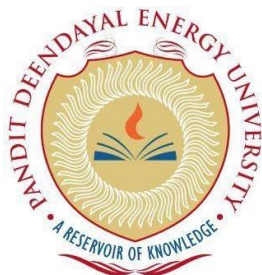


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF TECHNOLOGY

PANDIT DEENDAYAL ENERGY UNIVERSITY

SESSION 2024-25



SUBMITTED BY

NAME : VRAJ PATEL
ROLL NO. : 23BCP516D
DIVISION : 3
**COURSE NAME : Object Oriented Programming
with JAVA Lab**
COURSE CODE : 23CP201P

SUBMITTED TO

Dr. Nandini Modi

Assistant Professor

Department of Computer Science and Engineering

Pandit Deendayal Energy University

INDEX

Exp. No.	Title of Lab Work	Date	Page No.	Signature
1.	Set up and get familiar with Java programming environment.	22/7/2024	1	
2.	Study language features of Java (variables, data types, declarations, loop and branch constructs, etc.)	29/7/2024	7	
3.	Class and Objects: study and implement classes based application using Java	5/8/2024	16	
4.	Inheritance: study and implement various types of inheritance in Java.	12/8/2024	23	
5.	Polymorphism: study and implement various types of Polymorphism in java.	2/9/2024	28	
6.	Study and implement Abstract class and Interfaces in Java	9/9/2024	31	
7.	Study and implement Exception handling in Java	30/9/2024	38	
8.	Study and implement File Handling in Java	7/10/2024	44	
9.	Study and implement multi-threaded application in Java	14/10/2024	55	
10.	GUI programming using Java Applet/Swings Components and Event Handling	21/10/2024	63	

Practical – 1

Aim – Set up and get familiar with Java programming environment:

- i. Install JDK, setup Java environment and write a program to print
—CODING IS FUN, ENJOY IT!.
- ii. Write a Java program to print the sum of two numbers.

Steps to Install JDK and Setting up Environment Variable

Step-1 : Download JDK from the site

- Go to the Oracle site and open the Java SE download page. Under the latest version of Java Platform, Standard Edition, click on the JDK download button.

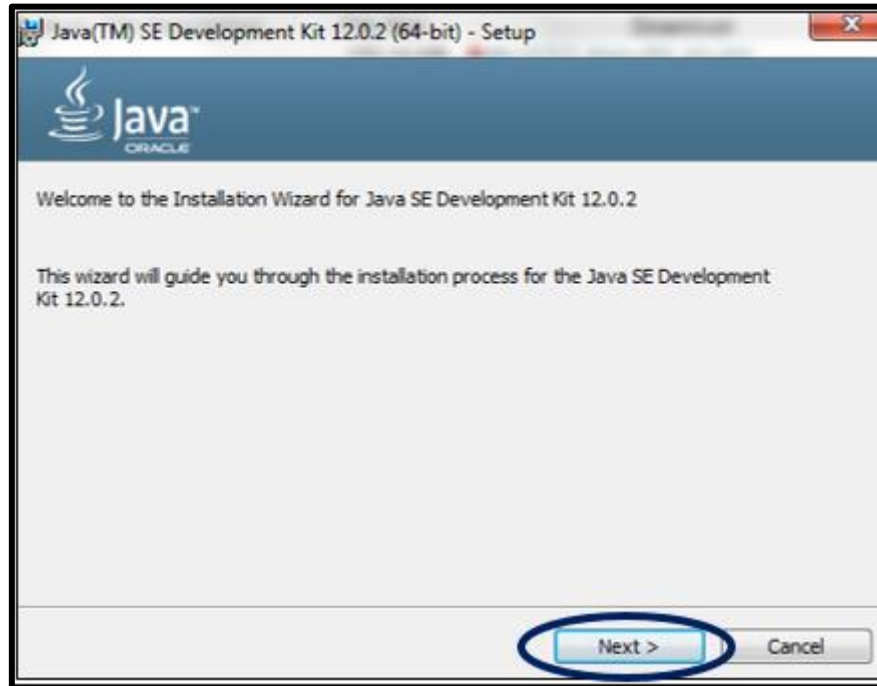


- Next, click on the Accept License Agreement button and choose your version of Java for Windows (32-bit or 64-bit) to proceed with downloading the JDK executable file.

Step-2 : Install the JDK exe File

- In this step, we will be running the executable JDK file (It will be a file with .exe as an extension) once the download is done. This installs JDK as well as JRE. For running this file on Windows, we will need Administrator rights.
- To begin the installation, we need to double-click on the downloaded file, and we will be presented with the below window.

- Click on Next to proceed with the installation, and follow the Installation guide provided for any queries.

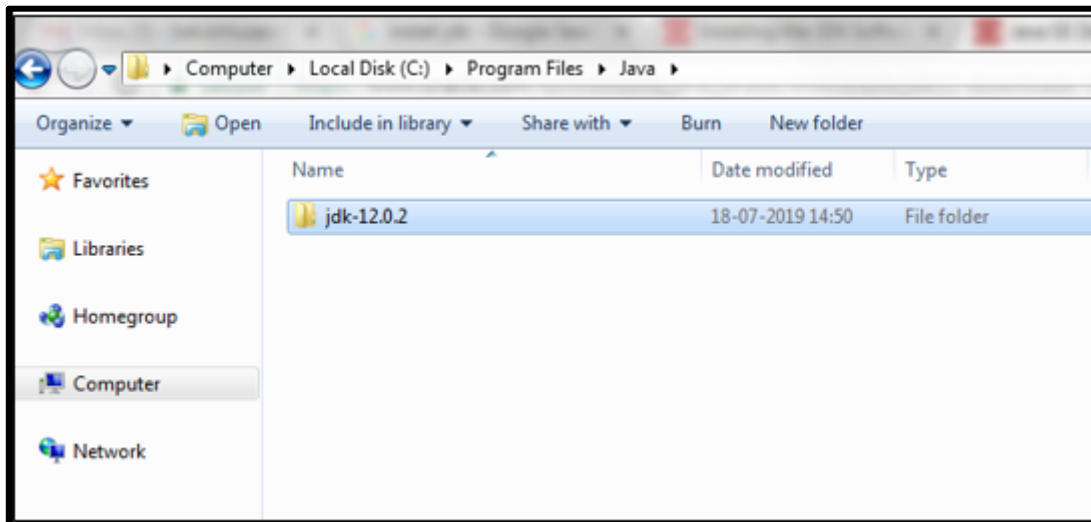


- Click on the Close button once the installation has finished.



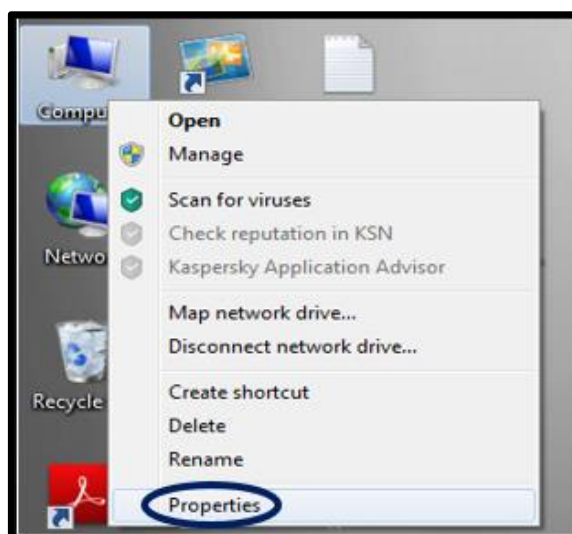
Step-3 : Check the Directory

- JDK gets installed in the C directory of our system by default having the path “C:\Program Files\Java\jdk-11.0”. If we make any change to this path at all, we need to make a note of it as it will be required in the upcoming steps.
- This is the directory structure for our example.

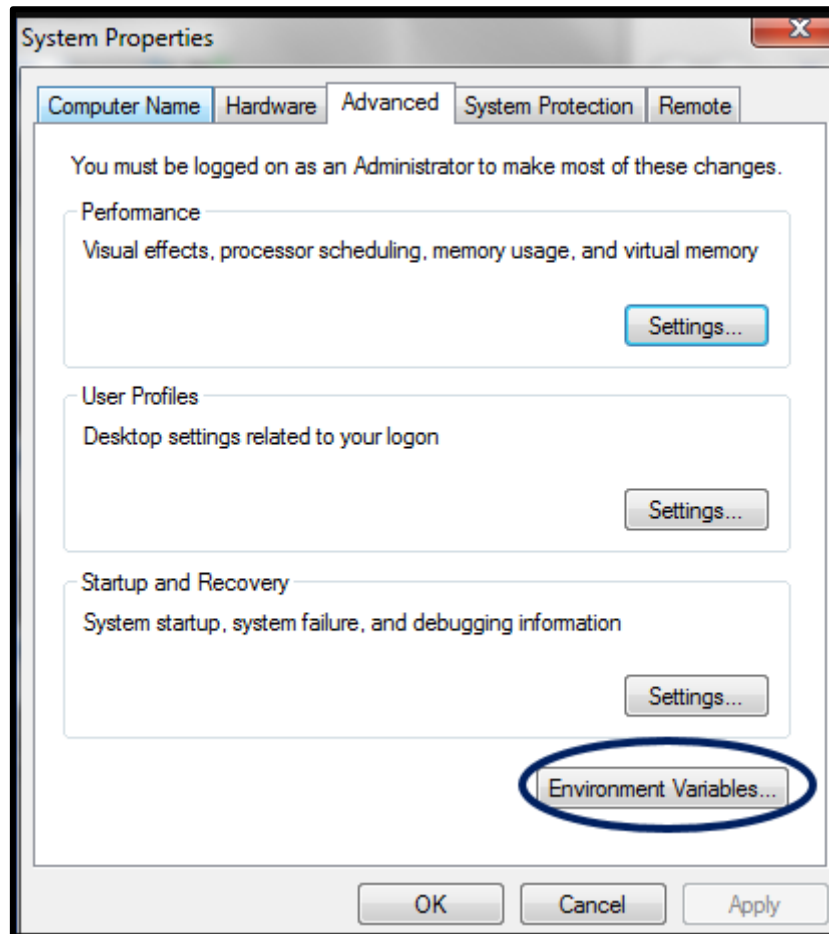


Step-4 : Update the Environment Variables

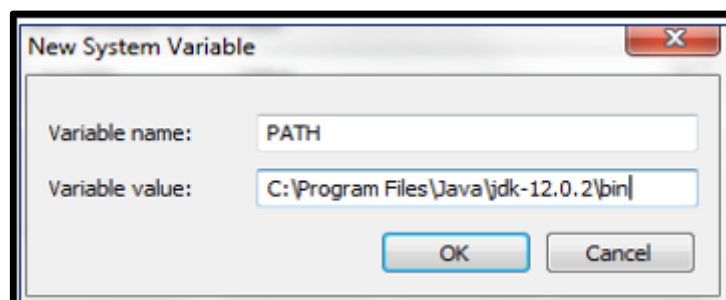
- We will need to update our system's Environment variables with our installed JDK bin path to run the Java programs because while executing the programs, the command prompt will look for the complete JDK bin path.
- The PATH variable in our system provides the exact location of executables that will be used for running Java programs, such as javac and java.
- If we do not set the PATH variable, we will specify the full path to the JDK bin every time we run a program.
- For example: C:\> "C:\Program Files\Java\jdk-11.0\bin\javac"
- So to set these variables, first right-click on My PC and select Properties.



