**Core Java Assessment: Matthew Buckle A (1008610)**

***1.Try accessing a private, protected, public variable of a class into another class? Explain your inferences.***

Access Modifier in Java is used to limit the accessibility of the variables and methods inside the class.

**Private**: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.

**Protected**: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.

**Public**: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

**Example:**

**AccessModifierDemo.java**

**package** com.maveric;

**public** **class** AccessModifierDemo {

**public** **static** String *name* = "Matthew";

**private** **static** **long** *id* = 1008610;

**protected** **static** String *designation* = "Senior Software Engineer";

}

**Child1.java**

**package** com.Matthew;

**import** com.maveric.AccessModifierDemo;

**public** **class** Child1 **extends** AccessModifierDemo {

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

System.***out***.println(AccessModifierDemo.*name*);

System.***out***.println(AccessModifierDemo.*designation*);

}

}

**Child2.java**

*package com.Matthew;*

**import** com.maveric.AccessModifierDemo;

**public** **class** Child2 {

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

System.***out***.println(AccessModifierDemo.*name*);

}

}

***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.***

Input : wFvLrl2xQjyrWrxeNI21@#9w

**package** com.Matthew;

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**public** **class** StringExtract {

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

String inputFilePath = "C:/Users/matthewb/Desktop/Think Nxt/input.txt";

String outputFilePath = "C:/Users/matthewb/Desktop/Think Nxt/output.txt";

**try** {

BufferedReader reader = **new** BufferedReader(**new** FileReader(inputFilePath));

String inputString = reader.readLine();

reader.close();

String numbers = *extractNumbers*(inputString);

String specialCharacters = *extractSpecialCharacters*(inputString);

BufferedWriter writer = **new** BufferedWriter(**new** FileWriter(outputFilePath));

writer.write("Numbers: " + numbers + "\n");

writer.write("Special Characters: " + specialCharacters + "\n");

writer.close();

System.***out***.println("Extraction complete. Results written to " + outputFilePath);

} **catch** (IOException e) {

e.printStackTrace();

}

}

**private** **static** String extractNumbers(String inputString) {

Pattern pattern = Pattern.*compile*("\\d+");

Matcher matcher = pattern.matcher(inputString);

StringBuilder sb = **new** StringBuilder();

**while** (matcher.find()) {

sb.append(matcher.group()).append(" ");

}

**return** sb.toString().trim();

}

**private** **static** String extractSpecialCharacters(String inputString) {

Pattern pattern = Pattern.*compile*("[^a-zA-Z0-9\\s]+");

Matcher matcher = pattern.matcher(inputString);

StringBuilder sb = **new** StringBuilder();

**while** (matcher.find()) {

sb.append(matcher.group()).append(" ");

}

**return** sb.toString().trim();

}

}

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

***3. Write a java program to explain Abstraction and Encapsulation***

**Abstraction**: Abstraction is the process of hiding the implementation details and showing only functionality to the user

Bank.java

**package** com.maveric;

**public** **abstract** **class** Bank {

**private** **float** rateOfInterest;

**public** **float** getRateOfInterst () {

**return** rateOfInterest;

}

}

SBI.java

**package** com.maveric;

**public** **class** SBI **extends** Bank{

**public** **float** getRateOfInterst () {

**return** 7.1f;

}

}

HDFC.java

**package** com.maveric;

**public** **class** HDFC **extends** Bank {

**public** **float** getRateOfInterst () {

**return** 6.9f;

}

}

ICICI.java

**package** com.maveric;

**public** **class** ICICI **extends** Bank {

**public** **float** getRateOfInterst () {

**return** 6.8f;

}

}

BankApp.java

**package** com.maveric;

**public** **class** BankApp {

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

Bank B1 = **new** SBI();

Bank B2 = **new** HDFC();

Bank B3 = **new** ICICI();

System.***out***.println("SBI Bank Interest : "+ B1.getRateOfInterst());

System.***out***.println("HDFC Bank Interest : "+ B2.getRateOfInterst());

System.***out***.println("ICICI Bank Interest : "+ B3.getRateOfInterst());

}

}

A screenshot of a computer program

Description automatically generated with low confidence

**Encapsulation**: Wrapping up of data and code in a single unit using getters and setters

Here Student Class is a fully encapsulated class. It has private data member and getter and setter methods.

