****

**Term Work**

**On**

**Java Programming Language**

**(PCS 408)**

**Submitted to: Submitted by:**

Dr. Prateek Srivastava Nishant Kumar  
Associate Professor University Roll. No.: 2018527  
GEHU, D. Dun Class Roll No./Section: 37/A

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GRAPHIC ERA HILL UNIVERSITY, DEHRADUN**

A picture containing text, clipart

Description automatically generated

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Program No.** | **Program Name** | **Page No.** |
| 1 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |
| 16 |  |  |
| 17 |  |  |
| 18 |  |  |
| 19 |  |  |
| 20 |  |  |
| 21 |  |  |
| 22 |  |  |
| 23 |  |  |
| 24 |  |  |
| 25 |  |  |
| 26 |  |  |
| 27 |  |  |
| 28 |  |  |
| 29 |  |  |
| 30 |  |  |

A picture containing logo

Description automatically generated

**DEPARTMENT OF CSE**

|  |
| --- |
| Photograph  Passport Size |

**STUDENT LAB REPORT SHEET**

**Name of Student .................................................... Mob. No ......................................**

**Address Permanent .....................................................................................................**

**Father’s Name ........................... Occupation ...................... Mob. No .........................**

**Mother’s Name ........................... Occupation ...................... Mob. No .......................**

**Section ............ Branch ............ Semester ............ Class Roll No ............ Grade A B C**

**Local Address ................................... Email ............................................ Marks 5 3 1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Practical** | **D.O.P.** | **Date of Submission** | **Grade (Viva)** | **Grade (Report File)** | **Total Marks (out of 10)** | **Student’s Signature** | **Teacher’s Signature** |
| **1** |  |  |  |  |  |  |  |  |
| **2** |  |  |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |  |  |
| **5** |  |  |  |  |  |  |  |  |
| **6** |  |  |  |  |  |  |  |  |
| **7** |  |  |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  |  |
| **9** |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |
| **11** |  |  |  |  |  |  |  |  |
| **12** |  |  |  |  |  |  |  |  |
| **13** |  |  |  |  |  |  |  |  |
| **14** |  |  |  |  |  |  |  |  |
| **15** |  |  |  |  |  |  |  |  |
| **16** |  |  |  |  |  |  |  |  |
| **17** |  |  |  |  |  |  |  |  |
| **19** |  |  |  |  |  |  |  |  |
| **20** |  |  |  |  |  |  |  |  |
| **21** |  |  |  |  |  |  |  |  |
| **22** |  |  |  |  |  |  |  |  |
| **23** |  |  |  |  |  |  |  |  |
| **24** |  |  |  |  |  |  |  |  |
| **25** |  |  |  |  |  |  |  |  |
| **26** |  |  |  |  |  |  |  |  |
| **27** |  |  |  |  |  |  |  |  |
| **28** |  |  |  |  |  |  |  |  |
| **29** |  |  |  |  |  |  |  |  |
| **30** |  |  |  |  |  |  |  |  |

**PRACTICAL 1**

**Question:**

Create a class “Student” having following instance variables and methods. Instance variables: ID, Name, Branch and university Method: setDetails() and showDetails(). The setDetails() method sets the values of ID, Name, Branch and University. And showDetails() method shows the value of each field.

**Source Code:**

import java.util.Scanner;

public class Student {

String name;

int id;

String branch;

String university;

public void getDetails (String stName, int stId, String stBranch, String stUniversity) {

name = stName;

id = stId;

branch = stBranch;

university = stUniversity;

}

public void showDetails (String stName, int stId, String stBranch, String stUniversity) {

System.out.println("Name : " +stName);

System.out.println("Student Id. : " +stId);

System.out.println("Branch Name : " +stBranch);

System.out.println("University Name : " +stUniversity);

}

public static void main (String[] args) {

Scanner Sc = new Scanner(System.in);

Student St = new Student();

System.out.print("Enter the Student Name : ");

String stName = Sc.nextLine();

System.out.print("Enter the Student Id : ");

int stId = Sc.nextInt();

Sc.nextLine();

System.out.print("Enter the Student Branch : ");

String stBranch = Sc.nextLine();

System.out.print("Enter the University Name: ");

String stUniversity = Sc.nextLine();

St.getDetails(stName, stId, stBranch, stUniversity);

St.showDetails(stName, stId, stBranch, stUniversity);

}

}

**OUTPUT:**

Text

Description automatically generated

**PRACTICAL- 2**

**Question:**

Write a Java Program to demonstrate the working of a banking-system

Instance variables: name, account\_no, amount

Instance methods: deposit(), withdraw(), checkBalance(), insert() and display().

Here we can deposit and withdraw amount from our account using deposit() and withdraw() methods respectively. The insert() method is to initialize state and display() method is to display state values.

**Source Code:**

import java.util.Scanner;

public class BankSystem {

String bankName, accountNo;

double amount;

public void insert(String name, String accountNumber, double balance) {

bankName = name;

accountNo = accountNumber;

amount = balance;

}

public double deposit(double depositAmount) {

amount = amount + depositAmount;

return amount;

}

public void withdraw(double withdrawalAmount) {

amount = amount - withdrawalAmount;

}

public void display() {

System.out.println();

System.out.println("Bank Name : " +bankName);

System.out.println("Account Number : " +accountNo);

System.out.println("Current Amount : " +amount);

}

public static void main(String[] args) {

Scanner Sc = new Scanner(System.in);

BankSystem Bs = new BankSystem();

System.out.print("Bank Name : ");

String name = Sc.nextLine();

System.out.print("Account Number : ");

String accountNumber = Sc.nextLine();

System.out.print("Original Amount : ");

double balance = Sc.nextDouble();

Bs.insert(name, accountNumber, balance);

Bs.display();

System.out.println();

System.out.print("Enter the amount for Deposit : ");

double depositAmount = Sc.nextDouble();

double ans = Bs.deposit(depositAmount);

Bs.display();

System.out.println();

System.out.print("Enter the amount for withdrawal : ");

double withdrawalAmount = Sc.nextDouble();

if (withdrawalAmount < ans) {

System.out.println("Money Available for withdrawal");

Bs.withdraw(withdrawalAmount);

Bs.display();

System.out.println();

}

else {

System.out.println("Insufficent Balance...");

}

}

}

Text

Description automatically generated

**PRACTICAL 3**

**Question:**

Write a program to sum two numbers. Here inputs are provided through command line argument.

**Source Code:**

public class sumCommand {

public static void main(String[] args) {

int a = Integer.parseInt(args[0]);

int b = Integer.parseInt(args[1]);

int sum = a + b;

System.out.println("Sum of Two number by Command Line : " +sum);

}

}

**OUTPUT:**

Text

Description automatically generated

**PRACTICAL 4**

**Question:**

Create class Employee with following attributes and methods ID, name, department, and salary. The setDetails() method sets the values of ID, name, department and salary. And showDetails() method shows the value of each field.

Note: (i) Values must be entered through Scanner class.

(ii) Use proper constructor

(iii) Use “this” reference variable to avoid ambiguity.