- Inside Properties, in the left-side panel, select Advanced System Settings, and here choose the option Environment Variables.



- Click on New, and type PATH in the Variable Name, and enter the path of the bin of installed JDK in the Variable Value field.



- If we already have the PATH variable, we can edit it by adding it to the existing values.
- Click on the OK button to apply the changes.

Step-5 : Verify the Java Installation

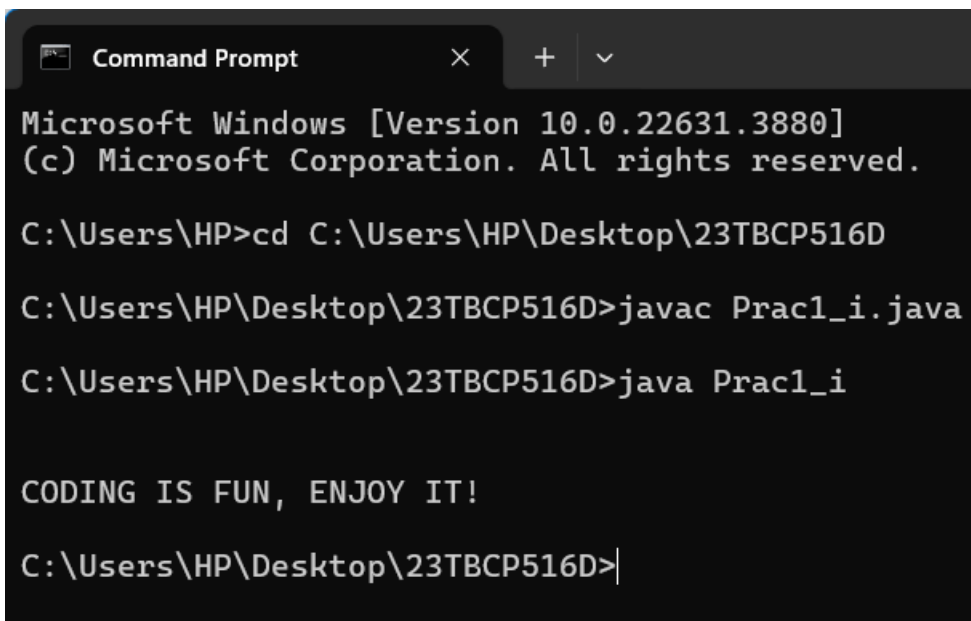
- Open the command prompt and enter the command “java –version”, and if it runs successfully, Java has been successfully installed.

Code –

// Practical No. -- 1 (i)
// Install JDK, setup Java environment and write a program to print - CODING IS FUN, ENJOY IT!.

```
class Prac1_i
{
    public static void main(String args[])
    {
        System.out.println("\n");
        System.out.println("CODING IS FUN, ENJOY IT!");
    }
}
```

Output –

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt" with standard window controls. The text inside shows the user navigating to a directory and running Java commands. The output of the program is displayed on the screen.

```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd C:\Users\HP\Desktop\23TBCP516D

C:\Users\HP\Desktop\23TBCP516D>javac Prac1_i.java

C:\Users\HP\Desktop\23TBCP516D>java Prac1_i

CODING IS FUN, ENJOY IT!

C:\Users\HP\Desktop\23TBCP516D>|
```

Code –

// Practical No. -- 1 (ii)

// Write a Java program to print the sum of two numbers.

```
import java.util.Scanner;
```

```
class Prac1_ii
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        int a,b,x;
```

```
        Scanner sc=new Scanner(System.in);
```

```
        System.out.println("\n");
```

```
        System.out.println("Enter Number 1 : ");
```

```
        a=sc.nextInt();
```

```
        System.out.println("\n");
```

```
        System.out.println("Enter Number 2 : ");
```

```
        b=sc.nextInt();
```

```
        x = a+b;
```

```
        System.out.println("\n");
```

```
        System.out.println("Sum of Number 1 and Number 2 : "+x);
```

```
    }
```

```
}
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac1_ii.java
```

```
C:\Users\HP\Desktop\23BCP516D>java Prac1_ii
```

```
Enter Number 1 :  
10
```

```
Enter Number 2 :  
20
```

```
Sum of Number 1 and Number 2 : 30
```

```
C:\Users\HP\Desktop\23BCP516D>|
```


Practical – 2

Aim – Study language features of Java (variables, data types, declarations, loop and branch constructs, etc.)

- i. You are developing a mathematical tool that requires generating a list of prime numbers. How would you implement a Java program to generate the first n prime numbers?
- ii. Write a program to enter two numbers and perform mathematical operations on them.
- iii. Write a program in Java to find maximum of three numbers using conditional operator.
- iv. You're working on a text analysis feature that counts the number of vowels and consonants in a given line of text. Write a program to accept a line and check how many consonants and vowels are there in line.
- v. Write an interactive program to print a string entered in a pyramid form. For instance, the string “stream” has to be displayed as follows:

```
      S
     S t
    S t r
   S t r e
  S t r e a m
```

- vi. Java Program to Find Largest Number in an array.
- vii. Write a java program to perform addition and multiplication of Two Matrices.

Code –

```
// Practical No. -- 2 (i)
// You are developing a mathematical tool that requires generating a list of prime numbers.
// How would you implement a Java program to generate the first n prime numbers?
```

```
import java.util.Scanner;
class Prac2_i
{
    public static void main(String arg[])
    {
        int i, count, n;
        System.out.println("\n");
        System.out.print("Enter a Number : ");
        Scanner sc=new Scanner(System.in);
```

```

        n=sc.nextInt();
        System.out.println("\n");
        System.out.println("Prime Numbers between 1 to "+n+" : ");
        System.out.println("\n");
        for(int j=2;j<=n;j++)
        {
            count=0;
            for(i=1;i<=j;i++)
            {
                if(j%i==0)
                {
                    count++;
                }
            }
            if(count==2)
                System.out.print(j+"\t");
        }
        System.out.println("\n");
    }
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac2_i.java
C:\Users\HP\Desktop\23BCP516D>java Prac2_i

Enter a Number : 10

Prime Numbers between 1 to 10 :

2      3      5      7

C:\Users\HP\Desktop\23BCP516D>

```

Code –

```
// Practical No. -- 2 (ii)  
// Write a program to enter two numbers and perform mathematical operations on them.
```

```
import java.util.Scanner;  
  
class Prac2_ii {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int a, b;  
        System.out.print("\nEnter Number 1 : ");  
        a = sc.nextInt();  
        System.out.print("Enter Number 2 : ");  
        b = sc.nextInt();  
  
        System.out.println("\nAddition : " + (a + b));  
        System.out.println("Subtraction : " + (a - b));  
        System.out.println("Multiplication : " + (a * b));  
        System.out.println("Division : " + (a / b));  
    }  
}
```

Output –

```
C:\Users\HP\Desktop\23TBCP516D>javac Prac2_ii.java  
C:\Users\HP\Desktop\23TBCP516D>java Prac2_ii  
  
Enter Number 1 : 20  
Enter Number 2 : 10  
  
Addition : 30  
Subtraction : 10  
Multiplication : 200  
Division : 2  
  
C:\Users\HP\Desktop\23TBCP516D>
```

Code –

// Practical No. -- 2 (iii)

// Write a program in to find maximum of three numbers using conditional operator.

```
import java.util.Scanner;
class Prac2_iii
{
    public static void main(String args[])
    {
        int i,a,b,c,x;

        Scanner sc=new Scanner(System.in);

        System.out.print("\nEnter Number 1 : ");
        a=sc.nextInt();

        System.out.print("\nEnter Number 2 : ");
        b=sc.nextInt();

        System.out.print("\nEnter Number 3 : ");
        c=sc.nextInt();

        x=((a>b)?((a>c)?a:c):(b>c)?b:c);

        System.out.print("\nMaximum Number : "+x);
        System.out.println("\n");
    }
}
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac2_iii.java
C:\Users\HP\Desktop\23BCP516D>java Prac2_iii

Enter Number 1 : 10

Enter Number 2 : 20

Enter Number 3 : 30

Maximum Number : 30

C:\Users\HP\Desktop\23BCP516D>|
```

Code –

// Practical No. -- 2 (iv)
// Write a program to accept a line and check how many consonants and vowels are there in line.

```
import java.util.Scanner;

public class Prac2_iv {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("\nEnter a line of text : ");
        String line = sc.nextLine();

        int vCount = 0;
        int cCount = 0;

        for (int i = 0; i < line.length(); i++) {
            char ch = Character.toLowerCase(line.charAt(i));

            if (ch >= 'a' && ch <= 'z') {
                if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                    vCount++;
                } else {
                    cCount++;
                }
            }
        }

        System.out.println("\nNumber of Vowels: " + vCount);
        System.out.println("Number of Consonants: " + cCount);
    }
}
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac2_iv.java
C:\Users\HP\Desktop\23BCP516D>java Prac2_iv
Enter a line of text : Hello, My name is Vraj Patel.
Number of Vowels: 8
Number of Consonants: 14
C:\Users\HP\Desktop\23BCP516D>|
```

Code –

// Practical No. -- 2 (v)
// Write an interactive program to print a string entered in a pyramid form.

```
import java.util.Scanner;

public class Prac2_v
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("\nEnter a String : ");
        String input = scanner.nextLine();
        int length = input.length();
        System.out.print("\n");
        for (int i = 0; i < length; i++)
        {
            for (int j = 0; j < length - i - 1; j++)
            {
                System.out.print(" ");
            }
            for (int k = 0; k <= i; k++)
            {
                System.out.print(input.charAt(k) + " ");
            }
            System.out.println();
        }
    }
}
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac2_v.java
C:\Users\HP\Desktop\23BCP516D>java Prac2_v

Enter a String : Stream

  S
 S t
S t r
S t r e
S t r e a
S t r e a m

C:\Users\HP\Desktop\23BCP516D>|
```

Code –

```
// Practical No. -- 2 (vi)
// Java Program to Find Largest Number in an array.

import java.util.Scanner;
public class Prac2_vi
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("\nEnter the number of elements in the array: ");
        int n = scanner.nextInt();
        int[] numbers = new int[n];
        System.out.println("\nEnter the elements of the array:");
        for (int i = 0; i < n; i++)
        {
            System.out.print("Element " + (i + 1) + ": ");
            numbers[i] = scanner.nextInt();
        }
        int max = numbers[0];
        for (int i = 1; i < numbers.length; i++)
        {
            if (numbers[i] > max)
            {
                max = numbers[i];
            }
        }
        System.out.println("\nThe largest number in the array is: " + max);
    }
}
```

Output –

```
C:\Users\HP\Desktop\23TBCP516D>javac Prac2_vi.java
C:\Users\HP\Desktop\23TBCP516D>java Prac2_vi

Enter the number of elements in the array: 6

Enter the elements of the array:
Element 1: 10
Element 2: 20
Element 3: 30
Element 4: 40
Element 5: 50
Element 6: 60

The largest number in the array is: 60

C:\Users\HP\Desktop\23TBCP516D>
```

Code –

// Practical No. -- 2 (vii)

