can be mentioned in place of “.”).

**Syntax:-** javac -d . <Filename.Java>

**Example: -** javac -d . MyMath.java

**Program2: - Experiment6.java**

java import mypack.MyMath;

import java.util.Scanner;

public class Experiment6

{

public static void main(String args[])

{

MyMath obj = new MyMath();

Scanner sc=new Scanner(System.in);

int n1,n2;

System.out.println("Enter two numbers\n");

n1=sc.nextInt();

n2=sc.nextInt();

obj.sum(n1,n2);

}

}

**OUTPUT:**

**Experiment-7:**

**Write a Java Program, using String Tokenizer class, which reads a line of integers and then displays each integer and the sum of all integers.**

**Source Code:**

import java.util.StringTokenizer;

import java.util.Scanner;

class Experiment7

{

public static void main(String args[])

{

int n, sum = 0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter some integers with one space gap:");

String str = sc.nextLine();

StringTokenizer st = new StringTokenizer(str,":");

System.out.println("Given Integers are");

while (st.hasMoreTokens())

{

String temp = st.nextToken();

n = Integer.parseInt(temp);

System.out.println(n);

sum = sum + n;

}

System.out.println("Sum of all integers = " + sum);

}

}

**OUTPUT:**

**Experiment-8:**

**Write a Java Program that reads on file name form the user then displays information about whether the file exists, whether the file is readable/writable, the type of file and the length of the file in bytes and display the content of the using File Input Stream class.**

**Source Code:**

import java.io.\*;

import java.util.Scanner;

class Experiment8

{

public static void main(String args[])throws Exception

{

String fn;

Scanner sc = new Scanner(System.in);

System.out.println("Enter a file name ");

fn=sc.next();

File f1 = new File(fn);

System.out.println("File Name: " + f1.getName());

System.out.println("Absolute Path: " + f1.getAbsolutePath());

System.out.println(f1.exists() ? "The file exists" : "The file does not exist");

System.out.println(f1.canWrite() ? "It is writeable" : "It is not writeable");

System.out.println(f1.canRead() ? "It is readable" : "It is not readable");

System.out.println(f1.isDirectory() ? "It is a directory " : "It is not a directory");

System.out.println(f1.isFile() ? "It is normal file" : "It might be a named pipe");

System.out.println(f1.isAbsolute() ? "It is an absolute file" : "It is not absolute file");

System.out.println("File size: " + f1.length() + " Bytes");

FileInputStream f = new FileInputStream(f1);

int n=f.available();

int nl=0,nw=0;

char ch;

System.out.println("Contents of the file: -\n");

for(int i=0;i<n;i++)

{

ch=(char)f.read();

System.out.print(ch);

if(ch=='\n')

nl++;

else if(ch==' ')

nw++;

}

System.out.println("\nNumber of lines : "+nl);

System.out.println("\nNumber of words : "+(nl+nw));

System.out.println("\nNumber of characters : "+n);

}

}

**OUTPUT:**

**Experiment-9:**

**Write a Java Program that displays the number of characters, lines, and words in a text/text file.**

**Source Code:**

import java.util.Scanner;

import java.io.\*;

class Experiment9

{

public static void main(String args[]) throws IOException

{

int nl=1,nw=0,nc=0; char ch;

String str;

Scanner sc = new Scanner(System.in);

System.out.println("Enter File name ");

str=sc.nextLine();

FileInputStream f = new FileInputStream(str);

int n=f.available();

for(int i=0;i<n;i++)

{

ch=(char)f.read();

if(ch=='\n')

nl++;

else if(ch==' ')

nw++;

}

System.out.println("\nNumber of lines : "+nl);

System.out.println("\nNumber of words : "+(nl+nw));

System.out.println("\nNumber of characters : "+n);

}

}

**OUTPUT:**

**Experiment-10:**

**Write an Applet that displays the content of a file.**

**Source Code:**

**Part1: -**

Create the following text file in the same folder in which you want save applet program.

**Aditya.txt: -**

Aditya is the best educational institution in AP.