**Source Code:**

import java.util.Scanner;

public class Employee {

int id, salary;

String name, department;

public void getDetails(int Id, String employeeName, String dep, int sal) {

id = Id;

name = employeeName;

department = dep;

salary = sal;

}

public void showDetails() {

System.out.println();

System.out.println("Id : " +id);

System.out.println("Name : " +name);

System.out.println("Department : " +department);

System.out.println("Salary : " +salary);

}

public static void main(String[] args) {

Scanner Sc = new Scanner(System.in);

Employee Em = new Employee();

System.out.print("Enter the Employee Id : ");

int Id = Sc.nextInt();

Sc.nextLine();

System.out.print("Enter the Employee name : ");

String employeeName = Sc.nextLine();

System.out.print("Enter the Employee Department : ");

String dep = Sc.nextLine();

System.out.print("Enter the Employee Salary : ");

int sal = Sc.nextInt();

Em.getDetails(Id, employeeName, dep, sal);

Em.showDetails();

}

}

**OUTPUT:**

Text

Description automatically generated

**PRACTICAL 5**

**Question:**

Re-write program 1 with better memory management approach.

**Source Code:**

import java.util.Scanner;

public class StudentInfo {

String name;

int id;

String branch;

static String university = "GEHU";

public void getDetails (String name, int id, String branch) {

this.name = name;

this.id = id;

this.branch = branch;

}

public void showDetails (String name, int id, String branch, String university) {

System.out.println();

System.out.println("Name : " +name);

System.out.println("Student Id. : " +id);

System.out.println("Branch Name : " +branch);

System.out.println("University Name : " +university);

}

public static void main (String[] args) {

Scanner Sc = new Scanner(System.in);

StudentInfo St = new StudentInfo();

System.out.print("Enter the Student Name : ");

String name = Sc.nextLine();

System.out.print("Enter the Student Id : ");

int id = Sc.nextInt();

Sc.nextLine();

System.out.print("Enter the Student Branch : ");

String branch = Sc.nextLine();

St.getDetails(name, id, branch);

St.showDetails(name, id, branch, university);

}

}

**OUTPUT:**

Text

Description automatically generated

**PRACTICAL 6**

**Question:**

Apply following functions on the String "Java".

(i) Try to concat "Welcome" and write down your observation.

(ii) Find character at index 1

(iii) Find index of first 'a'.

(iv) Find index of second 'a'

(v) Compare "Java" to "JAVA"

(vi) Compare "Java" to "JAVA" ignoring the case

(vii) Find the index of first 'a' from last

**Source Code:**

public class JavaFun {

public static void main(String[] args) {

String s = "Java";

System.out.println("String after the Concatentation : " +s.concat("Welcome "));

System.out.println("Character at First Index : " +s.charAt(1));

System.out.println("Index of First 'a' : " +s.indexOf('a'));

System.out.println("Index of Second 'a' : " +s.indexOf('a', s.indexOf('a') + 1));

System.out.println("Compare 'Java' to 'JAVA' : " +s.compareTo("JAVA"));

System.out.println("Compare 'Java' to 'JAVA' ignoring Case : " +s.equalsIgnoreCase("JAVA"));

System.out.println("Index of First 'a' from last : " +s.lastIndexOf('a'));

}

}

**OUTPUT:**

Text

Description automatically generated

**PRACTICAL 7**

**Question:**

Apply following functions on StringBuffer object "HELLO"

(i) Append "Java"

(ii) Insert "Java" at index 1

(iii) Replace with "Java" with characters between index 1 to 2

(iv) Delete characters between index 1 and 2

(v) Reverse the string "HELLO"

**Source Code:**

class Append {

public void app() {

StringBuffer Sb = new StringBuffer("Hello");

Sb.append("Java"); System.out.println(Sb);

}

} class Insert {

public void ins() {

StringBuffer Sb = new StringBuffer("Hello");

Sb.insert(1, "JAVA"); System.out.println(Sb);

}

} class Replace {

public void rep() {

StringBuffer Sb = new StringBuffer("Hello");

Sb.replace(1, 3, "JAVA"); System.out.println(Sb);

}

} class Delete {

public void del() {

StringBuffer Sb = new StringBuffer("Hello");

Sb.delete(1, 3); System.out.println(Sb);

}

} class Reverse {

public void rev() {

StringBuffer Sb = new StringBuffer("Hello");

Sb.reverse(); System.out.println(Sb);

}

}

public class Main {

public static void main (String [] args) {

Append Ap = new Append (); Ap.app();

Insert In = new Insert (); In.ins();

Replace Rp = new Replace (); Rp.rep();

Delete De = new Delete (); De.del();

Reverse Re = new Reverse (); Re.rev();

}

}

Text

Description automatically generated

**PRACTICAL - 8**

**Question:**

Create a class “Student” having following instance variables and methods.

Instance variables: ID, Name, Branch, city and university While creating constructors with one, two, three, four and five arguments reuse the constructors by construction chaining.

**Source Code:**

public class ConstructorChain {

int ID;

String name, city, branch, university;

ConstructorChain(int Id) {

ID = Id;

}

ConstructorChain(int Id, String nm) {

this(Id);

name = nm;

}

ConstructorChain(int Id, String nm, String bran) {

this(Id, nm);

branch = bran;

}

ConstructorChain(int Id, String nm, String bran, String City) {

this(Id, nm, bran);

city = City;

}

ConstructorChain(int Id, String nm, String bran, String City, String University) {

this(Id, nm, bran, City);

university = University;

}

public void Display() {

System.out.println(“ID : “ +ID);

System.out.println(“Name : “ +name);

System.out.println(“Branch : “ +branch);

System.out.println(“City : “ +city);

System.out.println(“University : “ +university);

}

public static void main(String[] args) {

ConstructorChain ConC = new ConstructorChain(20011940, “Nishan Kumar”, “CSE”, “Dehradun”, “GEHU”);

ConC.Display();

}

}

**OUTPUT:**

Text

Description automatically generated

**PRACTICAL - 9**

**Question:**

Create two-dimensional integer array and insert, search, and traverse this array.

Note: Use Scanner class to insert data.

**Source Code:**

import java.util.Scanner;

public class Arrays {

public static void main(String[] args) {

Scanner Sc = new Scanner(System.in);

System.out.print(“Enter the Number of Rows : “);

int row = Sc.nextInt();

System.out.print(“Enter the number of Column : “);

int column = Sc.nextInt();

int arr[][] = new int[row][column];

for (int I = 0; I < row; ++i) {

for (int j = 0; j < column; ++j) {

System.out.print(“Enter the element : “);

int val = Sc.nextInt();

arr[i][j] = val;

}

}

System.out.println(“After Enter the element in Array : “);

for (int I = 0; I < row; ++i) {

for (int j = 0; j < column; ++j) {

System.out.print(arr[i][j] +” “);

}

System.out.println(“ “);

}

System.out.print(“Enter the element to be Search : “);

int num = Sc.nextInt();

for (int I = 0; I < row; ++i) {

for (int j = 0; j < column; ++j) {

if (arr[i][j] == num) {

System.out.println(“Element Found : “ +num);

break;

}

}

}

}

}

**OUTPUT:**

Text

Description automatically generated

**PRACTICAL - 10**

**Question:**

Create a jagged array having three rows. Where 1st row contains 3 columns, 2nd row contains 4 columns and 3rd row contains 2 columns. Insert and traverse it.

**Source Code:**

import java.util.Scanner;

public class Jagged {

public static void main(String[] args) {

Scanner Sc = new Scanner(System.in);

int arr[][] = new int[3][];

arr[0] = new int[3];

arr[1] = new int[5];

arr[2] = new int[6];

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

for (int j = 0; j < arr[i].length; ++j) {

System.out.print("Enter the Value : ");

int val = Sc.nextInt();

arr[i][j] = val;

}

}

System.out.println("Jagged Array : ");

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

for (int j = 0; j < arr[i].length; ++j) {

System.out.print(arr[i][j] + " ");

}

System.out.println(" ");

}

}

}

Text

Description automatically generated

**PRACTICAL – 11**

**Question:**

Create a class “Shape” having area () method to calculate area. Overload the area () method for shapes like triangle, rectangle, and circle.