// Write a java program to perform addition and multiplication of Two Matrices.

```
import java.util.Scanner;
public class Prac2_vii
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("\nEnter the number of Rows : ");
        int rows = scanner.nextInt();
        System.out.print("\nEnter the number of Columns : ");
        int columns = scanner.nextInt();
        int[][] matrix1 = new int[rows][columns];
        int[][] matrix2 = new int[rows][columns];
        int[][] sumMatrix = new int[rows][columns];
        int[][] productMatrix = new int[rows][columns];
        System.out.println("\nEnter the Elements of First Matrix : ");
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < columns; j++)
            {
                matrix1[i][j] = scanner.nextInt();
            }
        }
        System.out.println("\nEnter the Elements of Second Matrix : ");
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < columns; j++)
            {
                matrix2[i][j] = scanner.nextInt();
            }
        }
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < columns; j++)
            {
                sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < columns; j++)
            {
                productMatrix[i][j] = 0;
                for (int k = 0; k < columns; k++)
                {
                    productMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
                }
            }
        }
    }
}
```



```

    }
    System.out.println("\nSum of Matrices : ");
    printMatrix(sumMatrix);
    System.out.println("\nProduct of Matrices : ");
    printMatrix(productMatrix);
}
public static void printMatrix(int[][] matrix)
{
    for (int[] row : matrix)
    {
        for (int element : row)
        {
            System.out.print(element + "\t");
        }
        System.out.println();
    }
}
}

```

Output –

```

C:\Users\HP\Desktop\23TBCP516D>javac Prac2_vii.java
C:\Users\HP\Desktop\23TBCP516D>java Prac2_vii

Enter the number of Rows : 2

Enter the number of Columns : 2

Enter the Elements of First Matrix :
3
5
7
4

Enter the Elements of Second Matrix :
7
5
3
6

Sum of Matrices :
10      10
10      10

Product of Matrices :
36      45
61      59

C:\Users\HP\Desktop\23TBCP516D>

```

Practical – 3

Aim – Class and Objects: study and implement classes-based application using Java

- i. Write a program to create a “distance” class with methods where distance is computed in terms of feet and inches, how to create objects of a class.
- ii. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
- iii. Write a program that implements two constructors in the class. We call the other constructor using ‘this’ pointer, from the default constructor of the class.
- iv. Write a program in Java in which a subclass constructor invokes the constructor of the super class and instantiate the values.
- v. Write a program in Java to develop overloaded constructor. Also develop the copy constructor to create a new object with the state of the existing object.

Code –

```
// Practical No. -- 3 (i)
```

```
// Write a program to create a "distance" class with methods where distance is computed in terms of feet and inches.
```

```
import java.util.Scanner;
```

```
public class Distance  
{
```

```
    private float feet;  
    private float inch;
```

```
    static float inchToFeet(float n)  
    {  
        float feet = n / 12;  
        return feet;  
    }
```

```
    static float feetToInch(float n)  
    {  
        float inch = n * 12;
```

```

        return inch;
    }

    public void display()
    {
        System.out.println("\nDistance: " + feet + " feet " + inch + " inches");
    }

    public static void main(String[] args)
    {
        Distance distance1 = new Distance();
        distance1.display();
        Scanner sc = new Scanner(System.in);

        System.out.print("\nEnter Value in Inch: ");
        float inch = sc.nextFloat();

        System.out.print("Enter Value in Feet: ");
        float feet = sc.nextFloat();

        System.out.println("\nInch value in Feet: " + inchTofeet(inch));
        System.out.println("Feet value in Inch: " + feetToinch(feet));
    }
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Distance.java
C:\Users\HP\Desktop\23BCP516D>java Distance
Distance: 0.0 feet 0.0 inches

Enter Value in Inch: 100
Enter Value in Feet: 10

Inch value in Feet: 8.333333
Feet value in Inch: 120.0

C:\Users\HP\Desktop\23BCP516D>|

```

Code –

// Practical No. -- 3 (ii)

// Write a program to show the difference between public and private access specifiers.

```
public class Prac3_ii
{
    public static void main(String[] args)
    {
        // Public access demonstration
        Person person = new Person("Vraj", 18);
        System.out.println("Person name: " + person.getName());

        // Primitive data type (passed by value)
        int number = 10;
        System.out.println("Primitive before: " + number);
        modifyPrimitive(number);
        System.out.println("Primitive after: " + number);

        // Object (passed by reference)
        Person newPerson = new Person("Vidhi", 17);
        System.out.println("Object before: " + newPerson.getName());
        modifyObject(newPerson);
        System.out.println("Object after: " + newPerson.getName());

        // Final keyword usage
        final int constantValue = 100;
        System.out.println("Final constant: " + constantValue);
    }

    public static void modifyPrimitive(int value) {
        value = 20;
    }

    public static void modifyObject(Person person) {
        person.setName("Dharmil");
    }
}

class Person {
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
```

```

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

Output –

```

C:\Users\HP\Desktop\23TBCP516D>javac Prac3_ii.java

C:\Users\HP\Desktop\23TBCP516D>java Prac3_ii
Person name: Vraj
Primitive before: 10
Primitive after: 10
Object before: Vidhi
Object after: Dharmil
Final constant: 100

C:\Users\HP\Desktop\23TBCP516D>|

```

Code –

// Practical No. -- 3 (iii)

// Write a program that implements two constructors in the class. We call the other constructor using 'this' pointer, from the default constructor of the class.

```

public class Prac3_iii {
    int a;
    int b;

    // Default constructor
    public Prac3_iii() {
        // Calling the parameterized constructor using 'this'
        this(10, 30);
        System.out.println("Default constructor called.");
    }

    // Parameterized constructor
    public Prac3_iii(int a, int b) {
        this.a = a;
        this.b = b;
        System.out.println("Parameterized constructor called: a = " + a + ", b = " + b);
    }

    public static void main(String[] args) {

```

```

        // Creating object using default constructor
        Prac3_iii obj = new Prac3_iii();
        System.out.println("Object created. Values after default constructor: a = " + obj.a + ", b = " + obj.b);
    }
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac3_iii.java

C:\Users\HP\Desktop\23BCP516D>java Prac3_iii
Parameterized constructor called: a = 10, b = 30
Default constructor called.
Object created. Values after default constructor: a = 10, b = 30

C:\Users\HP\Desktop\23BCP516D>|

```

Code –

// Practical No. -- 3 (iv)
 // Write a program in Java in which a subclass constructor invokes the constructor of the super class and instantiate the values.

```

class Parent {
    String name;

    Parent(String name) {
        this.name = name;
        System.out.println("Parent class constructor called.");
    }

    void display() {
        System.out.println("Name in Parent class: " + name);
    }
}

class Child extends Parent {
    String name;

    Child(String parentName, String childName) {
        super(parentName); // Invoking the superclass constructor
        this.name = childName;
        System.out.println("Child class constructor called.");
    }

    @Override
    void display() {

```

```

        super.display(); // Calling the superclass display method
        System.out.println("Name in Child class: " + this.name);
    }
}

public class Prac3_iv {
    public static void main(String[] args) {
        System.out.println("Creating Child object...");
        Child obj = new Child("ParentName", "ChildName");
        System.out.println("Displaying values:");
        obj.display();
    }
}

```

Output –

```

C:\Users\HP\Desktop\23TBCP516D>javac Prac3_iv.java

C:\Users\HP\Desktop\23TBCP516D>java Prac3_iv
Creating Child object...
Parent class constructor called.
Child class constructor called.
Displaying values:
Name in Parent class: ParentName
Name in Child class: ChildName

C:\Users\HP\Desktop\23TBCP516D>

```

Code –

// Practical No. -- 3 (v)
 // Write a program in Java to develop overloaded constructor. Also develop the copy constructor to create a new object with the state of the existing object.

```

public class Prac3_v {
    private int number;

    // Default constructor
    public Prac3_v() {
        this.number = 20;
        System.out.println("Default constructor: number = 20");
    }

    // Parameterized constructor
    public Prac3_v(int number) {
        this.number = number;
        System.out.println("Parameterized constructor: number = " + number);
    }
}

```

```

// Copy constructor
public Prac3_v(Prac3_v obj) {
    this.number = obj.number;
    System.out.println("Copy constructor: number = " + obj.number);
}

public void displayNumber() {
    System.out.println("Number: " + number);
}

public static void main(String[] args) {
    Prac3_v obj1 = new Prac3_v(); // Default constructor
    Prac3_v obj2 = new Prac3_v(10); // Parameterized constructor
    Prac3_v obj3 = new Prac3_v(obj2); // Copy constructor

    System.out.println("\nDisplaying numbers:");
    obj1.displayNumber();
    obj2.displayNumber();
    obj3.displayNumber();
}
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac3_v.java

C:\Users\HP\Desktop\23BCP516D>java Prac3_v
Default constructor: number = 20
Parameterized constructor: number = 10
Copy constructor: number = 10

Displaying numbers:
Number: 20
Number: 10
Number: 10

C:\Users\HP\Desktop\23BCP516D>

```


Practical – 4

Aim – Inheritance: study and implement various types of inheritance in Java.

- i. Write a program in Java to demonstrate single inheritance, multilevel inheritance, and hierarchical inheritance.
- ii. Java Program to demonstrate the real scenario (e.g., bank) of Java Method Overriding where three classes are overriding the method of a parent class. Creating a parent class.
- iii. Write a java program for the use of super and this keyword.
- iv. Write a java program for the use of final keyword.

Code –

```
// Practical No. -- 4 (i)
// Write a program in Java to demonstrate single inheritance, multilevel inheritance, and
// hierarchical inheritance.
class Pr1_Parent {
    Pr1_Parent() {
        System.out.println("\nParent class constructor called!");
    }
}
class Child extends Pr1_Parent {
    Child() {
        System.out.println("Child class constructor called! - Single Inheritance");
    }
}
class Child2 extends Pr1_Parent {
    Child2() {
        System.out.println("Child2 class constructor called! - Hierarchical Inheritance");
    }
}
class Subchild extends Child {
    Subchild() {
        System.out.println("Subchild class constructor called! - Multilevel Inheritance");
    }
}
public class Prac4_i {
    public static void main(String[] args)
    {
        new Subchild();
    }
}
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac4_i.java
C:\Users\HP\Desktop\23BCP516D>java Prac4_i
Parent class constructor called!
Child class constructor called! - Single Inheritance
Subchild class constructor called! - Multilevel Inheritance
C:\Users\HP\Desktop\23BCP516D>|
```

Code –

// Practical No. -- 4 (ii)

// Java Program to demonstrate the real scenario (e.g., bank) of Java Method Overriding where three classes are overriding the method of a parent class. Creating a parent class.

```
class BankAccount {
    public void calculateInterest() {
        System.out.println("Generic interest calculation.");
    }
}

class SavingsAccount extends BankAccount {
    @Override
    public void calculateInterest() {
        System.out.println("Interest for Savings Account.");
    }
}

class CheckingAccount extends BankAccount {
    @Override
    public void calculateInterest() {
        System.out.println("Interest for Checking Account.");
    }
}

class BusinessAccount extends BankAccount {
    @Override
    public void calculateInterest() {
        System.out.println("Interest for Business Account.");
    }
}
```

```

public class Prac4_ii {
    public static void main(String[] args) {
        System.out.println("Calculating interests:");

        BankAccount savings = new SavingsAccount();
        BankAccount checking = new CheckingAccount();
        BankAccount business = new BusinessAccount();

        savings.calculateInterest();
        checking.calculateInterest();
        business.calculateInterest();
    }
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac4_ii.java

C:\Users\HP\Desktop\23BCP516D>java Prac4_ii
Calculating interests:
Interest for Savings Account.
Interest for Checking Account.
Interest for Business Account.

