

### Program 13

Create a package **pack1** having one class **C1** and one interface **I1**. Class **C1** has two methods **int sum(int, int)** and **int sub(int, int)**. The **I1** has one method **int division(int, int)**. Create another package **pack2** having class **C2**. Reuse **C1** and **I1** in **C2** and show the results.

**Note: Use appropriate Access Modifiers as required.**

#### Source Code:

```
package p1;
public class C1
{
    public int sum(int x,int y)
    {
        return(x+y);
    }
    public int sub(int a, int b)
    {
        return(a-b);
    }
}
```

```
package p1;
public interface I1
{
    int div(int a ,int b);
}
```

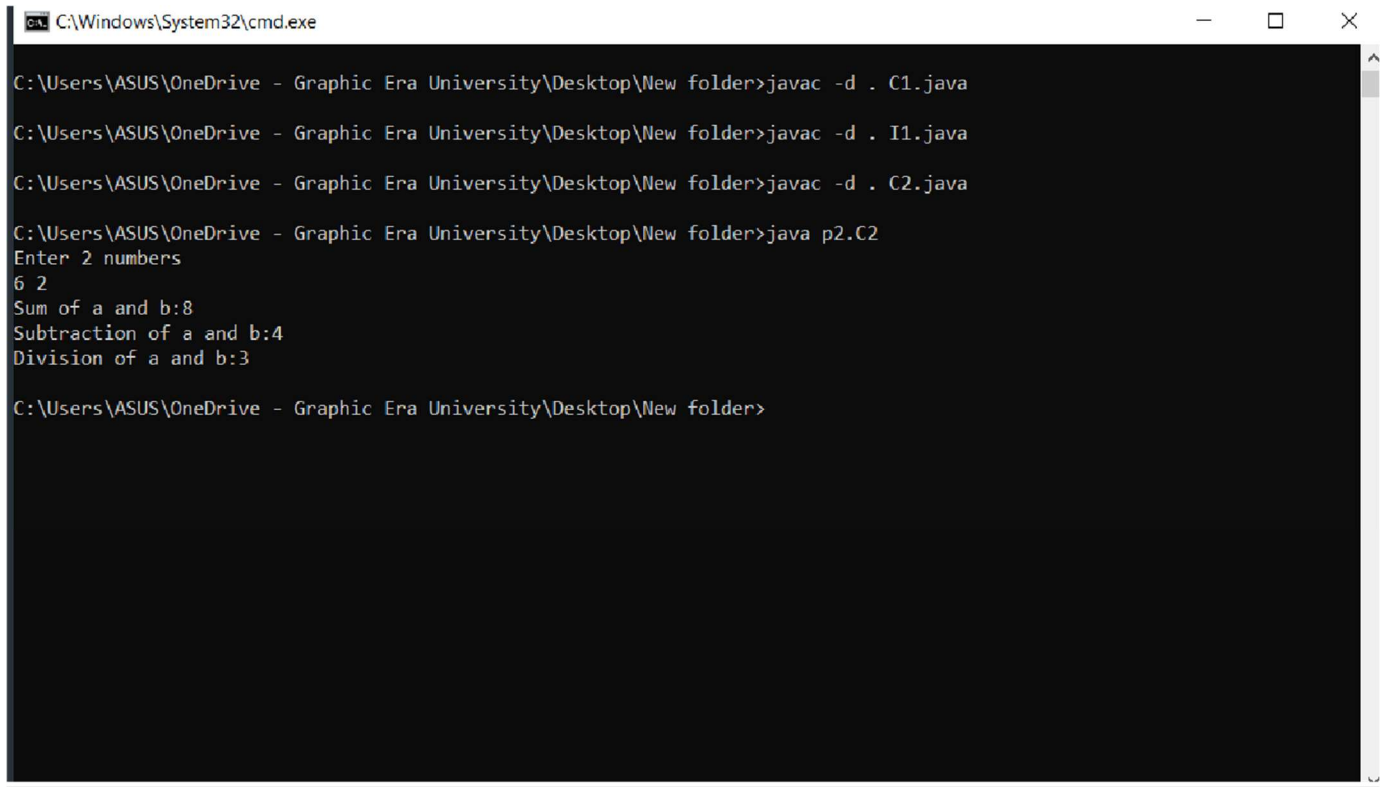
```
package p2;
import p1.*;
import java.util.*;
public class C2 implements I1
{
    public int div(int a ,int b)
    {
        if(b!=0)
        {
```

```
        return(a/b);
    }
    else
    {
        return(-1);
    }
}
public static void main(String args[])
{
    Scanner in=new Scanner(System.in);
    C1 d=new C1();
    C2 f=new C2();

    System.out.println("Enter 2 numbers");
    int a=in.nextInt();
    int b=in.nextInt();

    System.out.println("Sum of a and b:"+d.sum(a,b));
    System.out.println("Subtraction of a and b:"+d.sub(a,b));
    System.out.println("Division of a and b:"+f.div(a,b));
}
}
```

## OUTPUT



```
C:\Windows\System32\cmd.exe

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac -d . C1.java
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac -d . I1.java
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac -d . C2.java
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java p2.C2
Enter 2 numbers
6 2
Sum of a and b:8
Subtraction of a and b:4
Division of a and b:3
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>
```