**Source Code:**

import java.util.Scanner;

public class Shape {

int length, breadth, side;

double radius;

public void Area(int length, int breadth) {

this.length = length;

this.breadth = breadth;

int area = (this.length) \* (this.breadth);

System.out.println("The area of Rectangle : " +area);

System.out.println(" ");

}

public void Area(int side) {

this.side = side;

int area = (this.side) \* (this.side);

System.out.println("The area of Square : " +area);

System.out.println(" ");

}

public void Area(double radius) {

this.radius = radius;

double area = (3.14 \* (this.radius) \* (this.radius));

System.out.println("The area of Circle : " +area);

}

public static void main(String[] args) {

Scanner Sc = new Scanner(System.in);

Shape Sp = new Shape();

System.out.print("Enter the length of Rectangle : ");

int length = Sc.nextInt();

System.out.print("Enter the breadth of Rectangle : ");

int breadth = Sc.nextInt();

Sp.Area(length, breadth);

System.out.print("Enter the Side Of Square : ");

int side = Sc.nextInt();

Sp.Area(side);

System.out.print("Enter the radius of Circle : ");

double radius = Sc.nextDouble();

Sp.Area(radius);

}

}

Text

Description automatically generated

**PRACTICAL - 12**

**Question:**

Create a class “Bank” having method getRateOfInterest(). Create child classes as HDFC, SBI and PNB and override getRateOfInterest() and return interest rates as 4.0, 4.5 and 5% correspondingly. Use concept of Upcasting to implement this scenario.

**Source Code:**

class Bank {

double getRateOfInterest() {

double rateOfInterest = 2.0;

return rateOfInterest;

}

}

class Hdfc extends Bank {

double getRateOfInterest() {

double rateOfInterest = 4.0;

return rateOfInterest;

}

}

class Sbi extends Bank {

double getRateOfInterest() {

double rateOfInterest = 4.5;

return rateOfInterest;

}

}

class Pnb extends Bank {

double getRateOfInterest() {

double rateOfInterest = 5.0;

return rateOfInterest;

}

}

class Interests {

public static void main(String[] args) {

// Upcasting

Bank Bn = new Hdfc();

System.out.print("Rate Of Interest Of HDFC Bank : " +Bn.getRateOfInterest());

System.out.println();

Bn = new Sbi();

System.out.print("Rate Of Interest Of SBI Bank : " +Bn.getRateOfInterest());

System.out.println();

Bn = new Pnb();

System.out.print("Rate Of Interest Of PNB Bank : " +Bn.getRateOfInterest());

System.out.println();

}

}

Text

Description automatically generated

**Practical – 13**

**Question:**

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) {

Name- Nishant Kumar Roll Number: 2018861 Section-A V

} else {}

}

return(a/b);

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));

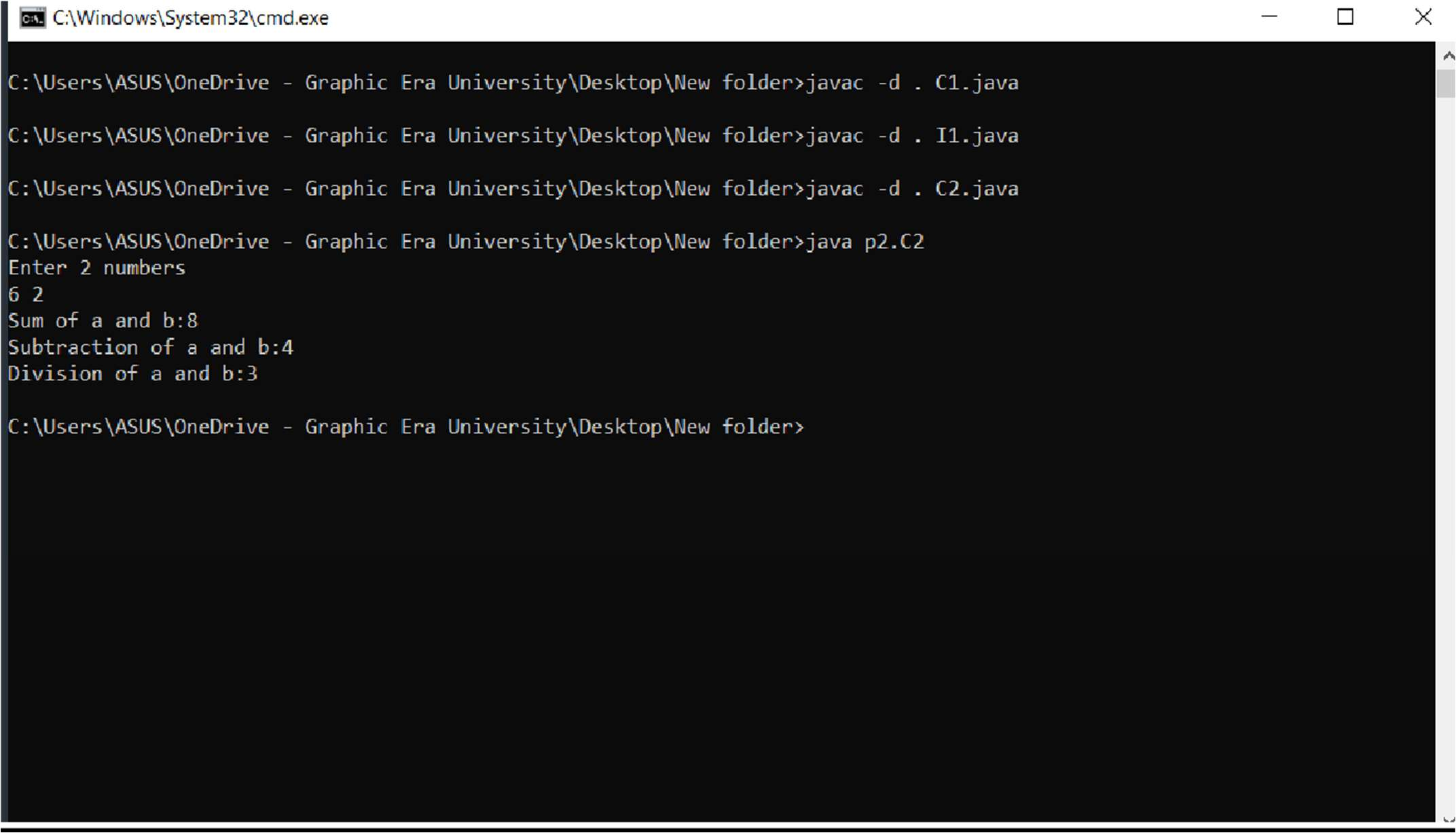
System.out.println("Subtraction of a and b:"+d.sub(a,b));

System.out.println("Division of a and b:"+f.div(a,b));

}

}

# OUTPUT



**Practical - 14**

**QUESTION:**

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)