C:\Users\HP\Desktop\23BCP516D>

```

Code –

```

// Practical No. -- 4 (iii)
// Write a java program for the use of super and this keyword.

class Parent {
    int value;
    public Parent(int value) {
        this.value = value;
        System.out.println("Parent constructor called. Value set to: " + value);
    }
    public void show() {
        System.out.println("Parent value: " + value);
    }
}

class Child extends Parent {
    int value;
    public Child(int parentValue, int childValue) {
        // Calling the parent class constructor
    }
}

```

```

        super(parentValue);
        // Initializing the child class field
        this.value = childValue;
        System.out.println("Child constructor called. Parent value: " + parentValue + ", Child
value: " + childValue);
    }

    @Override
    public void show() {
        // Calling the parent class method
        super.show();
        System.out.println("Child value: " + value);
    }
}

public class Prac4_iii {
    public static void main(String[] args) {
        System.out.println("Creating Child object...");
        new Child(20, 50).show();
    }
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac4_iii.java

C:\Users\HP\Desktop\23BCP516D>java Prac4_iii
Creating Child object...
Parent constructor called. Value set to: 20
Child constructor called. Parent value: 20, Child value: 50
Parent value: 20
Child value: 50

C:\Users\HP\Desktop\23BCP516D>

```

Code –

```

// Practical No. -- 4 (iv)
// Write a java program for the use of final keyword.

```

```

final class FinalClass {
    final int value;

    public FinalClass(int value) {
        this.value = value;
    }
}

```

```
public final void display() {  
    System.out.println("The value stored in FinalClass is: " + value);  
}  
}  
  
public class Prac4_iv {  
    public static void main(String[] args) {  
        FinalClass obj = new FinalClass(200);  
        obj.display();  
    }  
}
```

Output –

```
C:\Users\HP\Desktop\23TBCP516D>javac Prac4_iv.java  
  
C:\Users\HP\Desktop\23TBCP516D>java Prac4_iv  
The value stored in FinalClass is: 200  
  
C:\Users\HP\Desktop\23TBCP516D>
```

Practical – 5

Aim – Polymorphism: study and implement various types of Polymorphism in java.

- i. Write a program that implements simple example of Runtime Polymorphism with multilevel inheritance. (e.g., Animal or Shape)
- ii. Write a program to compute if one string is a rotation of another. For example, pit is rotation of tip as pit has same character as tip.

Code –

```
// Practical No. -- 5 (i)
// Write a program that implements simple example of Runtime Polymorphism with multilevel inheritance.

// Base class for different animals.
class Animal {
    public void sound() {
        System.out.println("Some generic animal sound");
    }
}

// Derived class that extends Animal and overrides sound method.
class Cat extends Animal {
    @Override
    public void sound() {
        System.out.println("Meow");
    }
}

// Further derived class that extends Cat and overrides sound method.
class Dog extends Cat {
    @Override
    public void sound() {
        System.out.println("Bark");
    }
}

// Main class to demonstrate runtime polymorphism with multilevel inheritance.
public class Prac5_i {
    public static void main(String[] args) {
        Animal myAnimal = new Dog();
        System.out.print("Animal reference holding Dog object: ");
        myAnimal.sound(); // Calls Dog's sound method
    }
}
```

```

        Cat myCat = new Dog();
        System.out.print("Cat reference holding Dog object: ");
        myCat.sound(); // Calls Dog's sound method

        Dog myDog = new Dog();
        System.out.print("Dog reference holding Dog object: ");
        myDog.sound(); // Calls Dog's sound method
    }
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac5_i.java

C:\Users\HP\Desktop\23BCP516D>java Prac5_i
Animal reference holding Dog object: Bark
Cat reference holding Dog object: Bark
Dog reference holding Dog object: Bark

C:\Users\HP\Desktop\23BCP516D>

```

Code –

// Practical No. -- 5 (ii)
 // Write a program to compute if one string is a rotation of another. For example, pit is rotation of tip as pit has same character as tip.

```

import java.util.Scanner;

public class Prac5_ii {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the first string: ");
        String str1 = sc.nextLine();

        System.out.print("Enter the second string: ");
        String str2 = sc.nextLine();

        boolean isRotation = isRotation(str1, str2);
        System.out.println(str1 + " is a rotation of " + str2 + ": " + isRotation);
    }

    public static boolean isRotation(String str1, String str2) {
        if (str1 == null || str2 == null) {
            System.out.println("One or both strings are null.");
            return false;
        }
    }
}

```

```
        if (str1.length() != str2.length()) {  
            return false;  
        }  
  
        String combined = str2 + str1;  
        return combined.contains(str1);  
    }  
}
```

Output –

```
C:\Users\HP\Desktop\23TBCP516D>javac Prac5_ii.java  
  
C:\Users\HP\Desktop\23TBCP516D>java Prac5_ii  
Enter the first string: Vraj  
Enter the second string: Vidhi  
Vraj is a rotation of Vidhi: false  
  
C:\Users\HP\Desktop\23TBCP516D>
```


Practical – 6

Aim – Study and implement Abstract class and Interfaces in Java.

- i. Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for specific object i.e. area() of Triangle subclass should calculate area of triangle etc. Same for Rectangle and Circle.
- ii. Write a Java program to create an abstract class Employee with abstract methods calculateSalary() and displayInfo(). Create subclasses Manager and Programmer that extend the Employee class and implement the respective methods to calculate salary and display information for each role.
- iii. Write a Java program to create an interface Shape with the getArea() method. Create three classes Rectangle, Circle, and Triangle that implement the Shape interface. Implement the getArea() method for each of the three classes.

Code –

```
// Practical No. -- 6 (i)
// Abstract Class - Area of Circle, Triangle and Rectangle
```

```
import java.util.Scanner;
```

```
abstract class Shape {
    abstract double area();
}
```

```
class Triangle extends Shape {
    private double base;
    private double height;
```

```
    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }
```

```
    @Override
    double area() {
        return 0.5 * base * height;
    }
}
```

```

class Rectangle extends Shape {
    private double width;
    private double height;

    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    @Override
    double area() {
        return width * height;
    }
}

class Circle extends Shape {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    double area() {
        return Math.PI * radius * radius;
    }
}

public class Prac6_i {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the base of the Triangle: ");
        double base = sc.nextDouble();
        System.out.print("Enter the height of the Triangle: ");
        double height = sc.nextDouble();
        Shape triangle = new Triangle(base, height);

        System.out.print("Enter the width of the Rectangle: ");
        double width = sc.nextDouble();
        System.out.print("Enter the height of the Rectangle: ");
        double rectHeight = sc.nextDouble();
        Shape rectangle = new Rectangle(width, rectHeight);

        System.out.print("Enter the radius of the Circle: ");
        double radius = sc.nextDouble();
        Shape circle = new Circle(radius);
    }
}

```

```

        System.out.println("Area of Triangle: " + triangle.area());
        System.out.println("Area of Rectangle: " + rectangle.area());
        System.out.println("Area of Circle: " + circle.area());
    }
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac6_i.java

C:\Users\HP\Desktop\23BCP516D>java Prac6_i
Enter the base of the Triangle: 20
Enter the height of the Triangle: 10
Enter the width of the Rectangle: 20
Enter the height of the Rectangle: 10
Enter the radius of the Circle: 20
Area of Triangle: 100.0
Area of Rectangle: 200.0
Area of Circle: 1256.6370614359173

C:\Users\HP\Desktop\23BCP516D>|

```

Code –

```

// Practical No. -- 6 (ii)
// Abstract Class - Employee

import java.util.Scanner;
abstract class Employee {
    abstract double sal();
    abstract void info();
}

class Manager extends Employee {
    private double base, bonus;

    Manager(double base, double bonus) {
        this.base = base;
        this.bonus = bonus;
    }

    @Override
    double sal() {
        return base + bonus;
    }

    @Override

```

```

void info() {
    System.out.printf("Manager's Details:%nBase Salary: $%.2f%nBonus: $%.2f%nTotal
Salary: $%.2f%n%n", base, bonus, sal());
}
}

```

```

class Programmer extends Employee {

```

```

    private double base;
    private int proj;

```

```

    Programmer(double base, int proj) {
        this.base = base;
        this.proj = proj;
    }

```

```

    @Override
    double sal() {
        return base + 500 * proj;
    }

```

```

    @Override
    void info() {
        System.out.printf("Programmer's Details:%nBase Salary: $%.2f%nNumber of Projects:
%d%nTotal Salary: $%.2f%n%n", base, proj, sal());
    }
}

```

```

public class Prac6_ii {

```

```

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

```

```

        System.out.println("Enter details for Manager:");
        System.out.print("Base Salary: $");
        double mBase = sc.nextDouble();
        System.out.print("Bonus: $");
        double mBonus = sc.nextDouble();
        Manager manager = new Manager(mBase, mBonus);

```

```

        System.out.println("\nEnter details for Programmer:");
        System.out.print("Base Salary: $");
        double pBase = sc.nextDouble();
        System.out.print("Number of Projects: ");
        int pProj = sc.nextInt();
        Programmer programmer = new Programmer(pBase, pProj);
        System.out.println("\nEmployee Details:");

```

```
        manager.info();
        programmer.info();
    }
}
```

Output –

```
C:\Users\HP\Desktop\23TBCP516D>javac Prac6_ii.java

C:\Users\HP\Desktop\23TBCP516D>java Prac6_ii
Enter details for Manager:
Base Salary: $10000
Bonus: $200

Enter details for Programmer:
Base Salary: $8000
Number of Projects: 5

Employee Details:
Manager's Details:
Base Salary: $10000.00
Bonus: $200.00
Total Salary: $10200.00

Programmer's Details:
Base Salary: $8000.00
Number of Projects: 5
Total Salary: $10500.00

C:\Users\HP\Desktop\23TBCP516D>
```

Code –

```
// Practical No. -- 6 (iii)
// Interface - Area of Circle, Triangle and Rectangle

import java.util.Scanner;

public class Prac6_iii {
    public interface Shape {
        double getArea();
    }

    public static class Rectangle implements Shape {
        private double width, height;

        public Rectangle(double width, double height) {
            this.width = width;
```

```

        this.height = height;
    }

    @Override
    public double getArea() {
        return width * height;
    }
}

public static class Circle implements Shape {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public double getArea() {
        return Math.PI * radius * radius;
    }
}

public static class Triangle implements Shape {
    private double base, height;

    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }

    @Override
    public double getArea() {
        return 0.5 * base * height;
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.println("Enter details for Rectangle:");
    System.out.print("Width: ");
    double rectWidth = sc.nextDouble();
    System.out.print("Height: ");
    double rectHeight = sc.nextDouble();
    Shape rectangle = new Rectangle(rectWidth, rectHeight);

    System.out.println("\nEnter details for Circle:");
    System.out.print("Radius: ");

```

```

        double circleRadius = sc.nextDouble();
        Shape circle = new Circle(circleRadius);

        System.out.println("\nEnter details for Triangle:");
        System.out.print("Base: ");
        double triBase = sc.nextDouble();
        System.out.print("Height: ");
        double triHeight = sc.nextDouble();
        Shape triangle = new Triangle(triBase, triHeight);

        System.out.println("\nShape Areas:");
        System.out.printf("Rectangle Area: %.2f%n", rectangle.getArea());
        System.out.printf("Circle Area: %.2f%n", circle.getArea());
        System.out.printf("Triangle Area: %.2f%n", triangle.getArea());

    }
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac6_iii.java

C:\Users\HP\Desktop\23BCP516D>java Prac6_iii
Enter details for Rectangle:
Width: 20
Height: 10