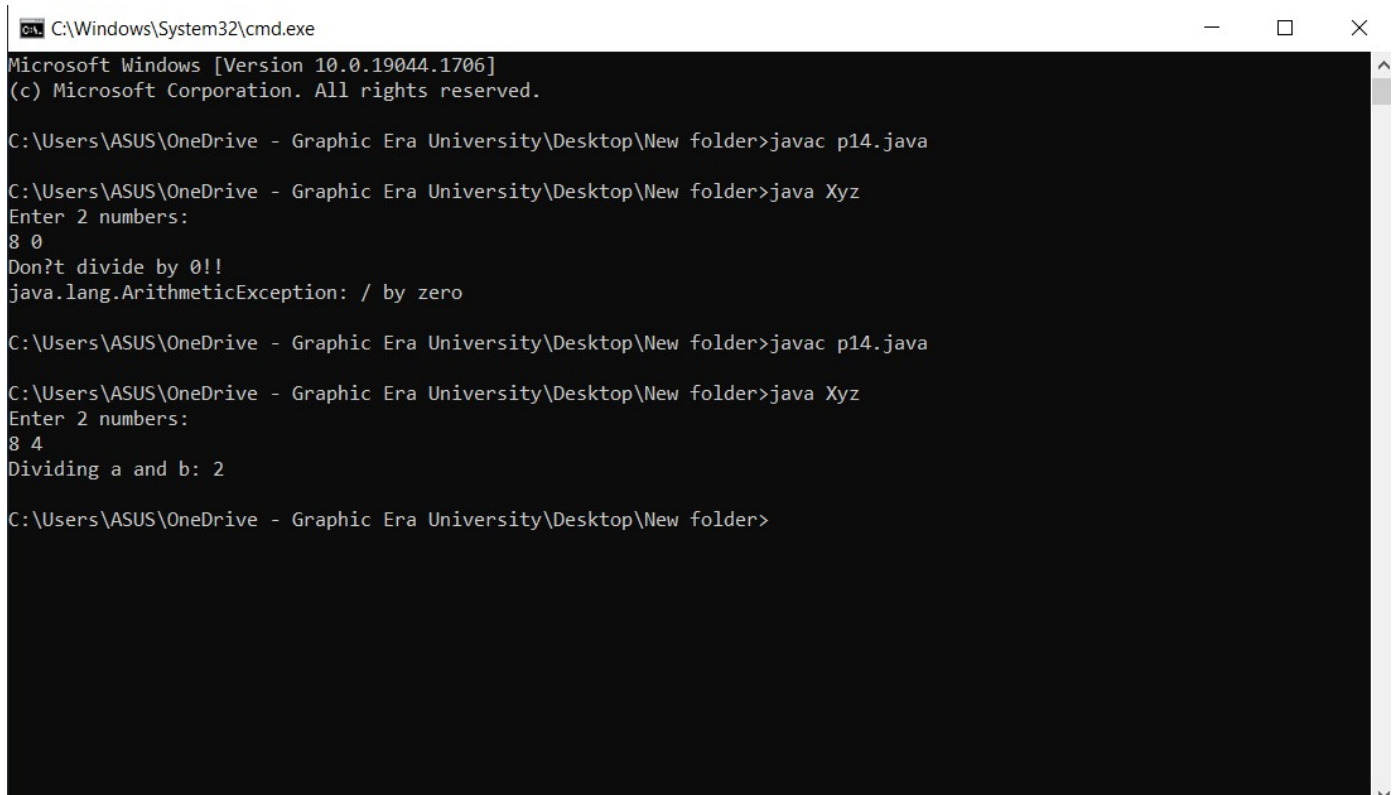
**Program 14**

Write a program to divide two numbers with proper exception handlers.

**Source Code:**

```
import java.util.*;
class Xyz
{
    public static void main(String args[])
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("Enter 2 numbers:");
        int a=sc.nextInt();
        int b=sc.nextInt();
        try{
            System.out.println("Dividing a and b: "+(a/b));
        }
        catch(ArithmeticException e)
        {
            System.out.println("Don't divide by 0!!\n"+e);
        }
    }
}
```

# OUTPUT



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p14.java

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Xyz
Enter 2 numbers:
8 0
Don?t divide by 0!!
java.lang.ArithmeticException: / by zero

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p14.java

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Xyz
Enter 2 numbers:
8 4
Dividing a and b: 2

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>
```

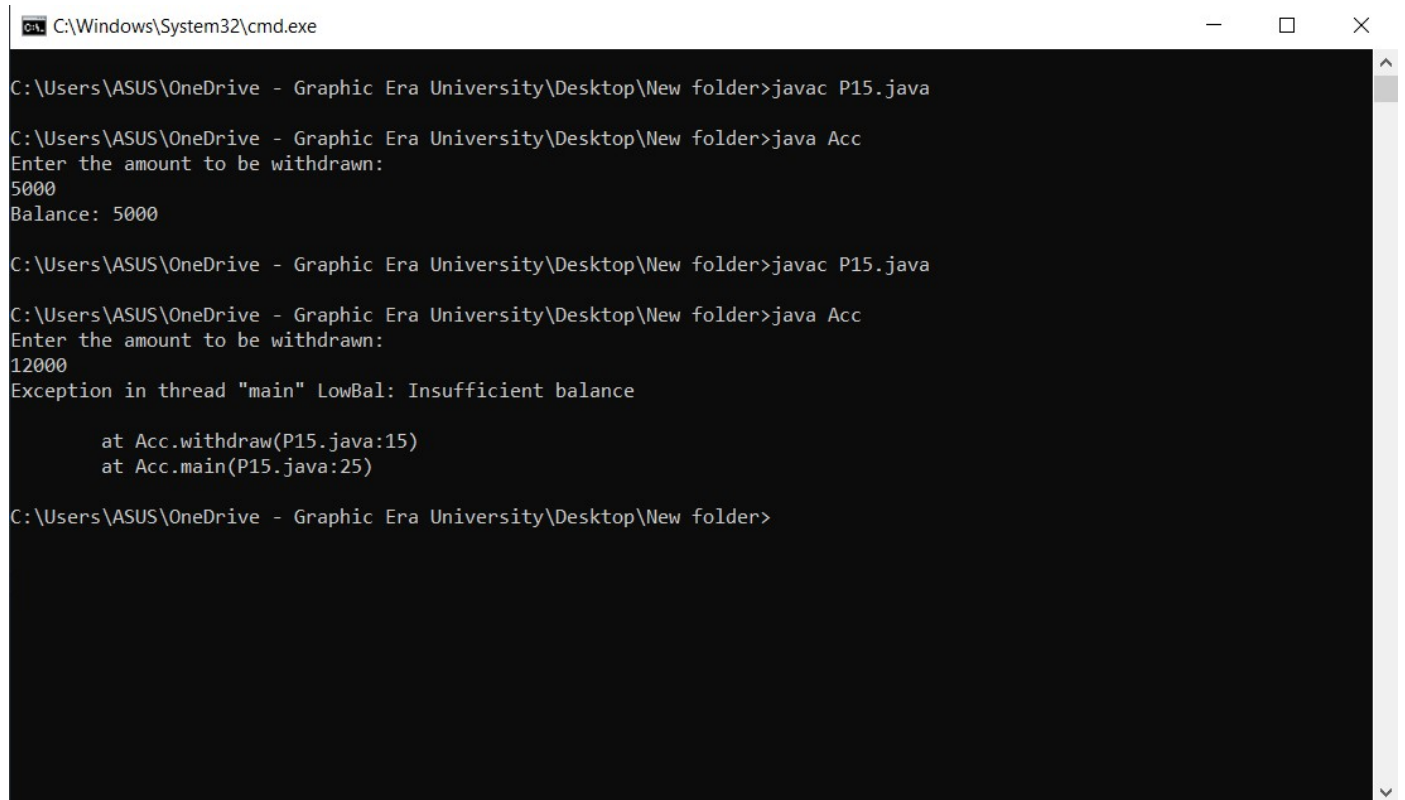
### Program 15

Create LowBalanceException that occurs when user tries to withdraw some amount that is greater than his current bank balance. To withdraw you have to write a **void withdrawal(int amount)** method.

#### Source Code:

```
import java.util.*;
class LowBalanceException extends RuntimeException
{
    LowBalanceException(String s)
    {
        super(s);
    }
}
class Acc
{
    int bal=10000;
    void withdraw(int amt)
    {
        if(amt>bal)
            throw new LowBalanceException("Insufficient balance\n");
        else
            bal-=amt;
    }
    public static void main(String args[])
    {
        Acc d=new Acc();
        Scanner in= new Scanner(System.in);
        System.out.println("Enter the amount to be withdrawn: ");
        int amt=in.nextInt();
        d.withdraw(amt);
        System.out.println("Balance: "+d.bal);
    }
}
```

## OUTPUT



```
C:\Windows\System32\cmd.exe

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac P15.java

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Acc
Enter the amount to be withdrawn:
5000
Balance: 5000

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac P15.java

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Acc
Enter the amount to be withdrawn:
12000
Exception in thread "main" LowBal: Insufficient balance

    at Acc.withdraw(P15.java:15)
    at Acc.main(P15.java:25)

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>
```

**Program 16**

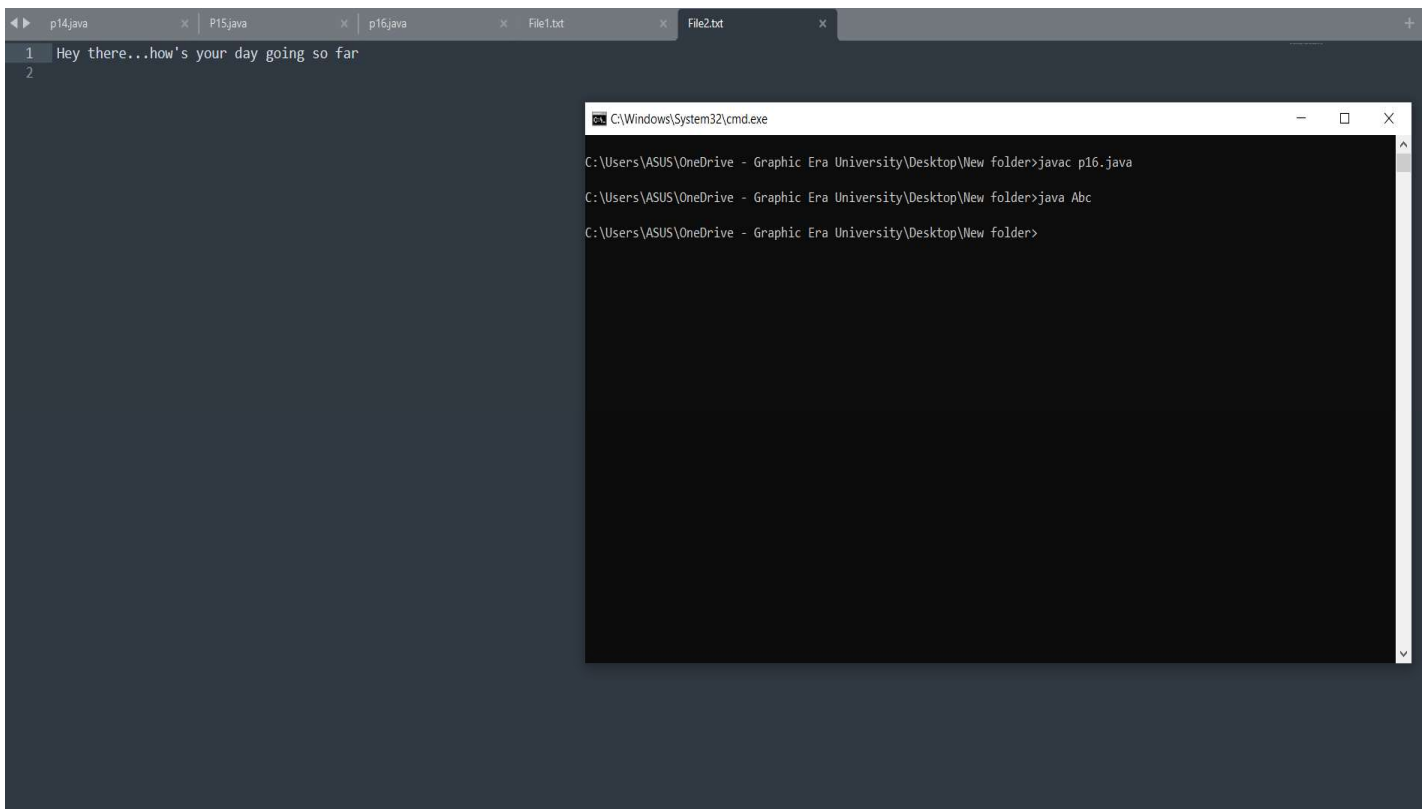
Write a program that reads from a text file byte by byte and writes in some another file. Write this program in an efficient way.

**Source Code:**