{

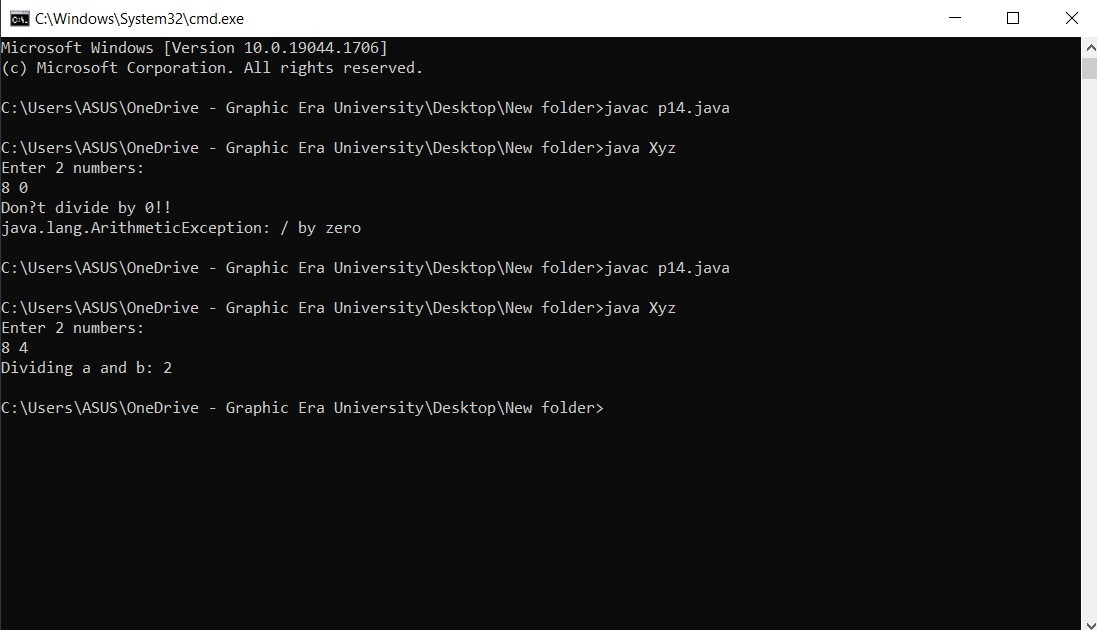
System.out.println("Don’t divide by 0!!\n"+e);

}

}

}

# OUTPUT



**Practical - 15**

**Question:**

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();

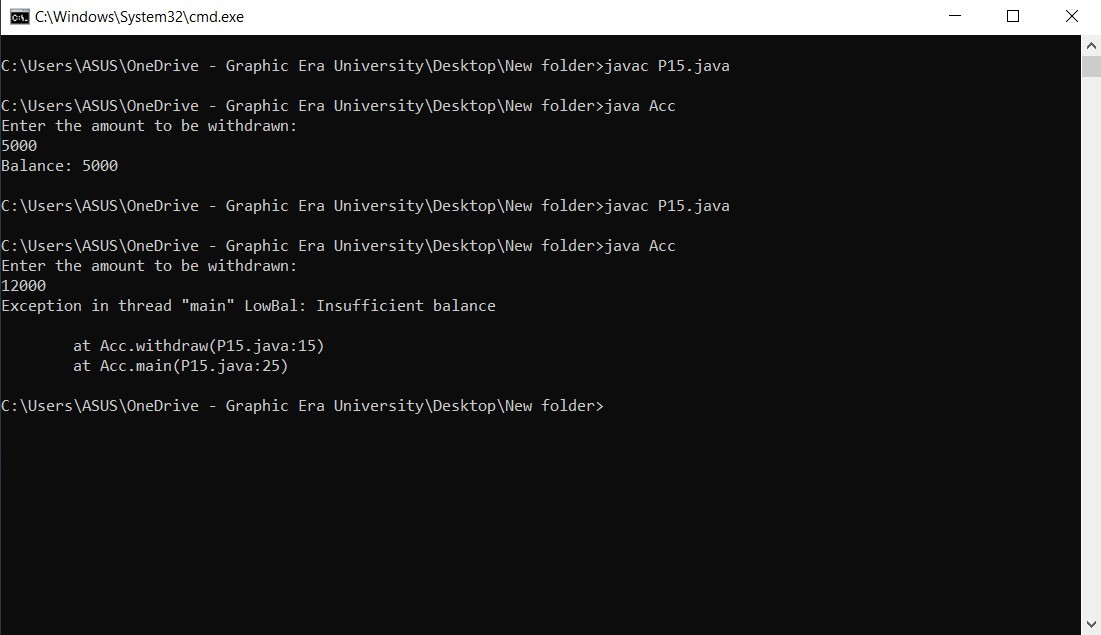
Scanner in= new Scanner(System.in); System.out.println("Enter the amount to be withdrawn: "); int amt=in.nextInt();

d.withdraw(amt); System.out.println("Balance: "+d.bal);

}

}

# OUTPUT



**Practical - 16**

**QUESTION:**

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();

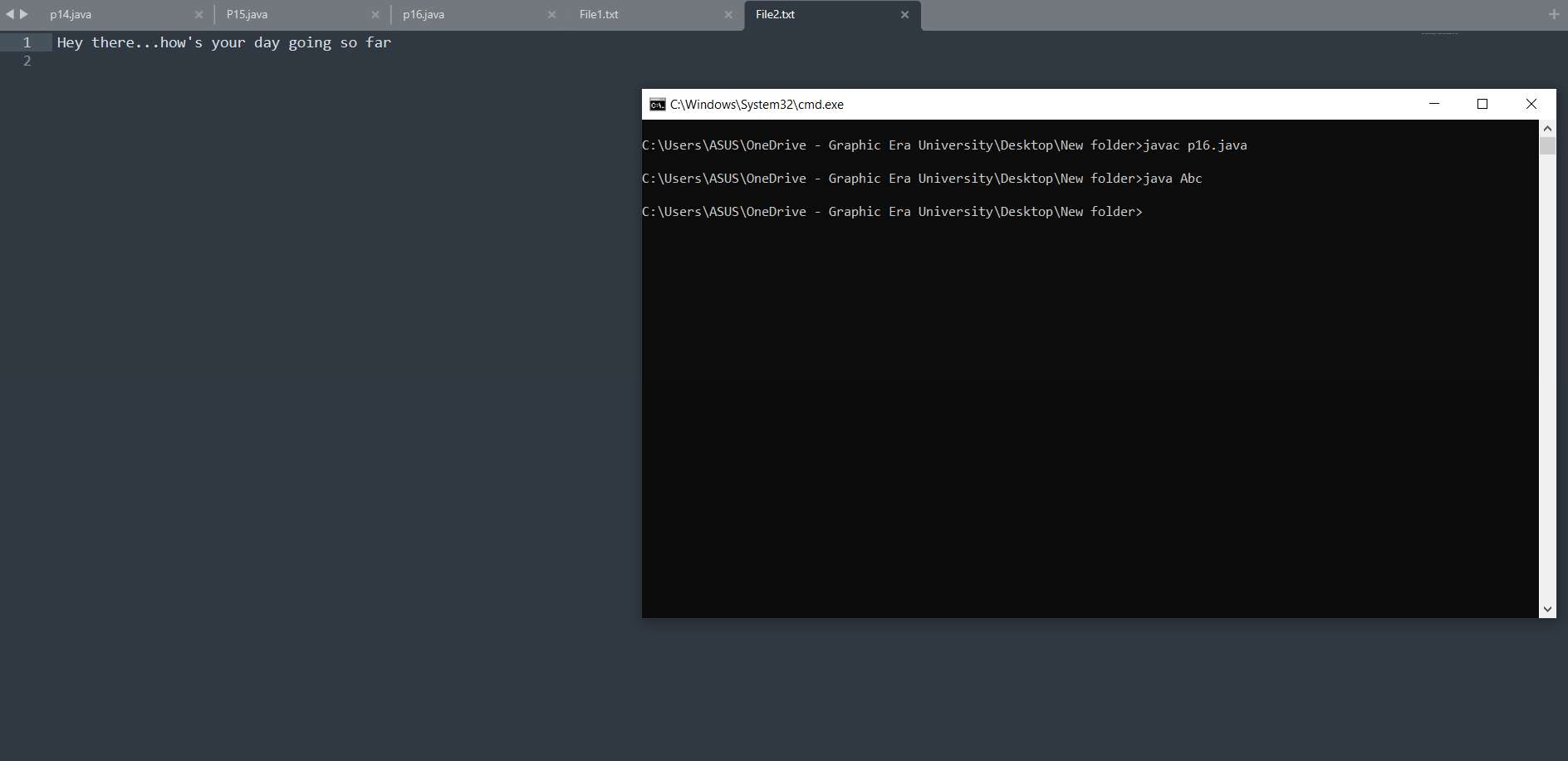
fos.close();