Enter details for Circle:
Radius: 10

Enter details for Triangle:
Base: 20
Height: 10

Shape Areas:
Rectangle Area: 200.00
Circle Area: 314.16
Triangle Area: 100.00

C:\Users\HP\Desktop\23BCP516D>|

```

Practical – 7

Aim – Study and implement Exception handling in Java

- i. Write a Java program for try-catch block in exception handling.
- ii. Write a Java for multiple catch block in exception handling.
- iii. Write a java program for nested of try in exception handling.
- iv. Write a small application in Java to develop Banking Application in which user deposits the amount Rs. 1000.00 and then start withdrawing of Rs. 400.00, Rs. 300.00 and it throws exception “Not Sufficient Fund” when user withdraws Rs. 500 thereafter.
- v. Write a java program for finally block in exception handling.

Code –

```
// Practical No. -- 7 (i)
// try-catch block in Exception Handling

public class Prac7_i
{
    public static void main(String[] args)
    {
        int numerator = 10;
        int denominator = 0;
        int result;

        try
        {
            // This will throw an ArithmeticException if denominator is 0
            result = numerator / denominator;
            System.out.println("\nResult: " + result);
        }

        catch (ArithmeticException e)
        {
            System.out.println("\nException caught: Division by zero is not allowed.");
        }

        System.out.println("Program continues after the try-catch block.");
    }
}
```


Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac7_i.java
C:\Users\HP\Desktop\23BCP516D>java Prac7_i
Exception caught: Division by zero is not allowed.
Program continues after the try-catch block.
C:\Users\HP\Desktop\23BCP516D>|
```

Code –

// Practical No. -- 7 (ii)

// Multiple catch block in Exception Handling

```
public class Prac7_ii
{
    public static void main(String[] args)
    {
        try
        {
            int[] numbers = { 10, 20, 30};
            int result = numbers[1] / 0; // This will cause ArithmeticException

            System.out.println(numbers[3]); // This will cause ArrayIndexOutOfBoundsException
        }

        catch (ArithmeticException e)
        {
            System.out.println("\nCannot divide by zero! " + e);
        }

        catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println("\nArray index is out of bounds! " + e);
        }

        catch (Exception e)
        {
            System.out.println("\nAn unexpected error occurred: " + e);
        }
    }
}
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac7_ii.java
C:\Users\HP\Desktop\23BCP516D>java Prac7_ii
Cannot divide by zero! java.lang.ArithmeticException: / by zero
C:\Users\HP\Desktop\23BCP516D>
```

Code –

```
// Practical No. -- 7 (iii)
// Nested of try in Exception Handling

public class Prac7_iii
{
    public static void main(String[] args)
    {
        // Outer try block
        try
        {
            // Array with three elements
            int[] numbers = { 1, 2, 3 };
            System.out.println("\nOuter try block: Accessing the array element...");

            // Inner try block
            try
            {
                // Trying to divide by zero
                System.out.println("Inner try block: Trying to divide...");
                int result = numbers[2] / 0; // This will throw an ArithmeticException
                System.out.println("Result of division: " + result);
            }

            catch (ArithmeticException e)
            {
                System.out.println("Inner catch block: ArithmeticException caught - " +
e.getMessage());
            }

            // Accessing an element out of array bounds
            System.out.println("Accessing out of bounds element: " + numbers[5]);
        }

        catch (ArrayIndexOutOfBoundsException e)
        {

```

```

        System.out.println("Outer catch block: ArrayIndexOutOfBoundsException caught - " +
e.getMessage());
    }

    System.out.println("Program ended normally.");
}
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac7_iii.java
C:\Users\HP\Desktop\23BCP516D>java Prac7_iii

Outer try block: Accessing the array element...
Inner try block: Trying to divide...
Inner catch block: ArithmeticException caught - / by zero
Outer catch block: ArrayIndexOutOfBoundsException caught - Index 5 out of bounds for length 3
Program ended normally.

C:\Users\HP\Desktop\23BCP516D>

```

Code –

```

// Practical No. -- 7 (iv)
// Banking Application

```

```

class InsufficientFundsException extends Exception {
    public InsufficientFundsException(String message) {
        super(message);
    }
}

class BankAccount {
    private double balance;

    public BankAccount(double initialBalance) {
        this.balance = initialBalance;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("\nDeposited: Rs. " + amount + " | Current Balance: Rs. " + balance);
    }

    public void withdraw(double amount) throws InsufficientFundsException {
        if (amount > balance) {
            throw new InsufficientFundsException("Not Sufficient Fund");
        }
        balance -= amount;
        System.out.println("Withdrawn: Rs. " + amount + " | Current Balance: Rs. " + balance);
    }
}

```

```

        public double getBalance() {
            return balance;
        }
    }

    public class Prac7_iv {
        public static void main(String[] args) {
            BankAccount account = new BankAccount(0.00);

            try {
                account.deposit(1000.00);
                account.withdraw(400.00);
                account.withdraw(300.00);
                account.withdraw(500.00); // This will throw an exception
            } catch (InsufficientFundsException e) {
                System.out.println("Exception: " + e.getMessage());
            }
        }
    }
}

```

Output –

```

C:\Users\HP\Desktop\23TBCP516D>javac Prac7_iv.java

C:\Users\HP\Desktop\23TBCP516D>java Prac7_iv

Deposited: Rs. 1000.0 | Current Balance: Rs. 1000.0
Withdrawn: Rs. 400.0 | Current Balance: Rs. 600.0
Withdrawn: Rs. 300.0 | Current Balance: Rs. 300.0
Exception: Not Sufficient Fund

C:\Users\HP\Desktop\23TBCP516D>

```

Code –

```

// Practical No. -- 7 (v)
// finally Block in Exception Handling

public class Prac7_v
{
    public static void main(String[] args)
    {
        try
        {
            int a = 10;
            int b = 0;
            int result = a / b; // This will cause an ArithmeticException
            System.out.println("\nResult: " + result);
        }
    }
}

```

```
        catch (ArithmeticException e)
        {
            System.out.println("\nException caught: " + e);
        }

        finally
        {
            System.out.println("This is the finally block. It always executes.");
        }

        System.out.println("End of program.");
    }
}
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac7_v.java
C:\Users\HP\Desktop\23BCP516D>java Prac7_v
Exception caught: java.lang.ArithmeticException: / by zero
This is the finally block. It always executes.
End of program.
C:\Users\HP\Desktop\23BCP516D>
```

Practical – 8

Aim – Study and implement File Handling in Java

- i. Read a content from file: calculate number of sentences, words, and characters from the file.
- ii. Read content from a file convert it to uppercase and save it into another file.
- iii. Remove duplicate lines from a File.
- iv. Create a class called Student. Write a student manager program to manipulate the student information from files by using FileInputStream and FileOutputStream
- v. Refine the student manager program to manipulate the student information from files by using the BufferedReader and BufferedWriter
- vi. Write a program to manipulate the information from files by using the Reader and Writer class. Assume suitable data.

Code –

```
// Practical No. -- 8 (i)
// Read a content from file: calculate number of sentences, words and characters from the file.

import java.io.FileInputStream;
import java.io.IOException;
import java.util.Scanner;

class Prac8_i {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input file name
        System.out.print("Enter the file name: ");
        String fileName = sc.nextLine();

        try (FileInputStream fis = new FileInputStream(fileName)) {
            int characterCount = 0;
            int wordCount = 0;
            int sentenceCount = 0;
            StringBuilder fileContent = new StringBuilder();
            int i;

            // Read the file content
            while ((i = fis.read()) != -1) {
                char currentChar = (char) i;
                fileContent.append(currentChar);
                characterCount++;
            }
        }
    }
}
```

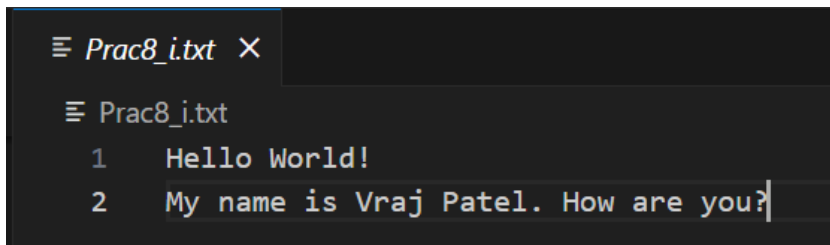
```

        // Count words (assumes space as word separator)
        if (currentChar == ' ' || currentChar == '\n' || currentChar == '\r') {
            wordCount++;
        }
        // Count sentences (assumes ., !, ? as sentence terminators)
        if (currentChar == '.' || currentChar == '!' || currentChar == '?') {
            sentenceCount++;
        }
    }
    // Handle the last word if the file does not end with a space or newline
    if (fileContent.length() > 0) {
        wordCount++;
    }
    System.out.println("\nFile content:\n" + fileContent);
    System.out.println("\nNumber of characters: " + characterCount);
    System.out.println("Number of words: " + wordCount);
    System.out.println("Number of sentences: " + sentenceCount);

} catch (IOException e) {
    System.out.println("An error occurred while reading the file: " + e.getMessage());
} finally {
    sc.close();
}
}
}

```

Output –

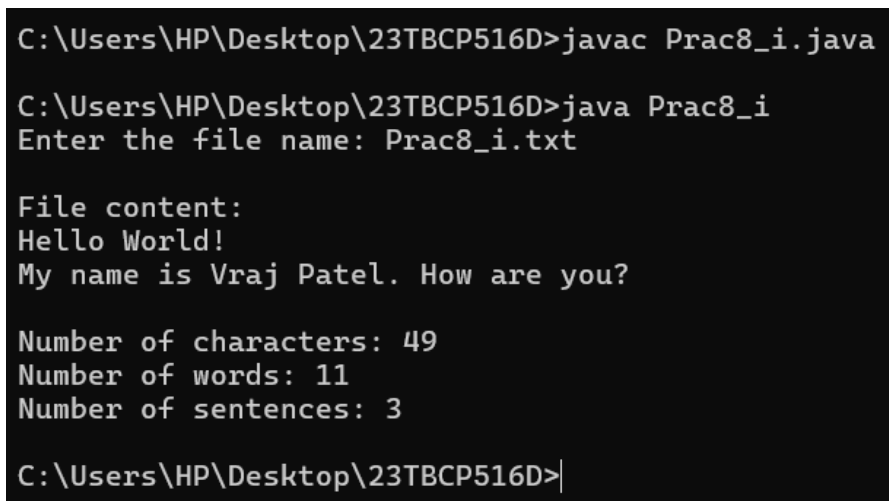


The screenshot shows a text editor window titled 'Prac8_i.txt'. The content of the file is as follows:

```

1 Hello World!
2 My name is Vraj Patel. How are you?