Student.java

**package** com.maveric;

**public** **class** Student {

**private** **int** rollNo;

**private** String name ;

**private** String email;

**private** **long** phoneNO;

//public getter and setter methods

**public** **int** getRollNo() {

**return** rollNo;

}

**public** **void** setRollNo(**int** rollNo) {

**this**.rollNo = rollNo;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** **long** getPhoneNO() {

**return** phoneNO;

}

**public** **void** setPhoneNO(**long** phoneNO) {

**this**.phoneNO = phoneNO;

}

}

Encapsulation.java

**package** com.maveric;

**public** **class** Encapsulation {

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

//creating instance of Student class

Student stud = **new** Student();

//setting values through setter methods

stud.setRollNo(58042);

stud.setName("Gowtham");

stud.setEmail("gowtham@gmail.com");

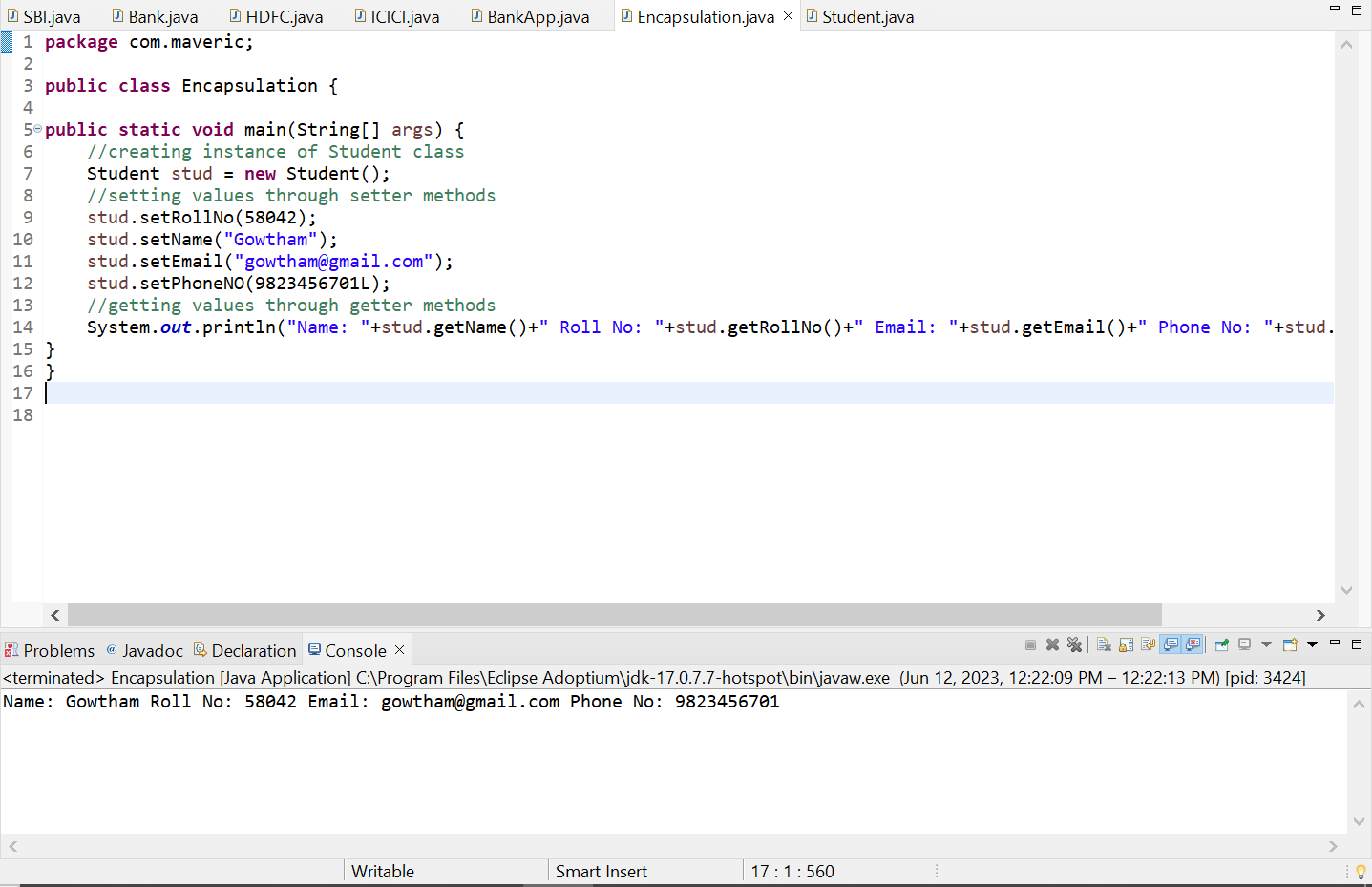
stud.setPhoneNO(9823456701L);