```
import java.io.*;
class Abc
{
    public static void main(String args[])throws IOException
    {
        FileInputStream fis=new FileInputStream("file1.txt");
        BufferedInputStream bis=new BufferedInputStream(fis);
        FileOutputStream fos=new FileOutputStream("file2.txt");
        BufferedOutputStream bos=new BufferedOutputStream(fos);
        int i=0;
        while((i=bis.read())!=-1)
        {
            bos.write(i);
        }
        System.out.println();
        fis.close();
        bis.close();
        fos.close();
        bos.close();
    }
}
```



## OUTPUT



The screenshot displays an IDE with several tabs: p14.java, P15.java, p16.java, File1.txt, File2.txt, and an unnamed tab. The p16.java tab is active, showing two lines of code: `1 Hey there...how's your day going so far` and `2`. Overlaid on the IDE is a Windows command prompt window titled `C:\Windows\System32\cmd.exe`. The command prompt shows the following commands and their outputs: `C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p16.java`, `C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Abc`, and `C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>`.

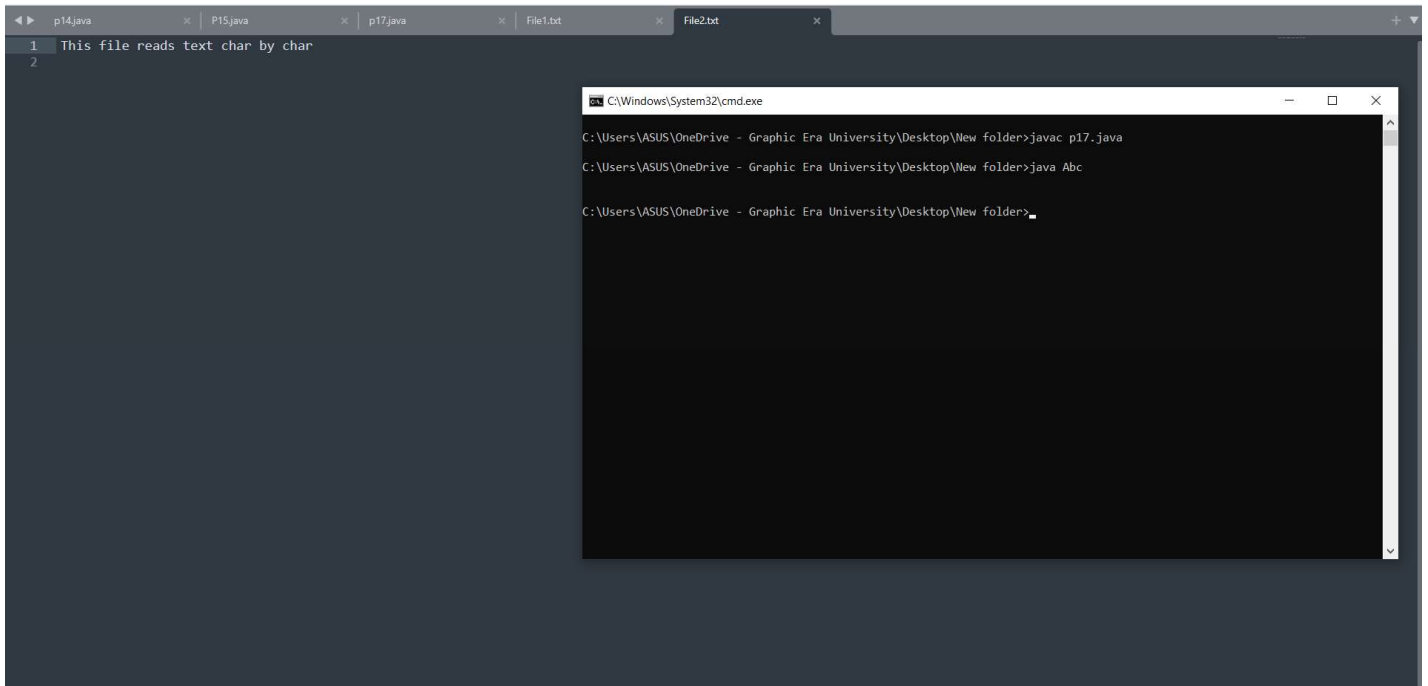
**Program 17**

Write a program that reads from a text file char by char and writes in some another file. Write this program in an efficient way.

**Source Code:**

```
import java.io.*;
class Abc
{
    public static void main(String args[])throws IOException
    {
        FileReader fr=new FileReader("file1.txt");
        BufferedReader br=new BufferedReader(fr);
        FileWriter fw=new FileWriter("file2.txt");
        BufferedWriter bw=new BufferedWriter(fw);
        int i=0;
        while((i=br.read())!=-1)
        {
            bw.write((char)i);
        }
        System.out.println();
        fr.close();
        br.close();
        fw.close();
        bw.close();
    }
}
```

# OUTPUT



The image shows a screenshot of a development environment. On the left, an IDE window with multiple tabs (p14.java, P15.java, p17.java, File1.txt, File2.txt) displays a Java file with two lines of code: `1 This file reads text char by char` and `2`. On the right, a Windows Command Prompt window titled `C:\Windows\System32\cmd.exe` shows the execution of two Java commands: `C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p17.java` and `C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Abc`. The prompt is currently on a new line.

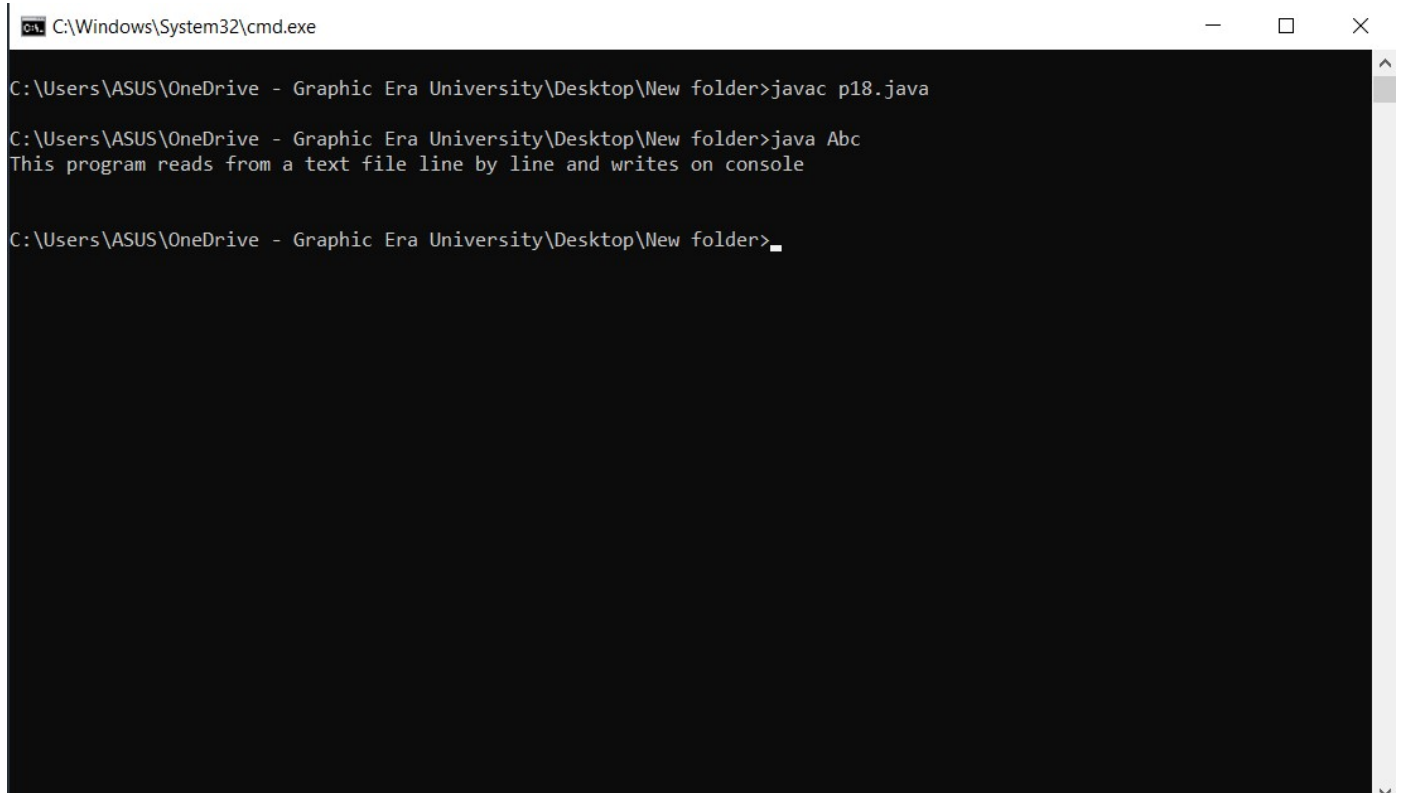
**Program 18**

Write a program that reads from a text file line by line and writes on console.

**Source Code:**

