**Q.9. Write a java program to create user defined package.**

package mypackage;

public class MyClass {

public void display() {

System.out.println("This is MyClass in the mypackage package.");

}

}

\*\*\*

package mypackage;

public class MainClass {

public static void main(String[] args) {

MyClass myObject = new MyClass();

myObject.display();

}

}

**Q. 10. Write a java program for creating multiple catch blocks.**

public class MultipleCatchBlock1 {

public static void main(String[] args) {

try{

int a[]=new int[5];

a[5]=30/0;

}

catch(ArithmeticException e)

{

System.out.println("Arithmetic Exception occurs");

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("ArrayIndexOutOfBounds Exception occurs");

}

catch(Exception e)

{

System.out.println("Parent Exception occurs");

}

System.out.println("rest of the code");

}

}

**Q11. Write a java program to represent the Array List class.**

import java.io.\*;

import java.util.\*;

class ArrayListExample {

public static void main(String[] args)

{

int n = 5;

ArrayList<Integer> arr1 = new ArrayList<Integer>(n);

ArrayList<Integer> arr2 = new ArrayList<Integer>();

System.out.println("Array 1:" + arr1);

System.out.println("Array 2:" + arr2);

for (int i = 1; i <= n; i++) {

arr1.add(i);

arr2.add(i);

}

System.out.println("Array 1:" + arr1);

System.out.println("Array 2:" + arr2);

}

}

**Q12. Write a Java program that implements a multi-thread application that has**

**three threads.**

public class MultiThreadExample {

public static void main(String[] args) {

MyThread thread1 = new MyThread("Thread 1");

MyThread thread2 = new MyThread("Thread 2");

MyThread thread3 = new MyThread("Thread 3");

thread1.start();

thread2.start();

thread3.start();

}

}

class MyThread extends Thread {

private String threadName;

public MyThread(String name) {

this.threadName = name;

}

public void run() {

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

System.out.println(threadName + ": Count " + i);

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println(threadName + " has completed.");

}

}

**Q13. Write a Java program loads phone no, name from a text file using hash**

**table**

import java.util.ArrayList;

public class ArrayListExample {

public static void main(String[] args) {

ArrayList<String> nameList = new ArrayList<>();

nameList.add("Alice");

nameList.add("Bob");

nameList.add("Charlie");

nameList.add("David");

System.out.println("Names in the ArrayList:");

for (String name : nameList) {

System.out.println(name);

}

}

}

**Hashtable Example**

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.util.Hashtable;

public class LoadPhoneNumbers {

public static void main(String[] args) {

Hashtable<String, String> phoneBook = new Hashtable<>();

try (BufferedReader br = new BufferedReader(new FileReader("phone\_numbers.txt"))) {

String line;

while ((line = br.readLine()) != null) {

String[] parts = line.split(",");

if (parts.length == 2) {

String name = parts[0].trim();

String phoneNumber = parts[1].trim();

phoneBook.put(name, phoneNumber);

}

}

} catch (IOException e) {

e.printStackTrace();

}

System.out.println("Phone Numbers loaded from the file:");

for (String name : phoneBook.keySet()) {

String phoneNumber = phoneBook.get(name);

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

}

}

}

**Q.14 Write an applet program that displays a simple message.**

import java.applet.Applet;

import java.awt.Graphics;

public class SimpleMessageApplet extends Applet {

public void paint(Graphics g) {

g.drawString("Hello, this is a simple message from an applet!", 20, 20);

}

}

\*\*HTML

<html>

<head>

<title>Simple Message Applet</title>

</head>

<body>

<applet code="SimpleMessageApplet.class" width="300" height="50">

</applet>

</body>

</html>

**Q.7. Write a Java program compute factorial value using Applet.**

import java.applet.Applet;

import java.awt.Button;

import java.awt.Label;

import java.awt.TextField;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class FactorialApplet extends Applet implements ActionListener {

private TextField inputField;

private Label resultLabel;

public void init() {

Label inputLabel = new Label("Enter a number:");

inputField = new TextField(10);

resultLabel = new Label("Factorial will be displayed here");

Button calculateButton = new Button("Calculate");

calculateButton.addActionListener(this);

add(inputLabel);

add(inputField);

add(calculateButton);

add(resultLabel);

}

public void actionPerformed(ActionEvent e) {

try {

int number = Integer.parseInt(inputField.getText());

long factorial = calculateFactorial(number);

resultLabel.setText("Factorial of " + number + " is: " + factorial);

} catch (NumberFormatException ex) {

resultLabel.setText("Invalid input. Please enter a valid integer.");

}

}

private long calculateFactorial(int n) {

if (n < 0) {

return -1; // Negative numbers do not have a factorial

}

return (n == 0 || n == 1) ? 1 : n \* calculateFactorial(n - 1);

}

}

**\*\*HTML**

<html>

<head>

<title>Factorial Applet</title>

</head>

<body>

<applet code="FactorialApplet.class" width="300" height="150">

</applet>

</body>

</html>

**Q.8. Write a java program to create inner classes**

import java.util.\*;

class Outer {

private static void outerMethod()

{

System.out.println("inside outerMethod");

}

static class Inner {

public static void display()

{

System.out.println("inside inner class Method");

outerMethod();

}

}

}

class GFG {

public static void main(String args[])

{

Outer.Inner.display();

}

}