1. Try accessing a private, protected, public variable of a class into another class? Explain your inferences.
2. import java.lang.reflect.Field;  
     
   public class Main {  
    public static void main(String[] args) throws Exception {  
    MyClass obj = new MyClass();  
     
    int privateVariable=obj.getPrivateVariable();  
    System.*out*.println("private Variable= "+privateVariable);  
    int publicValue=obj.publicValue;  
    System.*out*.println("Public value= "+publicValue);  
    int protectedValue=obj.protectedValue;  
    System.*out*.println("protected Value= "+protectedValue);  
     
    }  
   }  
     
   class MyClass {  
    private int privateVariable = 42;  
     
    public int publicValue=52;  
    protected int protectedValue=62;  
     
    public int getPrivateVariable() {  
    return privateVariable;  
    }  
     
    public void setPrivateVariable(int privateVariable) {  
    this.privateVariable = privateVariable;  
    }  
   }

OUTPUT:

private Variable= 42

Public value= 52

protected Value= 62

2.Write a Java Program to extract numbers & special characters from a string given below using Regular Expression? Read this String from a text file and display the results in another text file.

import java.io.\*;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class SpecialCharacters {  
  
 public static void main(String[]args){  
 String inputFile = "D:/Projects/client/input.txt";  
 String outputFile = "D:/Projects/client/output.txt";  
  
 try {  
 String inputString = *readFromFile*(inputFile);  
 String result = *extractNumbersAndSpecialCharacters*(inputString);  
 *writeToFile*(outputFile, result);  
 System.*out*.println("Extraction successful. Results written to " + outputFile);  
 } catch (IOException e) {  
 System.*out*.println("An error occurred: " + e.getMessage());  
 }  
 }  
  
 private static String readFromFile(String fileName) throws IOException {  
 StringBuilder content = new StringBuilder();  
 BufferedReader reader = new BufferedReader(new FileReader(fileName));  
 String line;  
  
 while ((line = reader.readLine()) != null) {  
 content.append(line);  
 }  
  
 reader.close();  
 return content.toString();  
 }  
  
 private static String extractNumbersAndSpecialCharacters(String inputString) {  
 // Regular expression pattern to match numbers and special characters  
 String regex = "[0-9!@#$%^&\*()\\-=\_+\\[\\]{}|;':\",.<>/?]+";  
  
 StringBuilder result = new StringBuilder();  
 Pattern pattern = Pattern.*compile*(regex);  
 Matcher matcher = pattern.matcher(inputString);  
  
 while (matcher.find()) {  
 result.append(matcher.group()).append(" ");  
 }  
  
 return result.toString();  
 }  
  
 private static void writeToFile(String fileName, String content) throws IOException {  
 BufferedWriter writer = new BufferedWriter(new FileWriter(fileName));  
 writer.write(content);  
 writer.close();  
 }  
}

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Write a java program to explain Abstraction and Encapsulation
2. abstract class Vehicle {  
    private String model;  
    private int year;  
     
    public Vehicle(String model, int year) {  
    this.model = model;  
    this.year = year;  
    }  
     
    public abstract void start(); // Abstract method  
     
    public String getModel() {  
    return model;  
    }  
     
    public int getYear() {  
    return year;  
    }  
   }  
     
   // encapsulating implementation details within a class  
   class Car extends Vehicle {  
    private String color;  
     
    public Car(String model, int year, String color) {  
    super(model, year);  
    this.color = color;  
    }  
     
    @Override  
    public void start() {  
    System.*out*.println("Car engine started.");  
    }  
     
    public String getColor() {  
    return color;  
    }  
   }  
   public class AbstractionAndEncapsulation {  
     
    public static void main(String[] args) {  
    Car car = new Car("Honda", 2023, "Red");  
     
    System.*out*.println("Model: " + car.getModel());  
    System.*out*.println("Year: " + car.getYear());  
    System.*out*.println("Color: " + car.getColor());  
     
    car.start();  
    }}

output:

Model: Honda

Year: 2023

Color: Red

4.Write a Java program to demonstrate at least five String class operations.

public class Operations {  
 public static void main(String[] args) {  
 String str1 = "Hello, Purushotham";  
  
 String str2 = "Hello, ";  
 String str3 = "Purushotham....";  
 String concatenatedStr = str2.concat(str3);  
  
 int length = str1.length();  
  
 String substring = str1.substring(7);  
  
  
 String lowercase = str1.toLowerCase();  
 String uppercase = str1.toUpperCase();  
 System.*out*.println("First operation Original String: " + str1);  
 System.*out*.println("Second operation Concatenated String: " + concatenatedStr);  
 System.*out*.println("third operation Length of String 1: " + length);  
 System.*out*.println("Fourth operationSubstring: " + substring);  
 System.*out*.println("fifth operation Lowercase: " + lowercase);  
 System.*out*.println("Uppercase: " + uppercase);  
 }  
}

output:

First operation Original String: Hello, Purushotham

Second operation Concatenated String: Hello, Purushotham....

third operation Length of String 1: 18

Fourth operationSubstring: Purushotham

fifth operation Lowercase: hello, purushotham

Uppercase: HELLO, PURUSHOTHAM

1. Write a Java program to find Rank of the students based on total marks for different subject. Implement different classes for Rank logic and calculating subject total marks.
2. import java.util.\*;  
    class Student {  
    private String name;  
    private Map<String, Integer> subjectMarks;  
     
    public Student(String name) {  
    this.name = name;  
    this.subjectMarks = new HashMap<>();  
    }  
     
    public void setMarks(String subject, int marks) {  
    subjectMarks.put(subject, marks);  
    }  
     
    public int getTotalMarks() {  
    int total = 0;  
    for (int marks : subjectMarks.values()) {  
    total += marks;  
    }  
    return total;  
    }  
     
    public String getName() {  
    return name;  
    }  
    }  
     
    class RankCalculator {  
    public static List<Student> calculateRank(List<Student> students) {  
    List<Student> rankedStudents = new ArrayList<>(students);  
    rankedStudents.sort(Comparator.*comparingInt*(Student::getTotalMarks).reversed());  
    return rankedStudents;  
    }  
    }  
     
    public class StudentRank{  
    public static void main(String[] args) {  
    Student student1 = new Student("purushotham");  
    student1.setMarks("Maths", 85);  
    student1.setMarks("Science", 90);  
    student1.setMarks("English", 80);  
     
    Student student2 = new Student("madhu");  
    student2.setMarks("Maths", 75);  
    student2.setMarks("Science", 85);  
    student2.setMarks("English", 90);  
     
    Student student3 = new Student("ganga");  
    student3.setMarks("Maths", 95);  
    student3.setMarks("Science", 80);  
    student3.setMarks("English", 85);  
     
    List<Student> students = new ArrayList<>();  
    students.add(student1);  
    students.add(student2);  
    students.add(student3);  
     
    List<Student> rankedStudents = RankCalculator.*calculateRank*(students);  
     
    System.*out*.println("Rank\tStudent\tTotal");  
    for (int i = 0; i < rankedStudents.size(); i++) {  
    Student student = rankedStudents.get(i);  
    System.*out*.println((i + 1) + "\t" + student.getName() + "\t\t" + student.getTotalMarks());  
    }  
    }  
    }

output:

Rank Student Total

1 ganga 260

2 purushotham 255

3 madhu 250