```
import java.io.*;
class Abc
{
    public static void main(String args[])throws IOException
    {
        FileReader fr=new FileReader("file1.txt");
        BufferedReader br=new BufferedReader(fr);
        String i;
        while((i=br.readLine())!=null)
        {
            System.out.println(i);
        }
        System.out.println();
        fr.close();
        br.close();
    }
}
```

## OUTPUT



```
C:\Windows\System32\cmd.exe

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p18.java

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Abc
This program reads from a text file line by line and writes on console

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>_
```

### Program 19

Write a program that take your name from keyboard and writes in some text file.

#### **Source Code:**

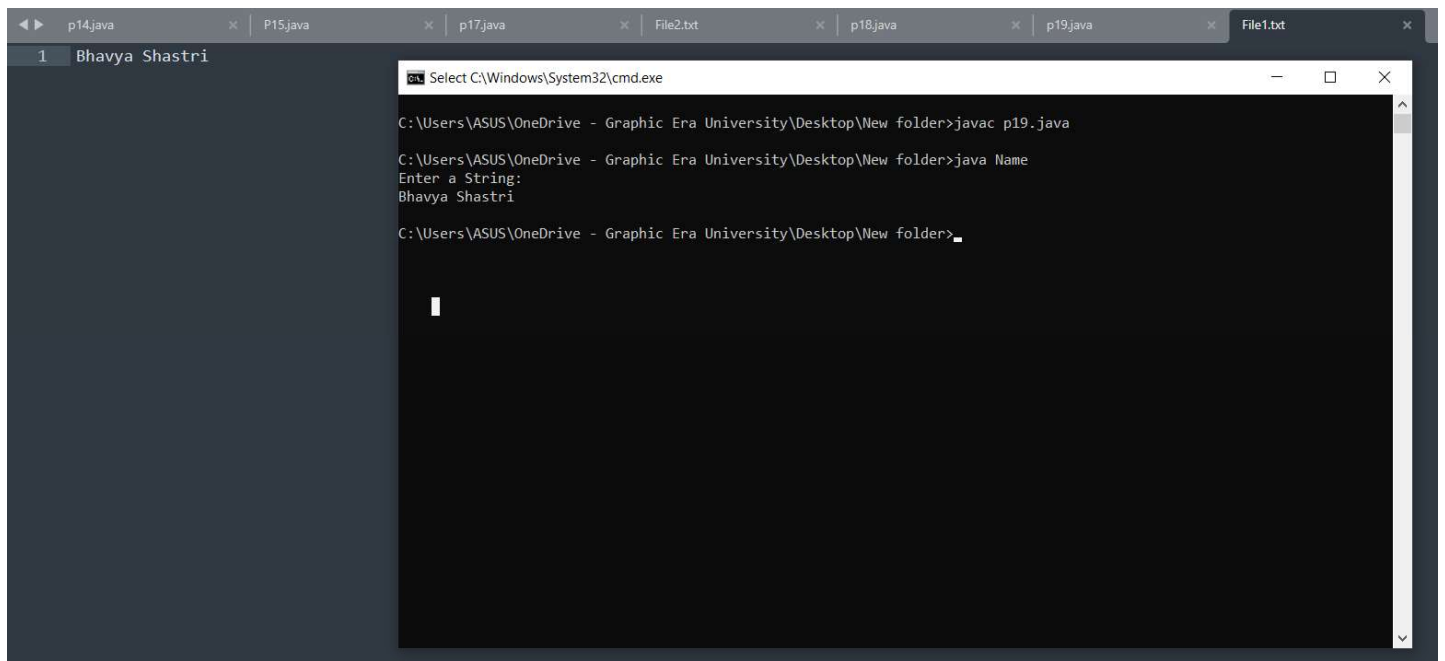
```
import java.io.*;
class Name
{
    public static void main(String args[])throws IOException
    {
        InputStreamReader isr=new InputStreamReader(System.in);
        BufferedReader br=new BufferedReader(isr);