bos.close();

}

}

# OUTPUT



**Practical - 17**

**QUESTION:**

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();

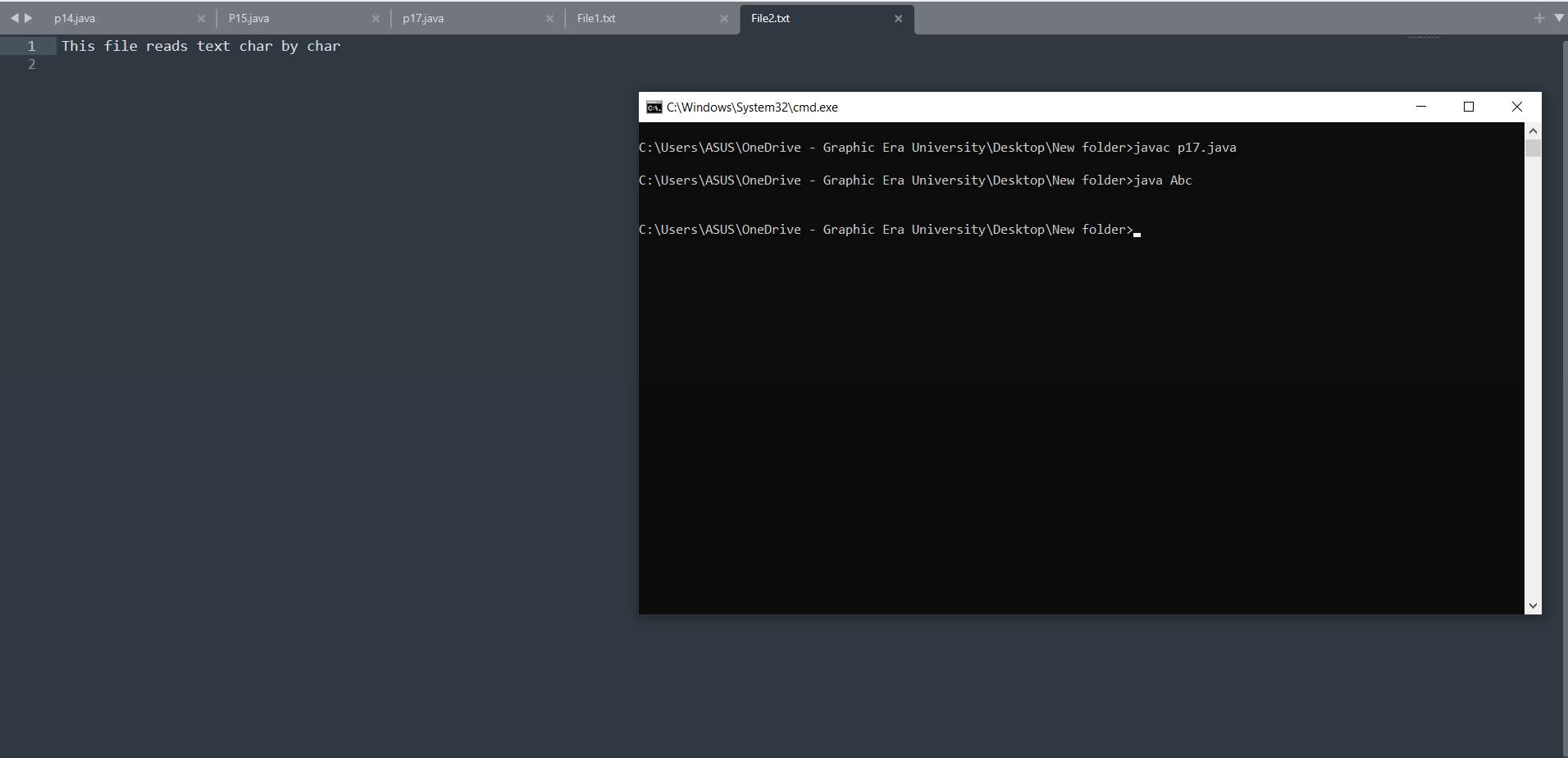
fw.close();

bw.close();

}

}

# OUTPUT



**Practical - 18**

**QUESTION:**

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);

}

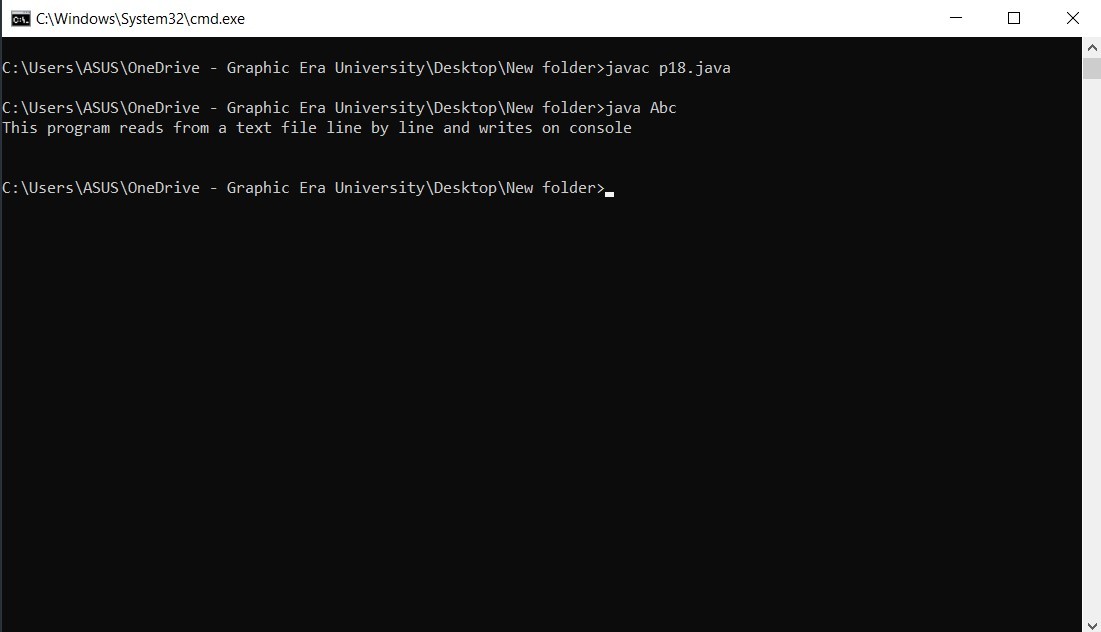
System.out.println(); fr.close();

br.close();