**Part2: -**

Program: - Experiment10.java

import java.applet.\*;

import java.awt.\*;

import java.io.\*;

/\* <applet code="Experiment10.class" height="300" width="500"> </applet> \*/

public class Experiment10 extends Applet

{

String content = "";

public void init()

{

try

{

char ch;

StringBuffer buff = new StringBuffer("");

FileInputStream fis = new FileInputStream("Aditya.txt");

while(fis.available()!=0)

{

ch = (char) fis.read();

buff.append(ch);

}

fis.close();

content = new String(buff);

}

catch(FileNotFoundException e)

{

content = "Cannot find the specified file...";

}

catch(IOException i)

{

content = "Cannot read from the file...";

}

}

public void paint(Graphics g)

{

Font f = new Font("Courier", Font.PLAIN, 30);

g.setFont(f);

g.drawString(content,50,50);

}

}

**OUTPUT:**

**Experiment-11:**

**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.**

**Source Code:**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*<applet code="Experiment11.class" width=300 height=300></applet>\*/

public class Experiment11 extends Applet implements ActionListener

{

String msg=" ";

int v1, v2, result;

TextField t1;

Button b[] = new Button[10];

Button add,sub,mul,div,clear,mod,EQ,BS,dot;

char OP;

public void init()

{

Color k=new Color(120,89,90);

setBackground(k);

t1 = new TextField(50);

GridLayout gl = new GridLayout(4,5);

setLayout(gl);

for(int i=0 ; i<10 ; i++)

{

b[i]=new Button("" + i);

}

add=new Button("+");

sub=new Button("-");

mul=new Button("X");

div=new Button("/");

mod=new Button("%");

clear=new Button("clear");

EQ=new Button("=");

BS=new Button("Back Space");

dot=new Button(".");

t1.addActionListener(this);

add(t1);

add(BS);

for(int i=0;i<10;i++)

{

add(b[i]);

}

add(add);

add(sub);

add(mul);

add(div);

add(mod);

add(clear);

add(dot);

add(EQ);

for(int i=0;i<10;i++)

{

b[i].addActionListener(this);

}

add.addActionListener(this);

sub.addActionListener(this);

mul.addActionListener(this);

div.addActionListener(this);

mod.addActionListener(this);

clear.addActionListener(this);

EQ.addActionListener(this);

BS.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

String str = ae.getActionCommand();

char ch = str.charAt(0);

if ( Character.isDigit(ch))

t1.setText(t1.getText()+str);

else if(str.equals("+"))

{

v1=Integer.parseInt(t1.getText()); OP='+';

t1.setText("");

}

else if(str.equals("-"))

{

v1=Integer.parseInt(t1.getText()); OP='-';

t1.setText("");

}

else if(str.equals("X"))

{

v1=Integer.parseInt(t1.getText()); OP='\*';

t1.setText("");

}

else if(str.equals("/"))

{

v1=Integer.parseInt(t1.getText());

OP='/';

t1.setText("");

}

else if(str.equals("%"))

{

v1=Integer.parseInt(t1.getText());

OP='%';

t1.setText("");

}

else if(str.equals("Back Space"))

{

String s=t1.getText();

s=s.substring(0,s.length()-1);

t1.setText(s);

}

if(str.equals("="))

{

v2=Integer.parseInt(t1.getText());

if(OP=='+')

result=v1+v2;

else if(OP=='-')

result=v1-v2;

else if(OP=='\*')

result=v1\*v2;

else if(OP=='/')

result=v1/v2;

else if(OP=='%')

result=v1%v2;

t1.setText(""+result);

}

if(str.equals("clear"))

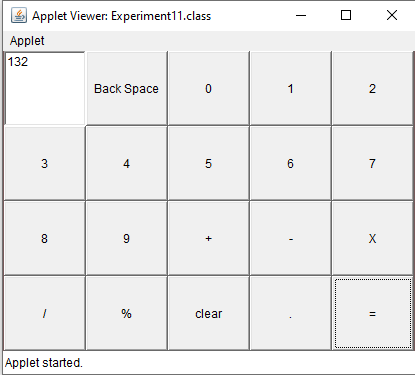
{

t1.setText("");

}

}

}

**OUTPUT:  
**

**Experiment12:**

**Write a Java Program for handling mouse events.**

**Source Code:**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\* <applet code="Experiment12.class" width=300 height=100> </applet> \*/

public class Experiment12 extends Applet implements MouseListener, MouseMotionListener

{

String msg = "";

int mouseX = 0, mouseY = 0; // coordinates of mouse

public void init()

{

addMouseListener(this);

addMouseMotionListener(this);

}

public void mouseClicked(MouseEvent me)

{

mouseX = 0;

mouseY = 10;

msg = "Mouse clicked.";

repaint();

}

public void mouseEntered(MouseEvent me)

{

mouseX = 0;

mouseY = 10;

msg = "Mouse entered.";

repaint();