```



The screenshot shows a command prompt window with the following text:

```

C:\Users\HP\Desktop\23TBCP516D>javac Prac8_i.java

C:\Users\HP\Desktop\23TBCP516D>java Prac8_i
Enter the file name: Prac8_i.txt

File content:
Hello World!
My name is Vraj Patel. How are you?

Number of characters: 49
Number of words: 11
Number of sentences: 3

C:\Users\HP\Desktop\23TBCP516D>

```

Code –

```
// Practical No. -- 8 (ii)
// Read content from a file convert it to uppercase and save it into another file.

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class Prac8_ii {

    public static void main(String[] args) {
        // Specify the input and output file names
        String inputFile = "Prac8_ii.txt";
        String outputFile = "Prac8_ii_UC.txt";

        // Call the method to convert the file content to uppercase
        convertToUppercase(inputFile, outputFile);
    }

    public static void convertToUppercase(String inputFile, String outputFile) {
        // Using try-with-resources to automatically close resources
        try (FileInputStream inputStream = new FileInputStream(inputFile);
            FileOutputStream outputStream = new FileOutputStream(outputFile)) {

            int byteRead;

            // Read each byte from the input file
            while ((byteRead = inputStream.read()) != -1) {
                // Convert to uppercase if it's a lowercase letter
                if (byteRead >= 'a' && byteRead <= 'z') {
                    byteRead -= 32; // Convert lowercase to uppercase
                }
                // Write the byte to the output file
                outputStream.write(byteRead);
            }

            System.out.println("Conversion to UPPERCASE completed successfully!");

        } catch (IOException e) {
            System.err.println("An error occurred during file operations: " + e.getMessage());
        }
    }
}
```


Output –

```
≡ Prac8_ii.txt X
≡ Prac8_ii.txt
1 Hello World!
2 My name is Vraj Patel. How are you?
```

```
C:\Users\HP\Desktop\23BCP516D>javac Prac8_ii.java

C:\Users\HP\Desktop\23BCP516D>java Prac8_ii
Conversion to UPPERCASE completed successfully!
```

```
≡ Prac8_ii_UC.txt X
≡ Prac8_ii_UC.txt
1 HELLO WORLD!
2 MY NAME IS VRAJ PATEL. HOW ARE YOU?
```

Code –

```
// Practical No. -- 8 (iii)
// Remove Duplicate Lines from a file

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class Prac8_iii {
    public static void main(String[] args) {
        String inputfile = "Prac8_iii.txt";
        String outputfile = "Prac8_iiiOUTPUT.txt";

        try (FileInputStream fis = new FileInputStream(inputfile);
            FileOutputStream fos = new FileOutputStream(outputfile)) {

            StringBuilder currentline = new StringBuilder();
            int byteData;
            boolean isnewline = false;

            while ((byteData = fis.read()) != -1) {
                char character = (char) byteData;
```

```

        if (character == '\n') {
            isnewline = true;
            if (currentline.length() > 0) {
                writeUniqueLine(fos, currentline.toString());
                currentline.setLength(0);
            }
        } else {
            currentline.append(character);
        }
    }
    if (currentline.length() > 0) {
        writeUniqueLine(fos, currentline.toString());
    }
    System.out.println("Duplicate lines removed. Check the output file: " + outputfile);
} catch (IOException e) {
    e.printStackTrace();
}
}

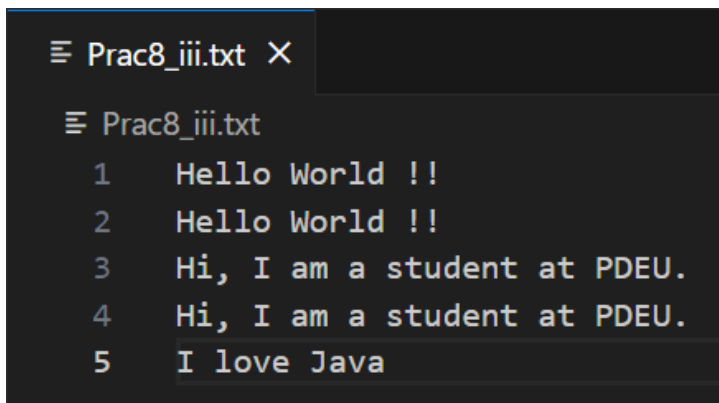
public static void writeUniqueLine(FileOutputStream fos, String line) throws IOException {
    try (FileInputStream checkFis = new FileInputStream("Prac8_iiiOUTPUT.txt")) {
        StringBuilder existingLines = new StringBuilder();
        int byteData;
        boolean isDuplicate = false;

        while ((byteData = checkFis.read()) != -1) {
            existingLines.append((char) byteData);
        }

        // Check if the line already exists in existingLines
        if (!existingLines.toString().contains(line)) {
            fos.write((line + System.lineSeparator()).getBytes());
        }
    }
}
}

```

Output –



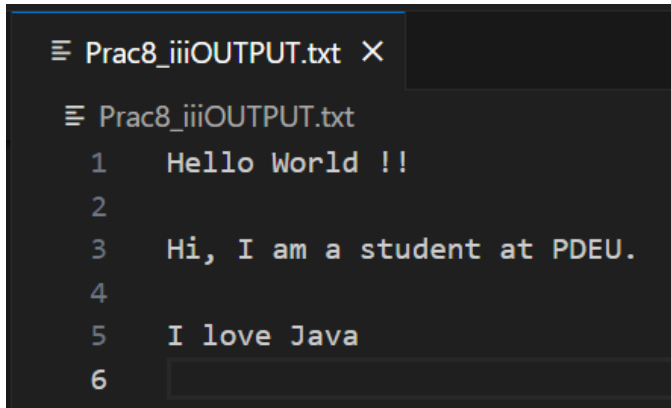
```

Prac8_iii.txt X
Prac8_iii.txt
1  Hello World !!
2  Hello World !!
3  Hi, I am a student at PDEU.
4  Hi, I am a student at PDEU.
5  I love Java

```

```
C:\Users\HP\Desktop\23BCP516D>javac Prac8_iii.java

C:\Users\HP\Desktop\23BCP516D>java Prac8_iii
Duplicate lines removed. Check the output file: Prac8_iiiOUTPUT.txt
```



```
Prac8_iiiOUTPUT.txt X
Prac8_iiiOUTPUT.txt
1   Hello World !!
2
3   Hi, I am a student at PDEU.
4
5   I love Java
6
```

Code –

```
// Practical No. -- 8 (iv)
// Student Manager using FileInputStream and FileOutputStream.
```

```
import java.io.*;
import java.util.ArrayList;
import java.util.List;
```

```
class Student implements Serializable {
    private int id;
    private String name;
    private int age;

    public Student(int id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}
```

```

    public void displayInfo() {
        System.out.println("ID: " + id + ", Name: " + name + ", Age: " + age);
    }
}

public class Prac8_iv {
    private static final String FILE_PATH = "Prac8_iv.txt";

    public void saveStudents(List<Student> students) {
        try (FileOutputStream fileOut = new FileOutputStream(FILE_PATH);
            ObjectOutputStream out = new ObjectOutputStream(fileOut)) {
            out.writeObject(students);
            System.out.println("Student data saved successfully.");
        } catch (IOException e) {
            System.out.println("Error saving student data: " + e.getMessage());
        }
    }

    @SuppressWarnings("unchecked")
    public List<Student> loadStudents() {
        List<Student> students = new ArrayList<>();
        try (FileInputStream fileIn = new FileInputStream(FILE_PATH);
            ObjectInputStream in = new ObjectInputStream(fileIn)) {
            students = (List<Student>) in.readObject();
        } catch (FileNotFoundException e) {
            System.out.println("No existing student data found. A new file will be created.");
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("Error loading student data: " + e.getMessage());
        }
        return students;
    }

    public void addStudent(Student student) {
        List<Student> students = loadStudents();
        students.add(student);
        saveStudents(students);
    }

    public void displayAllStudents() {
        List<Student> students = loadStudents();
        if (students.isEmpty()) {
            System.out.println("No student data found.");
        } else {
            System.out.println("Student List:");
            for (Student student : students) {
                student.displayInfo();
            }
        }
    }
}

```

```

    }
}

public static void main(String[] args) {
    Prac8_iv manager = new Prac8_iv();
    manager.addStudent(new Student(1, "Vraj", 20));
    manager.addStudent(new Student(2, "Vidhi", 18));
    manager.displayAllStudents();
}
}

```

Output –

```

C:\Users\HP\Desktop\23TBCP516D>javac Prac8_iv.java

C:\Users\HP\Desktop\23TBCP516D>java Prac8_iv
No existing student data found. A new file will be created.
Student data saved successfully.
Student data saved successfully.
Student List:
ID: 1, Name: Vraj, Age: 20
ID: 2, Name: Vidhi, Age: 18

C:\Users\HP\Desktop\23TBCP516D>

```

Code –

```

// Practical No. -- 8 (v)
// Student Manager using BufferedReader and BufferedWriter

```

```

import java.io.*;
import java.util.ArrayList;
import java.util.List;

class Student {
    private int id;
    private String name;
    private int age;

    public Student(int id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }
}

```

```

    public int getAge() {
        return age;
    }

    @Override
    public String toString() {
        return "Student{id=" + id + ", name=" + name + ", age=" + age + "}";
    }
}

public class Prac8_v {
    // Write student data to a file
    public static void writeStudentsToFile(List<Student> students, String fileName) {
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName))) {
            for (Student student : students) {
                writer.write(student.getId() + "," + student.getName() + "," + student.getAge());
                writer.newLine(); // Add a new line for each student
            }
            System.out.println("Student data saved to file.");
        } catch (IOException e) {
            System.out.println("Error writing to file: " + e.getMessage());
        }
    }

    // Read student data from a file
    public static List<Student> readStudentsFromFile(String fileName) {
        List<Student> students = new ArrayList<>();
        try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts.length == 3) {
                    try {
                        int id = Integer.parseInt(parts[0]);
                        String name = parts[1];
                        int age = Integer.parseInt(parts[2]);
                        students.add(new Student(id, name, age));
                    } catch (NumberFormatException e) {
                        System.out.println("Error parsing student data: " + e.getMessage());
                    }
                } else {
                    System.out.println("Skipping invalid line: " + line);
                }
            }
            System.out.println("Student data loaded from file.");
        } catch (IOException e) {
            System.out.println("Error reading from file: " + e.getMessage());
        }
    }
}

```

```

        return students;
    }

    public static void main(String[] args) {
        List<Student> students = new ArrayList<>();
        students.add(new Student(1, "Vraj Patel", 20));
        students.add(new Student(2, "Vidhi Dave", 18));
        String fileName = "Prac8_v.txt";

        // Write students to file
        writeStudentsToFile(students, fileName);

        // Read students from file
        List<Student> loadedStudents = readStudentsFromFile(fileName);
        System.out.println("Loaded Students:");
        for (Student student : loadedStudents) {
            System.out.println(student);
        }
    }
}

```

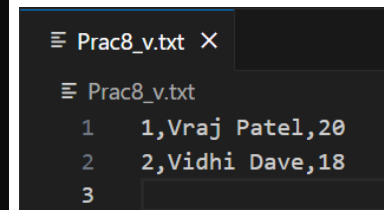
Output –

```

C:\Users\HP\Desktop\23TBCP516D>javac Prac8_v.java

C:\Users\HP\Desktop\23TBCP516D>java Prac8_v
Student data saved to file.
Student data loaded from file.
Loaded Students:
Student{id=1, name='Vraj Patel', age=20}
Student{id=2, name='Vidhi Dave', age=18}
C:\Users\HP\Desktop\23TBCP516D>

```



```

Prac8_v.txt X
Prac8_v.txt
1 1,Vraj Patel,20
2 2,Vidhi Dave,18
3

```

Code –

```

// Practical No. -- 8 (vi)
// Manipulate the information from files by using the Reader and Writer class.