}

}

# OUTPUT



**Practical -19**

**QUESTION:**

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();

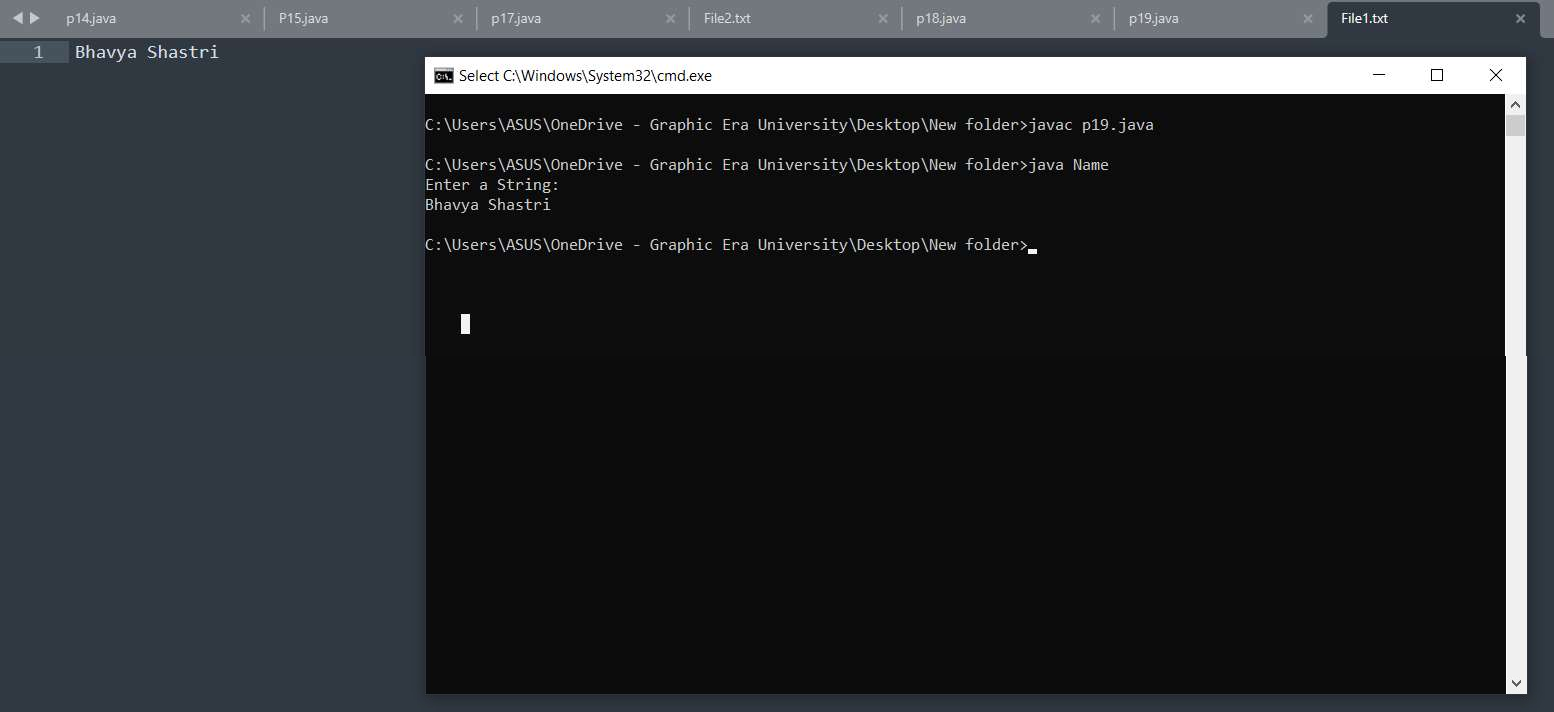
br.close();

isr.close();

}

}

# OUTPUT



**Practical - 20**

**QUESTION:**

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();

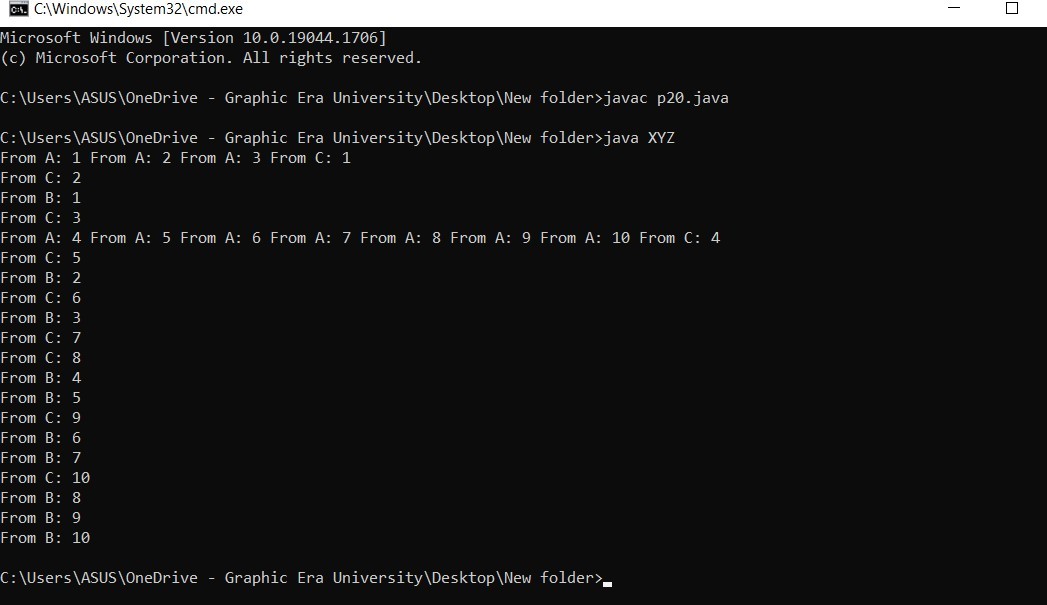
b.start();

c.start();

}

}

# OUTPUT



**Practical - 21**

**QUESTION:**

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("FromB: "+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());

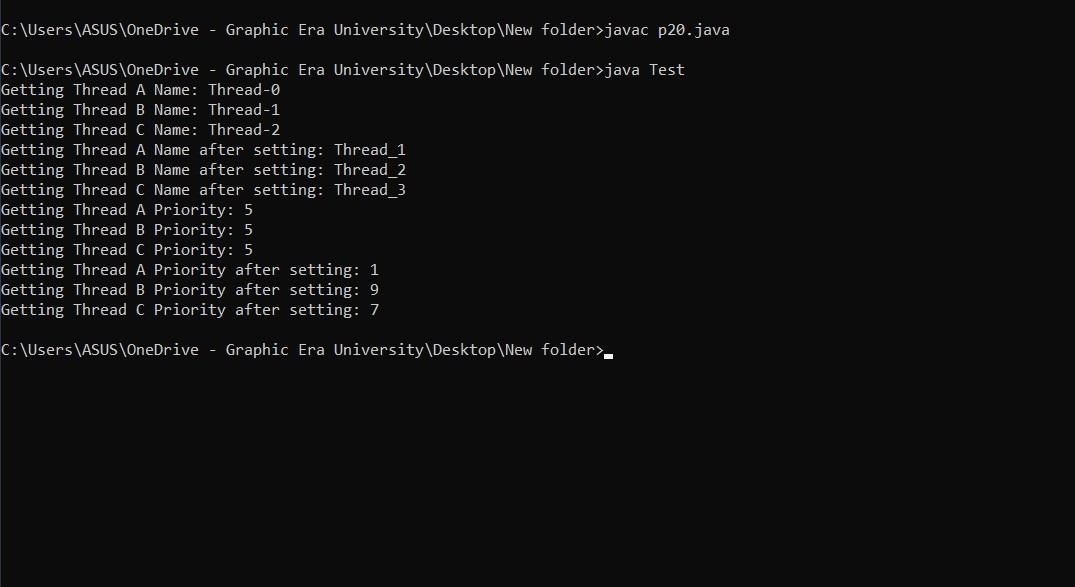
System.out.println("Getting Thread B Priority after setting: "+b.getPriority());