}

public void mouseExited(MouseEvent me)

{

mouseX = 0;

mouseY = 10;

msg = "Mouse exited.";

repaint();

}

public void mousePressed(MouseEvent me)

{

mouseX = me.getX();

mouseY = me.getY();

msg = "Pressed";

repaint();

}

public void mouseReleased(MouseEvent me)

{

mouseX = me.getX();

mouseY = me.getY();

msg = "Released";

repaint();

}

public void mouseDragged(MouseEvent me)

{

mouseX = me.getX();

mouseY = me.getY();

msg = "\*";

showStatus("Dragging mouse at " + mouseX + ", " + mouseY);

repaint();

}

public void mouseMoved(MouseEvent me)

{

showStatus("Moving mouse at " + me.getX() + ", " + me.getY());

}

public void paint(Graphics g)

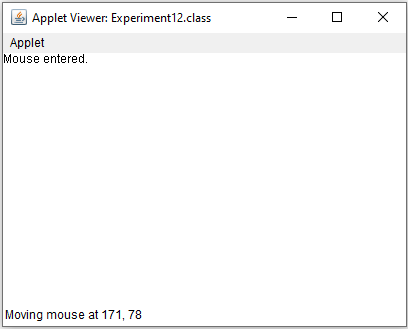
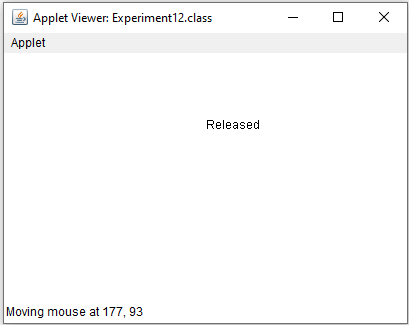
{

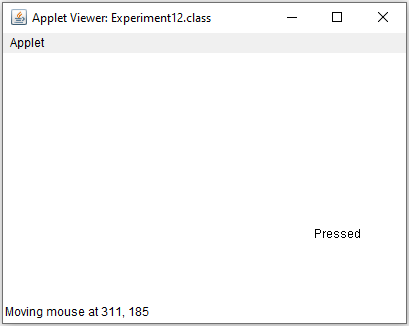
g.drawString(msg, mouseX, mouseY);

}

}

OUTPUT:



**Experiment13:**

**Write a Java Program demonstrating the life cycle of a thread.**

**Source Code:**

class A1 extends Thread

{

public void run ()

{

System.out.println ("Thread A");

System.out.println ("i in Thread A ");

for (int i = 1; i <= 5; i++)

{

System.out.println ("i = " + i);

try

{

Thread.sleep (1000);

}

catch (InterruptedException e)

{

e.printStackTrace ();

}

}

System.out.println ("Thread A Completed.");

}

}

class B extends Thread

{

public void run ()

{

System.out.println ("Thread B");

System.out.println ("i in Thread B ");

for (int i = 1; i <= 5; i++)

{

System.out.println ("i = " + i);

}

System.out.println ("Thread B Completed.");

}

}

public class Experiment13

{

public static void main (String[]args)

{

**// life cycle of Thread**

**// Thread's New State**

A1 threadA = new A1 ();

B threadB = new B ();

**// Both the above threads are in runnable state**

// Running state of thread A & B

threadA.start ();

**// Move control to another thread**

threadA.yield ();

**// Blocked State of thread B**

try

{

threadA.sleep (1000);

}

catch (InterruptedException e)

{

e.printStackTrace ();

}

threadB.start ();

System.out.println ("Main Thread End");

}

}

**OUTPUT:**

**Experiment-14:**

**Write a Java Program that lets users create Pie charts. Design your own user interface (with Swings & AWT).**

**Source Code:**

import java.awt.Color;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Rectangle;

import javax.swing.JComponent;

import javax.swing.JFrame;

class Slice

{

double value;

Color color;

public Slice(double value, Color color)

{

this.value = value;

this.color = color;

}

}

class MyComponent extends JComponent

{

Slice[] slices = {

new Slice(5, Color.black), new Slice(33, Color.green), new Slice(20, Color.yellow), new Slice(15, Color.red)

};

MyComponent() {}

public void paint(Graphics g)

{

drawPie((Graphics2D) g, getBounds(), slices);

}

void drawPie(Graphics2D g, Rectangle area, Slice[] slices)

{

double total = 0.0D;

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

total += slices[i].value;

}

double curValue = 0.0D;

int startAngle = 0;