        FileOutputStream fos=new FileOutputStream("File1.txt");
        BufferedOutputStream bos=new BufferedOutputStream(fos);

        System.out.println("Enter a String:");
        String str=br.readLine();
        byte b[]=str.getBytes();

        bos.write(b);
        bos.close();
        fos.close();
        br.close();
        isr.close();
    }
}
```

## OUTPUT



```
p14.java x P15.java x p17.java x File2.txt x p18.java x p19.java x File1.txt x
1 Bhavya Shastri

Select C:\Windows\System32\cmd.exe

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p19.java
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Name
Enter a String:
Bhavya Shastri
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>
```

## Program 20

Write a multithreaded program where three threads are there and printing the numbers from 1 to 10 concurrently.

### Source Code:

```
class A extends Thread{
    public void run(){
        for(int i=1;i<=10;i++){
            System.out.print("From A: "+i+" ");
        }
    }
}
class B extends Thread{
    public void run(){
        for(int j=1;j<=10;j++){
            System.out.println("From B: "+j+" ");
        }
    }
}
class C extends Thread {
    public void run(){
        for(int k=1;k<=10;k++){
            System.out.println("From C: "+k+" ");
        }
    }
}
class XYZ{
    public static void main(String args[]){
        A a=new A();
        B b=new B();
        C c=new C();
        a.start();
        b.start();
        c.start();
    }
}
```



## OUTPUT

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p20.java

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java XYZ
From A: 1 From A: 2 From A: 3 From C: 1
From C: 2
From B: 1
From C: 3
From A: 4 From A: 5 From A: 6 From A: 7 From A: 8 From A: 9 From A: 10 From C: 4
From C: 5
From B: 2
From C: 6
From B: 3
From C: 7
From C: 8
From B: 4
From B: 5
From C: 9
From B: 6
From B: 7
From C: 10
From B: 8
From B: 9
From B: 10

C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>_
```

### Program 21

Write a program to set and get the name of threads also set and get the priority of threads.

#### **Source Code:**

```
class A extends Thread
{
    public void run()
    {
        for(int i=1;i<=10;i++)
        {
            System.out.print("From A: "+i+" ");
        }
    }
}
class B extends Thread
{
    public void run(){
        for(int j=1;j<=10;j++){
            System.out.println("From B: "+j+" ");
        }
    }
}
class C extends Thread
{
    public void run(){
        for(int k=1;k<=10;k++){
            System.out.println("From C: "+k+" ");
        }
    }
}
class Test
{
    public static void main(String args[])
    {
```

```

A a=new A();
B b=new B();
C c=new C();

```

```

System.out.println("Getting Thread A Name: "+a.getName());
System.out.println("Getting Thread B Name: "+b.getName());
System.out.println("Getting Thread C Name: "+c.getName());

```

```

a.setName("Thread_1");
b.setName("Thread_2");
c.setName("Thread_3");
System.out.println("Getting Thread A Name after setting:
"+a.getName());
System.out.println("Getting Thread B Name after setting:
"+b.getName());
System.out.println("Getting Thread C Name after setting:
"+c.getName());

```

```

System.out.println("Getting Thread A Priority: "+a.getPriority());
System.out.println("Getting Thread B Priority: "+b.getPriority());
System.out.println("Getting Thread C Priority: "+c.getPriority());

```

```

a.setPriority(1);
b.setPriority(9);
c.setPriority(7);
System.out.println("Getting Thread A Priority after setting:
"+a.getPriority());
System.out.println("Getting Thread B Priority after setting:
"+b.getPriority());
System.out.println("Getting Thread C Priority after setting:
"+c.getPriority());

}
}

```

## OUTPUT

```
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p20.java
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Test
Getting Thread A Name: Thread-0
Getting Thread B Name: Thread-1
Getting Thread C Name: Thread-2
Getting Thread A Name after setting: Thread_1
Getting Thread B Name after setting: Thread_2
Getting Thread C Name after setting: Thread_3
Getting Thread A Priority: 5
Getting Thread B Priority: 5
Getting Thread C Priority: 5
Getting Thread A Priority after setting: 1
Getting Thread B Priority after setting: 9
Getting Thread C Priority after setting: 7
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>_
```

## Program 22

Write a class Display having void wish(String name) methods that wishes hello to given string name. Between printing hello and provided string name apply delay of 500 milliseconds. Suppose multiple threads are there and they are trying to access this wish() method concurrently on **same object** then irregular output will be there. Write this application in such a way so that output becomes regular.

### Source Code:

```
class Display
{
    public synchronized void wish(String s)
    {
        for(int i=1;i<=5;i++)
        {
            System.out.println(s);
            try
            {
                Thread.sleep(500);
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }
}

class MyThread extends Thread
{
    Display d;
    String name;
    MyThread(Display d,String name)
    {
        this.d=d;
        this.name=name;
    }
    public void run()
```

```
        {  
            d.wish(name);  
        }  
    }  
class Demo  
{  
    public static void main(String args[])  
    {  
        Display d= new Display();  
        MyThread t1=new MyThread(d,"Bhavya");  
        t1.start();  
        MyThread t2=new MyThread(d,"Shastri");  
        t2.start();  
    }  
}
```

## OUTPUT

```
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p22.java
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Demo
Bhavya
Bhavya
Bhavya
Bhavya
Bhavya
Shastri
Shastri
Shastri
Shastri
Shastri
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>_
```

### Program 23

Write a class Display having **synchronized void wish(String)** methods that wishes hello to given string name. Between printing hello and provided string name apply delay of 500 milliseconds. Suppose multiple threads are there and they are trying to access this wish() method concurrently on **different objects** then irregular output will be there. Write this application in such a way so that output becomes regular.