System.out.println("Getting Thread C Priority after setting: "+c.getPriority());

}

}

OUTPUT



**Practical – 22**

**QUESTION:**

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();

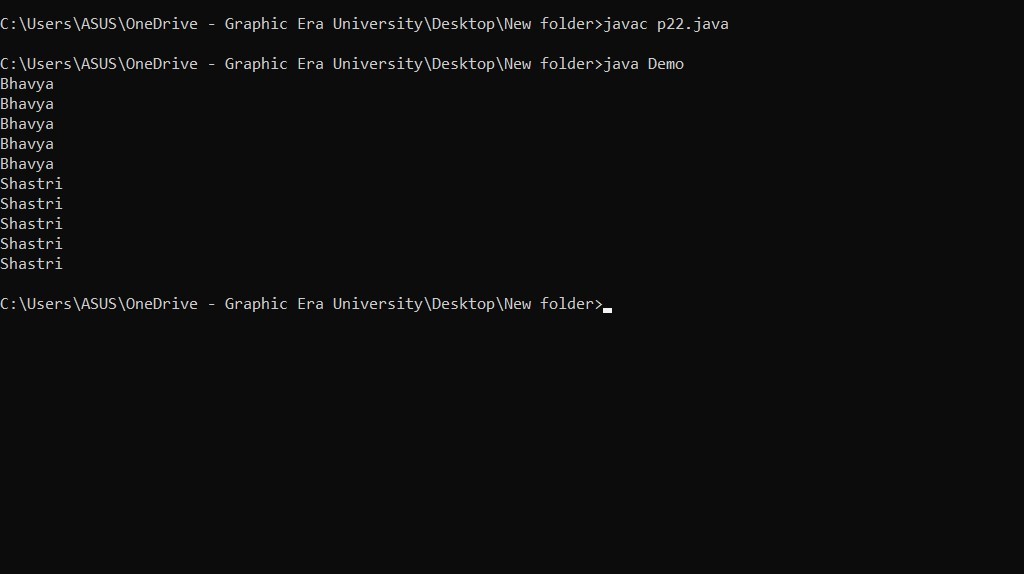
MyThread t1=new MyThread(d,"Bhavya"); t1.start();

MyThread t2=new MyThread(d,"Shastri"); t2.start();

}

}

# OUTPUT



**Practical - 23**

**QUESTION:**

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();

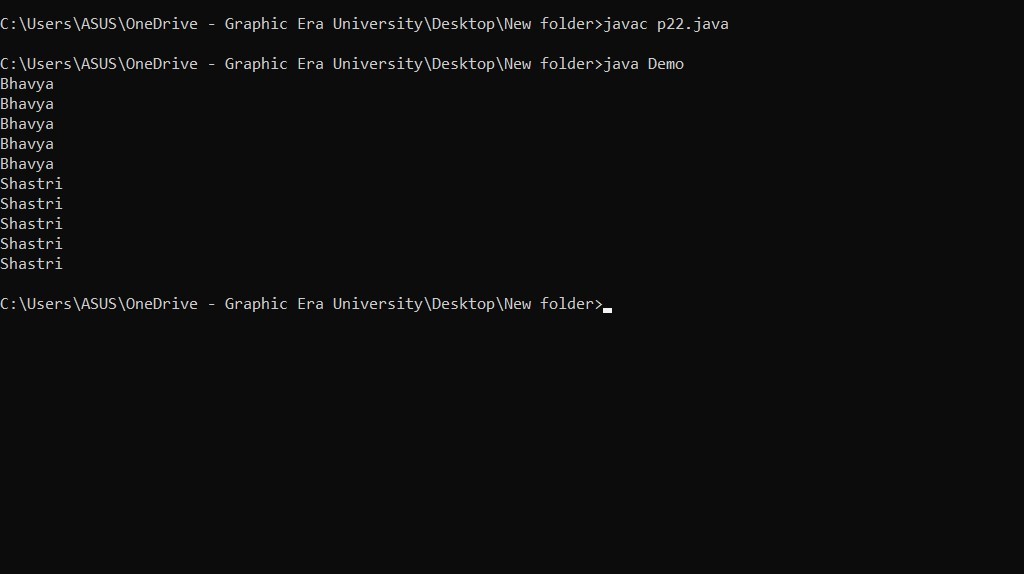
MyThread t1=new MyThread(d1,"Bhavya"); t1.start();

MyThread t2=new MyThread(d2,"Shastri"); t2.start();

}

}

# OUTPUT



**Practical - 24**

**QUESTION:**

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 withdrawl(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();

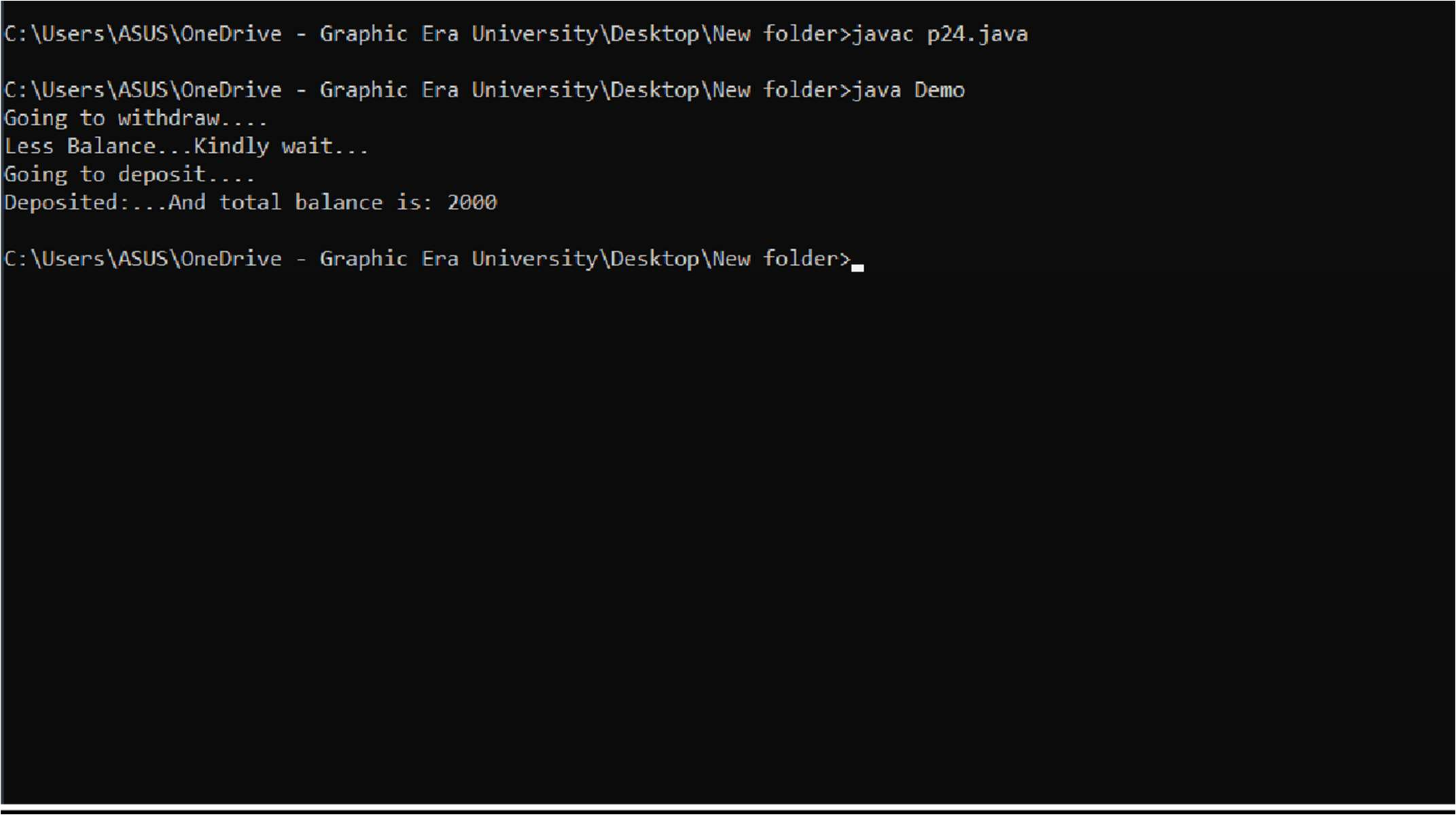
MyThread1 t1=new MyThread1(c); t1.start();