```

```

import java.io.*;

public class Prac8_vi {
    public static void main(String[] args) {
        try {
            // Create a FileReader to read from a file
            FileReader reader = new FileReader("Prac8_vi.txt");
            // Create a FileWriter to write to a file
            FileWriter writer = new FileWriter("Prac8_viOUTPUT.txt");

            // Read and manipulate data

```

```

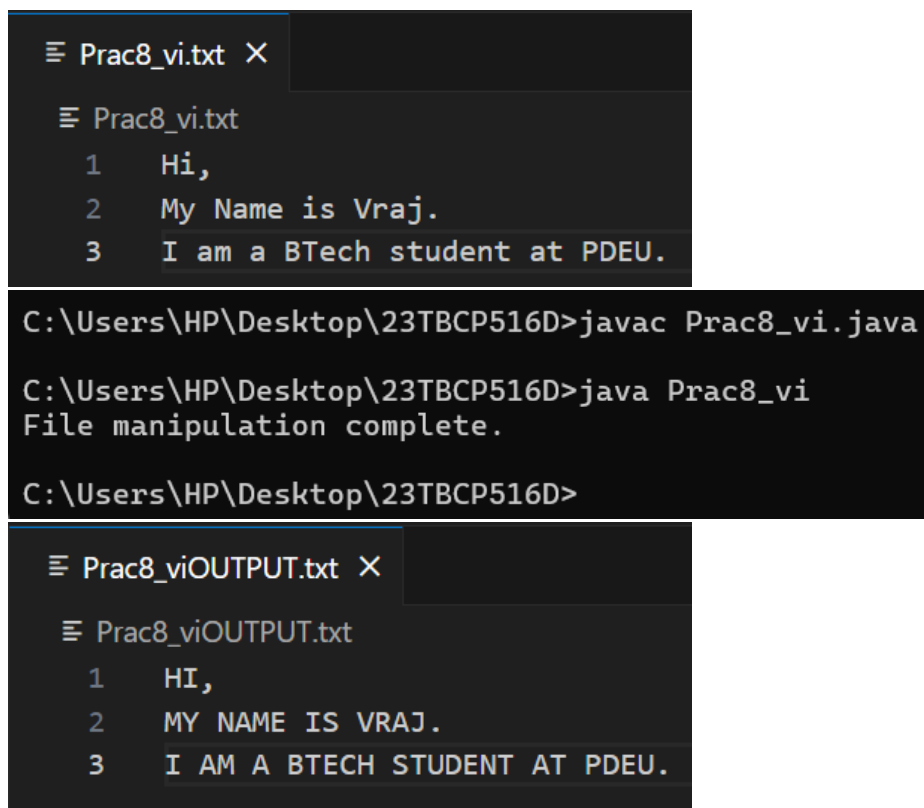
int character;
while ((character = reader.read()) != -1) {
    // Convert lowercase letters to uppercase
    if (Character.isLowerCase(character)) {
        character = Character.toUpperCase(character);
    }
    // Write the manipulated character to the output file
    writer.write(character);
}

// Close the reader and writer
reader.close();
writer.close();
System.out.println("File manipulation complete.");

} catch (IOException e) {
    System.out.println("An error occurred: " + e.getMessage());
}
}
}

```

Output –



The screenshot shows two windows from a code editor. The top window, titled 'Prac8_vi.txt', contains the following text:

```

1  Hi,
2  My Name is Vraj.
3  I am a BTech student at PDEU.

```

Below this window is a terminal window showing the execution of the Java program:

```

C:\Users\HP\Desktop\23BCP516D>javac Prac8_vi.java
C:\Users\HP\Desktop\23BCP516D>java Prac8_vi
File manipulation complete.
C:\Users\HP\Desktop\23BCP516D>

```

The bottom window, titled 'Prac8_viOUTPUT.txt', shows the output of the program:

```

1  HI,
2  MY NAME IS VRAJ.
3  I AM A BTECH STUDENT AT PDEU.

```


Practical – 9

Aim – Study and implement multi-threaded application in Java

- i. Write a Java program to demonstrate how to create and start a thread using both the Thread class and the Runnable interface.
- ii. Write a Java program that illustrates thread synchronization by ensuring multiple threads can safely access a shared resource without causing data inconsistency.
- iii. Write a Java program to demonstrate inter-thread communication using wait(), notify(), and notifyAll() methods, allowing threads to communicate and coordinate their actions.
- iv. Write a Java program to show how thread priority affects the execution order of threads, highlighting the use of setPriority() and getPriority() methods.
- v. Write a Java program to implement the producer-consumer problem, ensuring the handling of potential deadlock conditions using proper synchronization techniques.

Code –

```
// Practical No. -- 9 (i)
// Thread class and the Runnable interface

class MyThread extends Thread {
    @Override
    public void run() {
        System.out.println("Task 1 from MyThread");
    }
}

class MyRunnable implements Runnable {
    @Override
    public void run() {
        System.out.println("Task 2 from MyRunnable");
    }
}

public class Prac9_i {
    public static void main(String[] args) {
        MyThread thread1 = new MyThread();
        thread1.start();

        Thread thread2 = new Thread(new MyRunnable());
        thread2.start();
    }
}
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac9_i.java

C:\Users\HP\Desktop\23BCP516D>java Prac9_i
Task 1 from MyThread
Task 2 from MyRunnable

C:\Users\HP\Desktop\23BCP516D>
```

Code –

```
// Practical No. -- 9 (ii)
// Illustrating thread synchronization by ensuring multiple threads

class Counter {
    private int count = 0;

    // Synchronized method to increment the count
    public synchronized void increment() {
        count++;
    }

    public int getCount() {
        return count;
    }
}

class CounterThread extends Thread {
    private Counter counter;

    public CounterThread(Counter counter) {
        this.counter = counter;
    }

    @Override
    public void run() {
        counter.increment();
        System.out.println(Thread.currentThread().getName() + " incremented count to: " +
        counter.getCount());
    }
}

public class Prac9_ii {
    public static void main(String[] args) {
        System.out.println("23BCP514D");
        Counter counter = new Counter();
    }
}
```

```

// Creating multiple threads that share the same Counter object
CounterThread thread1 = new CounterThread(counter);
CounterThread thread2 = new CounterThread(counter);
CounterThread thread3 = new CounterThread(counter);

// Starting the threads
thread1.start();
thread2.start();
thread3.start();
}
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac9_ii.java

C:\Users\HP\Desktop\23BCP516D>java Prac9_ii
Thread-1 incremented count to: 2
Thread-2 incremented count to: 3
Thread-0 incremented count to: 1

C:\Users\HP\Desktop\23BCP516D>

```

Code –

```

// Practical No. -- 9 (iii)
// Demonstrating inter-thread communication

class Data {
    private int value;
    private boolean isProduced = false;

    // Producer method to generate data
    public synchronized void produce(int value) throws InterruptedException {
        while (isProduced) {
            wait(); // Wait if data is already produced
        }
        this.value = value;
        isProduced = true;
        System.out.println("Produced: " + value);
        notify(); // Notify the consumer that data is available
    }

    // Consumer method to consume data
    public synchronized void consume() throws InterruptedException {
        while (!isProduced) {
            wait(); // Wait if no data is produced
        }
        System.out.println("Consumed: " + value);
    }
}

```

```

        isProduced = false;
        notify(); // Notify the producer that data is consumed
    }
}

class Producer extends Thread {
    private Data data;

    public Producer(Data data) {
        this.data = data;
    }

    @Override
    public void run() {
        try {
            for (int i = 1; i <= 5; i++) {
                data.produce(i);
                Thread.sleep(1000); // Simulate time taken to produce data
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

class Consumer extends Thread {
    private Data data;

    public Consumer(Data data) {
        this.data = data;
    }

    @Override
    public void run() {
        try {
            for (int i = 1; i <= 5; i++) {
                data.consume();
                Thread.sleep(1500); // Simulate time taken to consume data
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

public class Prac9_iii {
    public static void main(String[] args) {
        Data data = new Data();
    }
}

```

```

        Producer producer = new Producer(data);
        Consumer consumer = new Consumer(data);
        producer.start();
        consumer.start();
    }
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac9_iii.java

C:\Users\HP\Desktop\23BCP516D>java Prac9_iii
Produced: 1
Consumed: 1
Produced: 2
Consumed: 2
Produced: 3
Consumed: 3
Produced: 4
Consumed: 4
Produced: 5
Consumed: 5

C:\Users\HP\Desktop\23BCP516D>

```

Code –

```

// Practical No. -- 9 (iv)
// Thread Priority

```

```

class MyThread extends Thread {
    public MyThread(String name) {
        super(name);
    }

    @Override
    public void run() {
        // Print thread name and its priority
        System.out.println(Thread.currentThread().getName() + " with priority " +
            Thread.currentThread().getPriority() + " is running.");
    }
}

public class Prac9_iv {
    public static void main(String[] args) {
        MyThread thread1 = new MyThread("Thread 1");
        MyThread thread2 = new MyThread("Thread 2");
        MyThread thread3 = new MyThread("Thread 3");

        // Set thread priorities
        thread1.setPriority(Thread.MIN_PRIORITY); // Priority 1

```

```

thread2.setPriority(Thread.NORM_PRIORITY); // Priority 5 (default)
thread3.setPriority(Thread.MAX_PRIORITY); // Priority 10

// Print the priorities
System.out.println(thread1.getName() + " priority: " + thread1.getPriority());
System.out.println(thread2.getName() + " priority: " + thread2.getPriority());
System.out.println(thread3.getName() + " priority: " + thread3.getPriority());

// Start the threads
thread1.start();
thread2.start();
thread3.start();
}
}

```

Output –

```

C:\Users\HP\Desktop\23BCP516D>javac Prac9_iv.java

C:\Users\HP\Desktop\23BCP516D>java Prac9_iv
Thread 1 priority: 1
Thread 2 priority: 5
Thread 3 priority: 10
Thread 3 with priority 10 is running.
Thread 2 with priority 5 is running.
Thread 1 with priority 1 is running.

C:\Users\HP\Desktop\23BCP516D>

```

Code –

```

// Practical No. -- 9 (v)
// Deadlock conditions using proper synchronization techniques.

class Buffer {
    private int value;
    private boolean isProduced = false; // Flag to check if value is produced or not

    // Producer method to produce data
    public synchronized void produce(int value) throws InterruptedException {
        while (isProduced) {
            wait(); // Wait if data has already been produced
        }
        this.value = value;
        System.out.println("Produced: " + value);
        isProduced = true;
        notify(); // Notify consumer that data is available
    }

    // Consumer method to consume data

```

```

        public synchronized void consume() throws InterruptedException {
            while (!isProduced) {
                wait(); // Wait if no data is produced
            }
            System.out.println("Consumed: " + value);
            isProduced = false;
            notify(); // Notify producer that data has been consumed
        }
    }

    class Producer extends Thread {
        private Buffer buffer;

        public Producer(Buffer buffer) {
            this.buffer = buffer;
        }

        @Override
        public void run() {
            try {
                for (int i = 1; i <= 5; i++) {
                    buffer.produce(i);
                    Thread.sleep(1000); // Simulate time taken to produce data
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

    class Consumer extends Thread {
        private Buffer buffer;

        public Consumer(Buffer buffer) {
            this.buffer = buffer;
        }

        @Override
        public void run() {
            try {
                for (int i = 1; i <= 5; i++) {
                    buffer.consume();
                    Thread.sleep(1500); // Simulate time taken to consume data
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```
public class Prac9_v {  
    public static void main(String[] args) {  
        Buffer buffer = new Buffer(); // Shared buffer  
        Producer producer = new Producer(buffer);  
        Consumer consumer = new Consumer(buffer);  
        producer.start();  
        consumer.start();  
    }  
}
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac9_v.java  
  
C:\Users\HP\Desktop\23BCP516D>java Prac9_v  
Produced: 1  
Consumed: 1  
Produced: 2  
Consumed: 2  
Produced: 3  
Consumed: 3  
Produced: 4  
Consumed: 4  
Produced: 5  
Consumed: 5  
  
C:\Users\HP\Desktop\23BCP516D>
```


Practical – 10

Aim – GUI programming using Java Applet/Swings Components and Event Handling

- i. Write a Java program to demonstrate various window handling events such as `windowOpened()`, `windowClosing()`, `windowClosed()`, `windowIconified()`, `windowDeiconified()`, `windowActivated()`, and `windowDeactivated()`.
- ii. Write a Java program to demonstrate various mouse handling events including `mouseClicked()`, `mouseEntered()`, `mouseExited()`, `mousePressed()`, `mouseReleased()`, and `mouseDragged()`.
- iii. Write a Java program to demonstrate different keyboard handling events such as `keyPressed()`, `keyReleased()`, and `keyTyped()`.
- iv. Write a Java program to create a simple GUI that includes a button and a label. When the button is clicked, the text of the label should change accordingly.