#### Source Code:

```
class Display
{
    public static synchronized void wish(String s)
    {
        for(int i=1;i<=5;i++)
        {
            System.out.println(s);
            try
            {
                Thread.sleep(500);
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }
}

class MyThread extends Thread
{
    Display d;
    String name;
    MyThread(Display d,String name)
    {
        this.d=d;
        this.name=name;
    }
}
```



```
    }  
    public void run()  
    {  
        d.wish(name);  
    }  
}  
class Demo  
{  
    public static void main(String args[])  
    {  
        Display d1= new Display();  
        Display d2= new Display();  
        MyThread t1=new MyThread(d1,"Bhavya");  
        t1.start();  
        MyThread t2=new MyThread(d2,"Shastri");  
        t2.start();  
    }  
}
```

## OUTPUT

```
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p22.java
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Demo
Bhavya
Bhavya
Bhavya
Bhavya
Bhavya
Shastri
Shastri
Shastri
Shastri
Shastri
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>_
```

## Program 24

Write a class Customer having **balance** as a property and **void withdrawal(int amount)**, and **void deposit(int amount)** as instance methods. There are two threads, the first thread wants to withdrawal some amount and second thread wants to deposit some amount. Apply inter thread communication where, if withdrawal amount is higher than current balance then first thread will wait for second thread to deposit then resume the withdrawal.

### Source Code:

```
class Customer
{
    int bal=1000;
    public synchronized void withdrawal(int amt)
    {
        System.out.println("Going to withdraw. ..");
        if(this.bal < amt)
        {
            System.out.println("Less Balance...Kindly wait. ");
            try{
                wait();
            }
            catch(Exception e){
                System.out.println(e);
            }
        }

        this.bal=this.bal - amt;
    }

    public synchronized void deposit(int amt)
    {
        System.out.println("Going to deposit... ");
        this.bal=this.bal + amt;
        System.out.println("Deposited:...And total balance is: "+bal);
        notify();
    }
}
```

```

}

class MyThread1 extends Thread{
    Customer c;
    MyThread1(Customer c){
        this.c=c;
    }
    public void run()
    {
        c.withdrawl(1500);
    }
}

class MyThread2 extends Thread
{
    Customer c;
    MyThread2(Customer c){
        this.c=c;
    }
    public void run()
    {
        c.deposit(1000);
    }
}

class Demo
{
    public static void main(String args[])
    {
        Customer c= new Customer();

        MyThread1 t1=new MyThread1(c);
        t1.start();
        MyThread2 t2=new MyThread2(c);
        t2.start();
    }
}

```

## OUTPUT

```
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p24.java  
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java Demo  
Going to withdraw....  
Less Balance...Kindly wait...  
Going to deposit....  
Deposited:...And total balance is: 2000  
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>_
```

### Program 25

Create a GUI for student's information system. A GUI that asks all the relevant information's related to a student.

#### Source Code:

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

class A extends JFrame
{
    JLabel l1,l2,l3;
    JTextField t1,t2,t3;

    A(){

        l1=new JLabel("Name");
        l2=new JLabel("RollNo");
        l3=new JLabel("Student ID");

        t1=new JTextField(10);
        t2=new JTextField(10);
        t3=new JTextField(10);

        JRadioButton rb1=new JRadioButton("Other");
        JRadioButton rb2=new JRadioButton("Female",true);
        JRadioButton rb3=new JRadioButton("Male");

        JRadioButton rb4=new JRadioButton("Sec: A");
        JRadioButton rb5=new JRadioButton("Sec: B");
        JRadioButton rb6=new JRadioButton("Sec: C");

        JCheckBox cb1=new JCheckBox();
        JCheckBox cb2=new JCheckBox("CSE",true);
        JCheckBox cb3=new JCheckBox("ESE");
        JCheckBox cb4=new JCheckBox("ME");
```

```

        JButton b1=new JButton("Submit");

        setLayout(new FlowLayout());

        add(l1);
        add(t1);
        add(l2);
        add(t2);
        add(rb4);
        add(rb5);
        add(rb6);
    add(cb1);
        add(cb2);
        add(cb3);
        add(cb4);
        add(l3);
        add(t3);
        add(rb1);
        add(rb2);
        add(rb3);
        add(b1);
    }

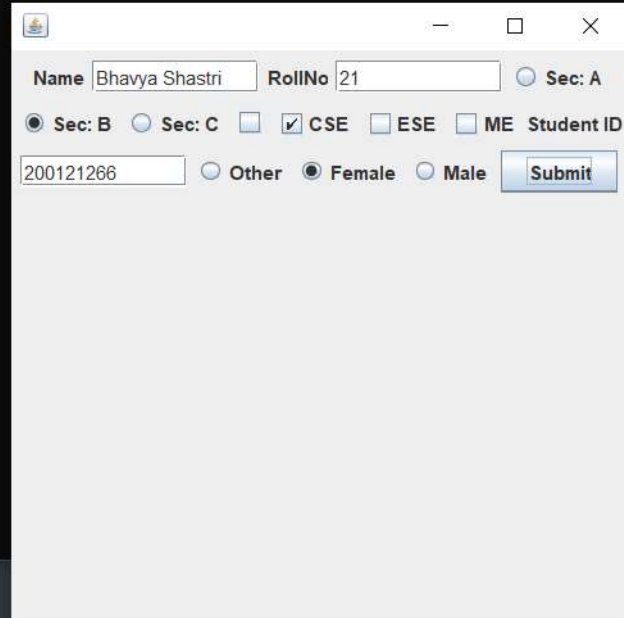
    public static void main(String args[])
    {
        A d=new A();
        d.setSize(400,400);
        d.setVisible(true);
    }

}

```

## OUTPUT

```
Microsoft Windows [Version 10.0.19044.1706]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p25.java  
  
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java A
```



The screenshot shows a Java Swing window titled 'A' with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a student registration form with the following fields and controls:

- Name:** A text field containing 'Bhavya Shastri'.
- RollNo:** A text field containing '21'.
- Sec:** Three radio buttons labeled 'Sec: A', 'Sec: B', and 'Sec: C'. 'Sec: B' is selected.
- Student ID:** A text field containing '200121266'.
- Gender:** Three radio buttons labeled 'Other', 'Female', and 'Male'. 'Female' is selected.
- Course:** Three checkboxes labeled 'CSE', 'ESE', and 'ME'. 'CSE' is checked.
- Submit:** A button labeled 'Submit'.



## Program 26

Create a canvas having smiley face.

### **Source Code:**

```
import java.awt.*;
import javax.swing.*;