MyThread2 t2=new MyThread2(c); t2.start();

}

}

# OUTPUT



**Practical - 25**

**QUESTION:**

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=newJLabel("Name");

l2=newJLabel("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);

}

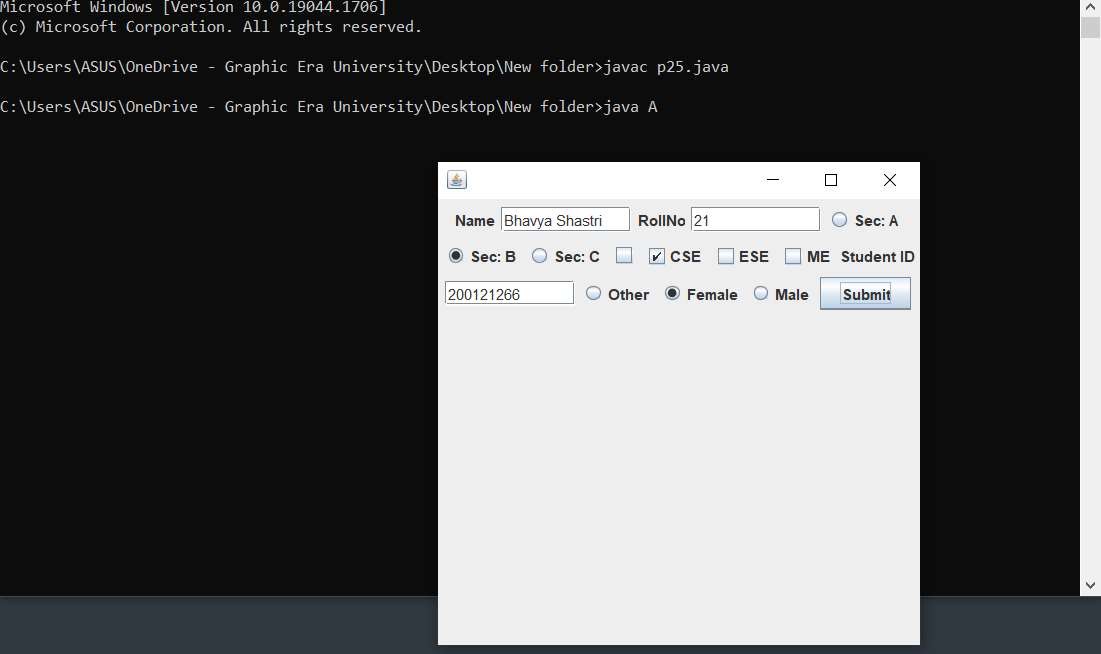
public static void main(String args[]) {

A d=new A(); d.setSize(400,400); d.setVisible(true);

}

}

OUTPUT



**Practical - 26**

**QUESTION:**

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[]) {

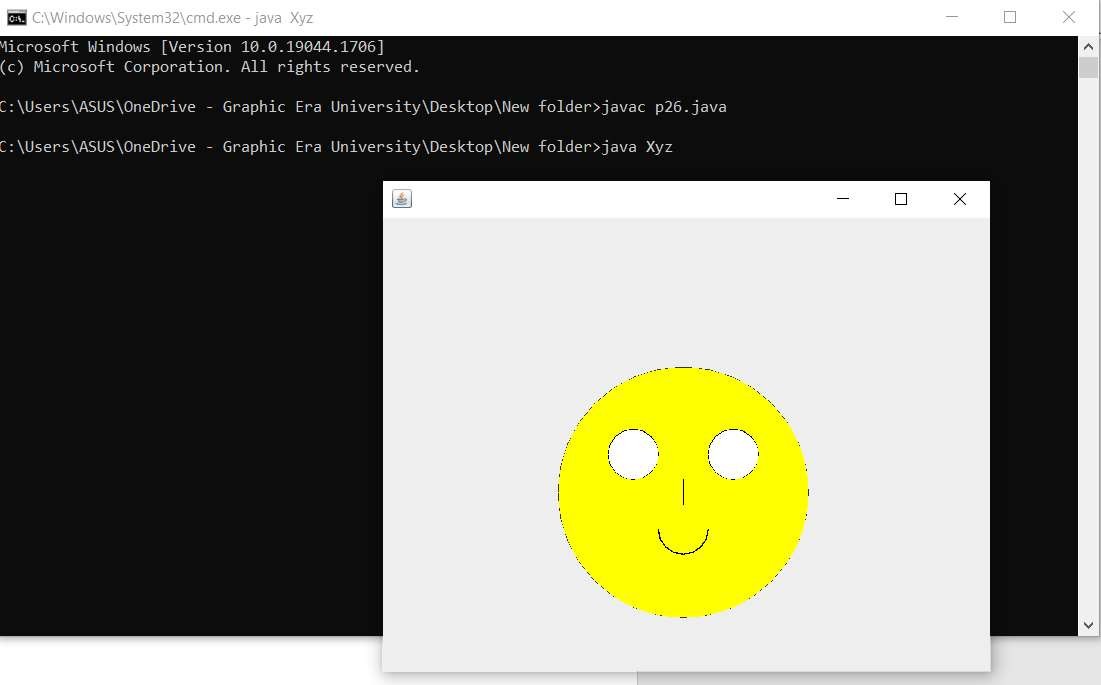
Xyz d=new Xyz(); JFrame f=new JFrame();

f.setSize(500,400); f.setVisible(true); f.add(d);

}

}

# OUTPUT



**Practical – 27**

**QUESTION:**

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