//getting values through getter methods

System.***out***.println("Name: "+stud.getName()+" Roll No: "+stud.getRollNo()+" Email: "+stud.getEmail()+" Phone No: "+stud.getPhoneNO());

}

}



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

**package** com.maveric;

**public** **class** StringDemo {

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

String str1 = "Matthew";

String str2 = " Buckle";

String str3 = "Maveric Systems";

System.***out***.println(str1.concat(str2));

System.***out***.println(str1.charAt(2));

System.***out***.println(str1.toUpperCase());

System.***out***.println(str1.contains(str2));;

System.***out***.println(str1.length());

System.***out***.println(str2.indexOf(5));

System.***out***.println(str3.substring(0,7));

}

}

A screenshot of a computer program

Description automatically generated with medium confidence

***5. 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.***

**package** com.Matthew;

**import** java.util.List;

**import** com.Matthew.Subject;

**public** **class** Student {

**private** **int** id;

**private** String name;

**private** List<Subject> subjects;

**public** Student(**int** id, String name, List<Subject> subjects) {

**this**.id = id;

**this**.name = name;

**this**.subjects = subjects;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** List<Subject> getSubjects() {

**return** subjects;

}

**public** **void** setSubjects(List<Subject> subjects) {

**this**.subjects = subjects;

}

**public** **int** getTotalMarks() {

**int** totalMarks = 0;

**for** (Subject subject : subjects) {

totalMarks = totalMarks + subject.getMarks();

}

**return** totalMarks;

}

}

**package** com.Matthew;

**public** **class** Subject {

**private** String name;

**private** **int** marks;

**public** Subject(String name, **int** marks) {

**this**.name = name;

**this**.marks = marks;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getMarks() {

**return** marks;

}

**public** **void** setMarks(**int** marks) {

**this**.marks = marks;

}

}

**package** com.maveric;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.List;

**import** com.Matthew.Student;

**public** **class** RankLogic {

**public** **static** **void** calculateRanks(List<Student> students) {

Collections.*sort*(students, Comparator.*comparingInt*(Student::getTotalMarks).reversed());

System.***out***.println("Rank\tName\t\tTotal Marks");

**int** rank = 1;

**for** (Student student : students) {

System.***out***.println(rank + "\t" + student.getName() + "\t\t" + student.getTotalMarks());

rank++;

}

}

}

**package** com.maveric;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**import** com.maveric.RankLogic;

**import** com.Matthew.Subject;

**public** **class** RankCalculator {

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

Subject subject1 = **new** Subject("Maths",80);

Subject subject2 = **new** Subject("Physics", 90);

Subject subject3 = **new** Subject("Chemistry", 95);

List<Subject> subject = Arrays.*asList*(subject1, subject2, subject3);

List<Student> students = **new** ArrayList<>();

students.add(**new** Student(1,"Gowtham", subject));

students.add(**new** Student(2,"Naveen", subject));

students.add(**new** Student(3,"Aravind", subject);

RankLogic.*calculateRanks*(students);

}

}