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

startAngle = (int) (curValue \* 360 / total);

int arcAngle = (int) (slices[i].value \* 360 / total);

g.setColor(slices[i].color);

g.fillArc(area.x, area.y, area.width, area.height, startAngle, arcAngle);

curValue += slices[i].value;

}

}

}

public class Experiment14 {

public static void main(String[] argv) {

JFrame frame = new JFrame();

frame.getContentPane().add(new MyComponent());

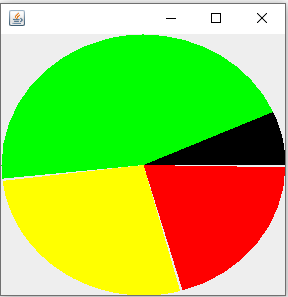
frame.setSize(300, 300);

frame.setVisible(true);

}

}

OUTPUT:



**Experiment-15:**

**Write a Java Program to implement a Queue, using user defined Exception Handling (also make use of throw, throws).**

**Source Code:**

import java.util.\*;

import java.lang.\*;

class QueueError extends Exception

{

public QueueError(String msg)

{

super(msg);

}

}

class Que

{

private int size;

private int front = -1;

private int rear = -1;

private Integer[] queArr;

public Que(int size)

{

this.size = size;

queArr = new Integer[size];

}

public void insert(int item) throws Exception,QueueError

{

try

{

if(rear == size-1)

{

throw new QueueError("Queue Overflow");

}

else if(front==-1)

{

rear++;

queArr[rear] = item;

front = rear;

}

else

{

rear++;

queArr[rear] = item;

}

}

catch(QueueError qe)

{

qe.printStackTrace();

}

}

public void delete() throws Exception,QueueError

{

try

{

if(front == -1)

{

throw new QueueError("Queue Underflow");

}

else if(front==rear)

{

System.out.println("\nRemoved "+queArr[front]+" from Queue");

queArr[front] = null;

front--;

rear--;

}

else

{

System.out.println("\nRemoved "+queArr[front]+" from Queue");

queArr[front] = null;

for(int i=front+1;i<=rear;i++)

{

queArr[i-1]=queArr[i];

}

rear--;

}

}

catch(QueueError qe)

{

qe.printStackTrace();

}

}

public void display() throws Exception,QueueError

{

try

{

if(front==-1)

throw new QueueError("Queue is Empty");

else

{

System.out.print("\nQueue is: ");

for(int i=front;i<=rear;i++)

{

System.out.print(queArr[i]+"\t");

}

System.out.println();

}

}

catch(QueueError qe)

{

qe.printStackTrace();

}

}

}

class Experiment15

{

public static void main(String[] args) throws Exception,QueueError

{

System.out.println("\n\n\tQueue test using Array\n\n");

Scanner scan = new Scanner(System.in);

System.out.print("Enter size of Queue array: ");

int size = scan.nextInt();

Que que = new Que(size);

char ch;

try

{

while(true)

{

System.out.println("\n\n\tQueue operations \n");

System.out.println("1. Insert");

System.out.println("2. Delete");

System.out.println("3. Display");

System.out.println("4. Exit\n");

System.out.print("Enter your choice: ");

int choice = scan.nextInt();

switch(choice)

{

case 1: System.out.print("\nEnter integer number to insert: ");

que.insert(scan.nextInt());

break;

case 2:que.delete();

break;

case 3:que.display();

break;

case 4:return ;

}

}

}

catch(QueueError qe)

{

qe.printStackTrace();

}

}

}

**OUTPUT:**

**Experiment-16:**

**Write Java Program to demonstrating thread synchronization in multithreading.**

**Source Code:**

import java.lang.\*;

class College

{

public synchronized void classRoom (String fn)

{

for (int i=1 ; i<10 ; i++)

{

System.out.println(i + " class taken by " + fn);

try

{

Thread.sleep(1000);

}

catch(InterruptedException e){}

}

System.out.println(fn + " task completed\n");

}

}

class MyThread extends Thread

{

College c;

String faculty;

MyThread(College obj, String name)

{

c=obj;

faculty=name;

}

public void run()

{

c.classRoom(faculty);

}

}

class Experiment16

{

public static void main(String args[])

{

College x = new College();

MyThread t1 = new MyThread(x,"Babuji Sir");

MyThread t2 = new MyThread(x,"Kumar Sir");

MyThread t3 = new MyThread(x,"Pradeep Sir");

t1.start();

t2.start();

t3.start();

}

}

**OUTPUT:**