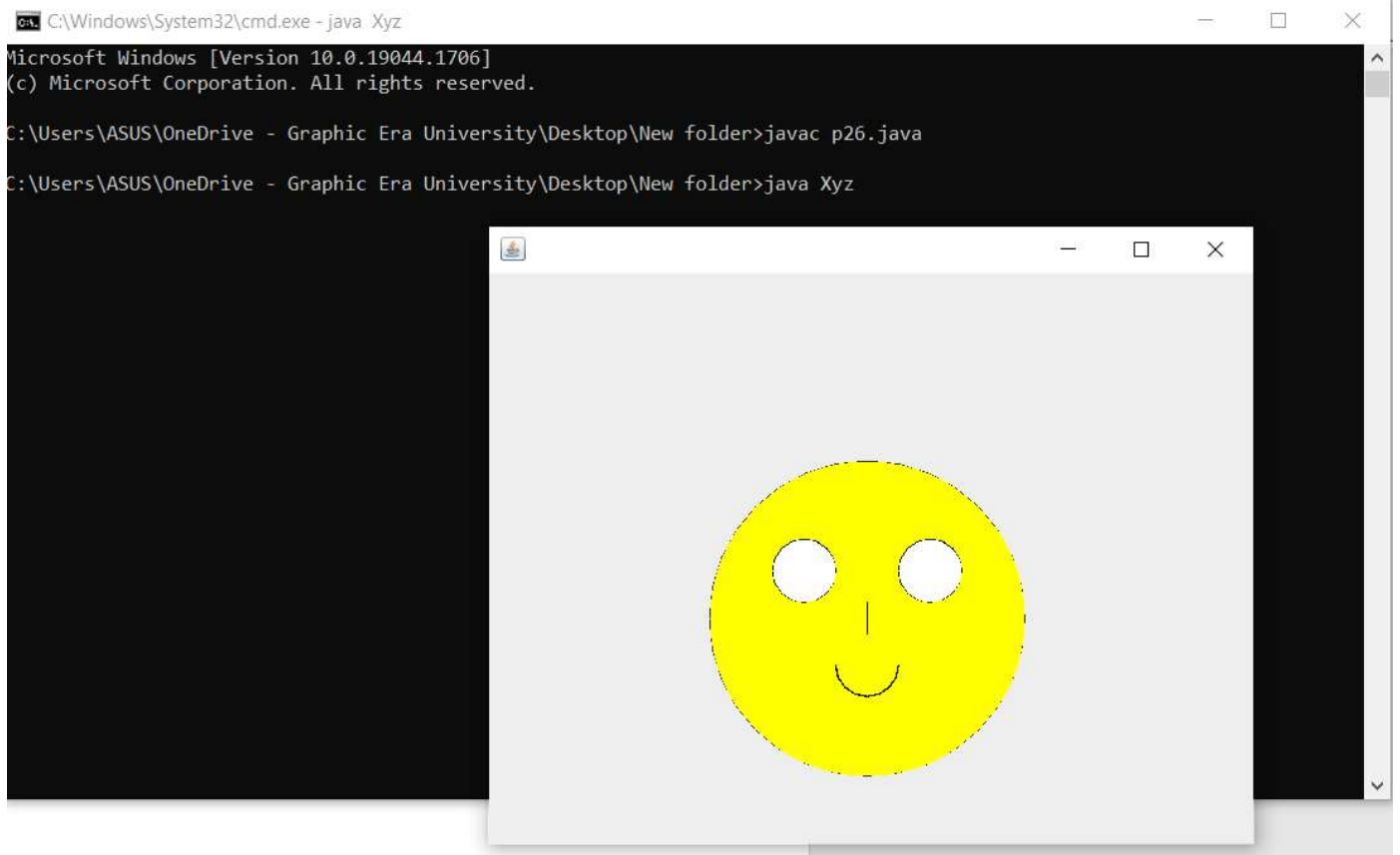
class Xyz extends Canvas
{
    public void paint(Graphics g)
    {
        g.drawOval(140,120,200,200);
        g.setColor(Color.yellow);
        g.fillOval(140,120,200,200);
        g.setColor(Color.black);

        g.drawOval(180,170,40,40);
        g.setColor(Color.white);
        g.fillOval(180,170,40,40);
        g.setColor(Color.black);

        g.drawOval(260,170,40,40);
        g.setColor(Color.white);
        g.fillOval(260,170,40,40);
        g.setColor(Color.black);

        g.drawLine(240,210,240,230);
        g.drawArc(220,230,40,40,0,-180);
    }
    public static void main(String args[])
    {
        Xyz d=new Xyz();
        JFrame f=new JFrame();
        f.setSize(500,400);
        f.setVisible(true);
        f.add(d);
    }
}
```

## OUTPUT



### Program 27

Write a JFrame having three textfields. The first two textfields refers to two numbers on which sum or subtraction will happen. The third textfield will show the result. There are two buttons “SUM” and “SUBTRACTION”. Once user will click on sum or subtraction buttons then the corresponding result will be displayed in result field.

#### Source Code:

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

class A extends JFrame implements ActionListener
{
    JLabel l1;
    JTextField t1;
    JLabel l2;
    JTextField t2;
    JButton b1;
    JButton b2;
    JLabel l3;
    JTextField t3;

    A()
    {
        setLayout(new FlowLayout());

        l1=new JLabel("Enter the first no.: ");
        t1=new JTextField(20);

        l2=new JLabel("Enter the second no.: ");
        t2=new JTextField(20);

        b1=new JButton("SUM");
        b2=new JButton("SUB");

        l3=new JLabel("Enter the result: ");
        t3=new JTextField(20);

        setLayout(new FlowLayout());

        add(l1);
```

```

        add(t1);
        add(l2);
        add(t2);
        add(b1);
        add(b2);
        add(l3);
        add(t3);

        b1.addActionListener(this);
        b2.addActionListener(this);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e)
    {
        String s=e.getActionCommand();
        int result=0;
        if(s.equals("SUM"))
        {
            result=Integer.parseInt(t1.getText())+Integer.parseInt(t2.getText());
        }
        if(s.equals("SUB"))
        {
            result=Integer.parseInt(t1.getText())-Integer.parseInt(t2.getText());
        }
        t3.setText(Integer.toString(result));
    }

    public static void main(String args[])
    {
        A d=new A();
        d.setSize(300,400);
        d.setVisible(true);
    }
}

```

# OUTPUT

```
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>javac p27.java  
C:\Users\ASUS\OneDrive - Graphic Era University\Desktop\New folder>java A
```

Enter the first no.:  
612

Enter the second no.:  
2009

SUM

SUB Enter the result:  
2621