Code –

```
// Practical No. -- 10 (i)
// Demonstrate various Window Handling Events

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

public class Prac10_i extends JFrame implements WindowListener {

    public Prac10_i() {
        // Set up the JFrame
        setTitle("Window Event Demo");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);

        // Add the WindowListener to the frame
        addWindowListener(this);

        // Label to display the event
        JLabel label = new JLabel("Perform a window event to see the message here.",
        SwingConstants.CENTER);
        add(label);
    }

    // Methods to handle each window event
```

```

@Override
public void windowOpened(WindowEvent e) {
    System.out.println("Window opened");
}

@Override
public void windowClosing(WindowEvent e) {
    System.out.println("Window closing");
    int choice = JOptionPane.showConfirmDialog(this, "Are you sure you want to close the
window?", "Close Confirmation", JOptionPane.YES_NO_OPTION);
    if (choice == JOptionPane.YES_OPTION) {
        dispose(); // Close the window
    }
}

@Override
public void windowClosed(WindowEvent e) {
    System.out.println("Window closed");
}

@Override
public void windowIconified(WindowEvent e) {
    System.out.println("Window iconified (minimized)");
}

@Override
public void windowDeiconified(WindowEvent e) {
    System.out.println("Window deiconified (restored)");
}

@Override
public void windowActivated(WindowEvent e) {
    System.out.println("Window activated");
}

@Override
public void windowDeactivated(WindowEvent e) {
    System.out.println("Window deactivated");
}

public static void main(String[] args) {
    // Run the program
    Prac10_i frame = new Prac10_i();
    frame.setVisible(true);
}
}

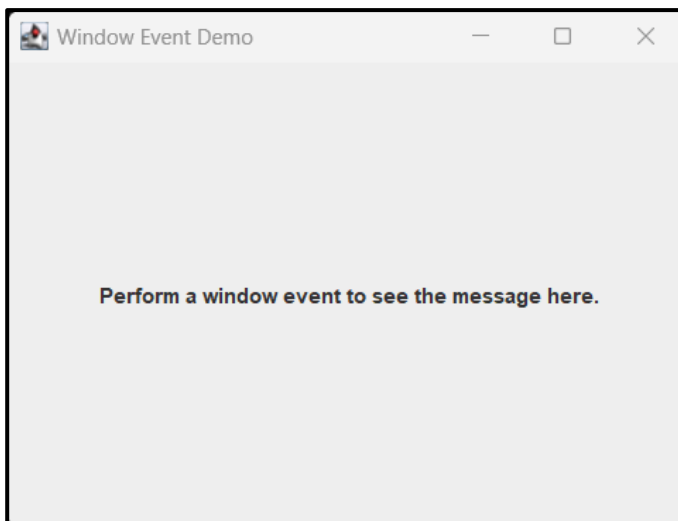
```

Output –

```
C:\Users\HP\Desktop\23BCP516D>javac Prac10_i.java

C:\Users\HP\Desktop\23BCP516D>java Prac10_i
Window activated
Window opened
Window iconified (minimized)
Window deactivated
Window deiconified (restored)
Window activated
Window deactivated
Window activated
Window closing
Window deactivated
Window closed

C:\Users\HP\Desktop\23BCP516D>
```



Code –

```
// Practical No. -- 10 (ii)
// Demonstrate various Mouse Handling Events

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

public class Prac10_ii extends JFrame implements MouseListener, MouseMotionListener {

    private JLabel label;

    public Prac10_ii() {
        // Set up the JFrame
        setTitle("Mouse Event Demo");
    }
}
```

```

setSize(500, 400);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Label to display event messages
label = new JLabel("Perform a mouse event to see the message here.",
SwingConstants.CENTER);
label.setFont(new Font("Arial", Font.PLAIN, 16));
add(label, BorderLayout.CENTER);

// Add MouseListener and MouseMotionListener to the frame
addMouseListener(this);
addMouseMotionListener(this);
}

// MouseListener methods

@Override
public void mouseClicked(MouseEvent e) {
    label.setText("Mouse clicked at (" + e.getX() + ", " + e.getY() + ")");
}

@Override
public void mouseEntered(MouseEvent e) {
    label.setText("Mouse entered the window");
}

@Override
public void mouseExited(MouseEvent e) {
    label.setText("Mouse exited the window");
}

@Override
public void mousePressed(MouseEvent e) {
    label.setText("Mouse pressed at (" + e.getX() + ", " + e.getY() + ")");
}

@Override
public void mouseReleased(MouseEvent e) {
    label.setText("Mouse released at (" + e.getX() + ", " + e.getY() + ")");
}

// MouseMotionListener methods

@Override
public void mouseDragged(MouseEvent e) {
    label.setText("Mouse dragged to (" + e.getX() + ", " + e.getY() + ")");
}

```

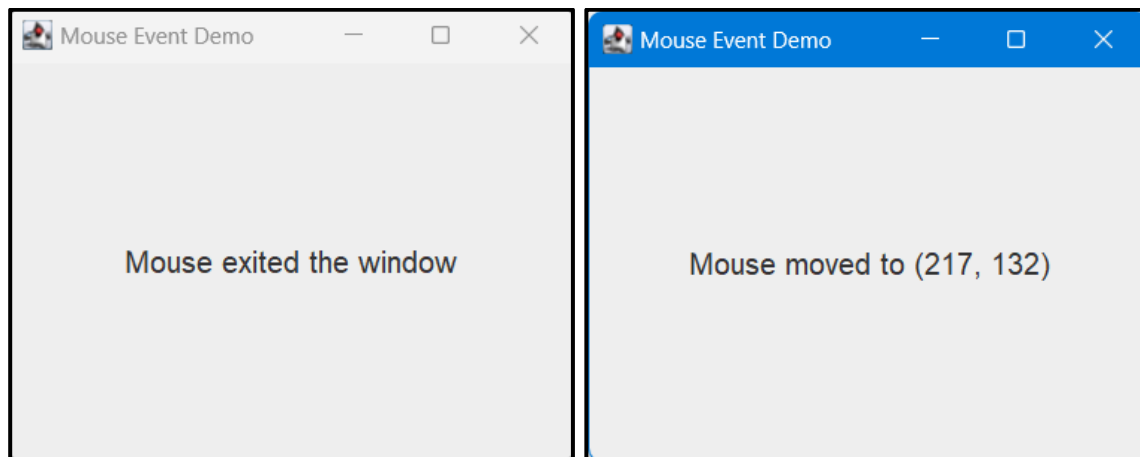
```

@Override
public void mouseMoved(MouseEvent e) {
    label.setText("Mouse moved to (" + e.getX() + ", " + e.getY() + ")");
}

public static void main(String[] args) {
    // Run the program
    Prac10_ii frame = new Prac10_ii();
    frame.setVisible(true);
}
}

```

Output –



Code –

```

// Practical No. -- 10 (iii)
// Demonstrate various Keyboard Handling Events

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

public class Prac10_iii extends JFrame implements KeyListener {

    private JLabel label;

    public Prac10_iii() {
        // Set up the JFrame
        setTitle("Keyboard Event Demo");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Label to display key event messages
        label = new JLabel("Press any key to see the event here.", SwingConstants.CENTER);
        label.setFont(label.getFont().deriveFont(16f));
        add(label);
    }
}

```

```

// Add KeyListener to the frame
addKeyListener(this);

// Make sure the frame is focusable to capture key events
setFocusable(true);
setFocusTraversalKeysEnabled(false);
}

// KeyListener methods

@Override
public void keyPressed(KeyEvent e) {
    label.setText("Key Pressed: " + KeyEvent.getKeyText(e.getKeyCode()));
}

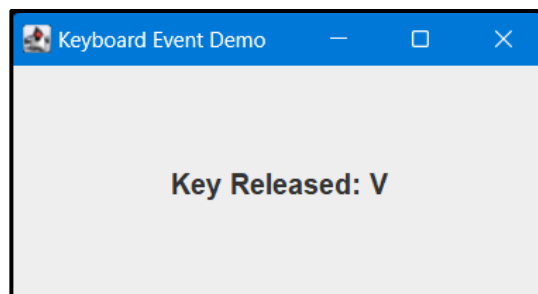
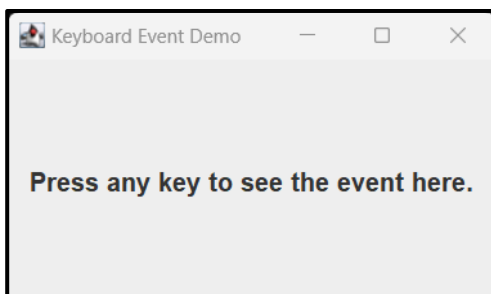
@Override
public void keyReleased(KeyEvent e) {
    label.setText("Key Released: " + KeyEvent.getKeyText(e.getKeyCode()));
}

@Override
public void keyTyped(KeyEvent e) {
    label.setText("Key Typed: " + e.getKeyChar());
}

public static void main(String[] args) {
    // Run the program
    Prac10_iii frame = new Prac10_iii();
    frame.setVisible(true);
}
}

```

Output –



Code –

// Practical No. -- 10 (iv)

// Button Click

```
import javax.swing.*;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
public class Prac10_iv extends JFrame {
```

```
    private JLabel label;
```

```
    private JButton button;
```

```
    public Prac10_iv() {
```

```
        // Set up the JFrame
```

```
        setTitle("Button Click Demo");
```

```
        setSize(300, 200);
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        setLayout(null);
```

```
        // Initialize label and button
```

```
        label = new JLabel("Click the button to change this text.", SwingConstants.CENTER);
```

```
        label.setBounds(30, 30, 240, 30); // Position label
```

```
        add(label);
```

```
        button = new JButton("Click Me");
```

```
        button.setBounds(100, 80, 100, 30); // Position button
```

```
        add(button);
```

```
        // Add ActionListener to the button
```

```
        button.addActionListener(new ActionListener() {
```

```
            @Override
```

```
            public void actionPerformed(ActionEvent e) {
```

```
                label.setText("Button was clicked!");
```

```
            }
```

```
        });
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        // Run the program
```

```
        Prac10_iv frame = new Prac10_iv();
```

```
        frame.setVisible(true);
```

```
    }
```

```
}
```

Output –

