**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Computer Engineering Department**

Program: BTech Integrated, Semester IV

**Course: Java Programming**

**Experiment No.05**

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Aim:** To implement the concept of Packages and Interface.

**Prerequisite: concept of class, object and Inheritance**

**Outcome: After successful completion of this practical students will be able to**

1. Understand how to implement to concept of inheritance in program
2. Will be able to understand the concept of method overriding and abstract class.
3. Will be able to implement concept of packages and interface.

**Theory:**

**Packages**

* Packages are container for classes and interface and sub packages.
* Define a package
  + package pname;
  + example to create package called student write
  + package student;
  + top of all the java code
  + To import student package use
  + import student.\*;

**How to set classpath**

* + Suppose your package name is MyPack and it is located at C:\Documents and Settings\Krishna\Desktop
  + Classpath should be set till one level above MyPack folder.
  + To set class path type following on command prompt
  + set CLASSPATH=.; C:\Documents and Settings\Krishna\Desktop;
  + Note: We use . as an alias for the current directory. Java will find our class file not only from C: \Documents and Settings\Krishna\Desktop directory but from the current directory as well. Also, we use ; (semicolon) to separate the directory location in case we keep class files in many places.

**How to compile and run package program**

* If your package/folder is in j2sdk1.4.2\_13\bin then
* To compile
  + C:\j2sdk1.4.2\_13\bin>javac Packagename\filename.java
* To run
  + C:\j2sdk1.4.2\_13\bin>java Packagename.filename
  + Make sure that classpath is set properly.

**Access Protection**

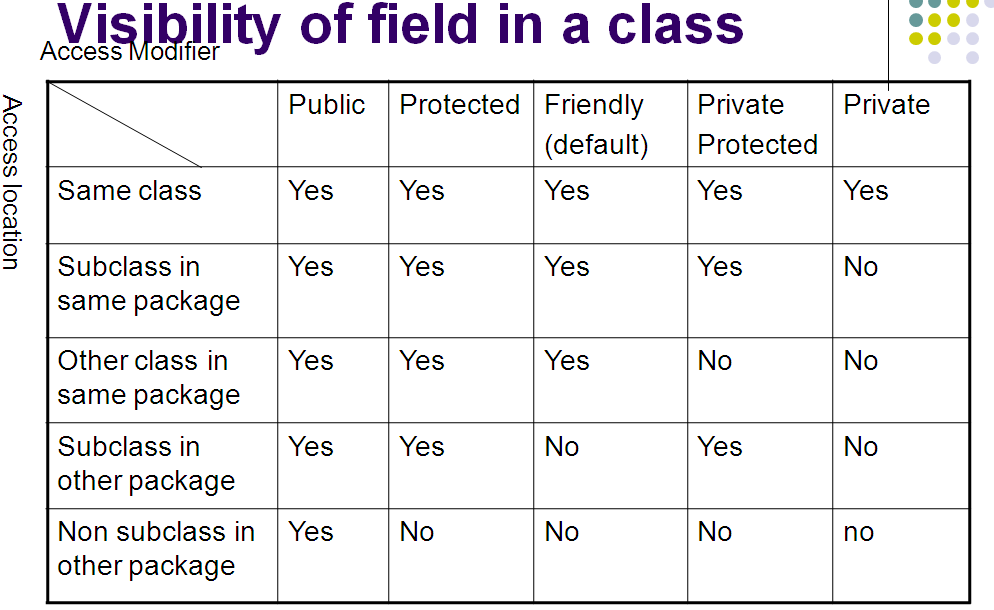
* **Java addresses four categories of visibility for class members:**

**■ Subclasses in the same package**

**■ Non-subclasses in the same package**

**■ Subclasses in different packages**

**■ Classes that are neither in the same package nor subclasses**

****

**Programs:**

* Create a class Employee having code, name as data members. Teacher class inherits Employee class and has number of lecture conducted, salary per lecture as data member. Include method calculate which calculates total salary on following condition: Number of lecture \* salary per lecture. Manager class inherits Employee class and has salary and unit of sales as data members. Include method calculate which calculates bonus on following condition: if unit of sales <100 then bonus= 10% of salary, else if unit of sales >=100 & <200, bonus=15% of salary, else if unit of sales >=200 then bonus= 20% of salary.
  + Create object of manager and teacher class and call respective calculate function. All the classes should have proper constructors and display method which prints respective details. Read values from the user.
* Use above inheritance program completed in lab 4 for this practical to implement the concept of packages.

Create a package person and it should contain two classes employee and teacher which extends the employee. (package should be outside jdk)

Create other package company and it should have class manager which extends employee class( package should be outside jdk) Set classpath properly.

Create a file packageDemo in outside both person and company packages and create object of teacher and manager and call calculate for teacher and calculate for manger also display all the details for both objects.

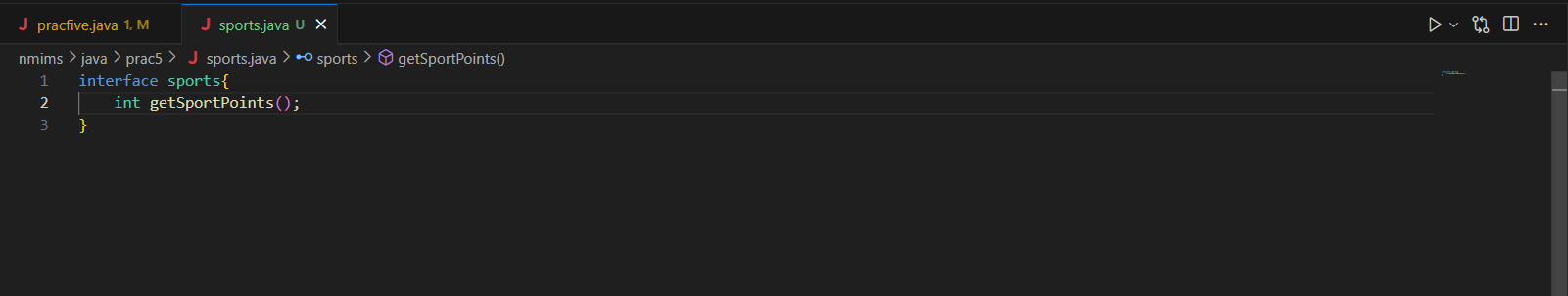
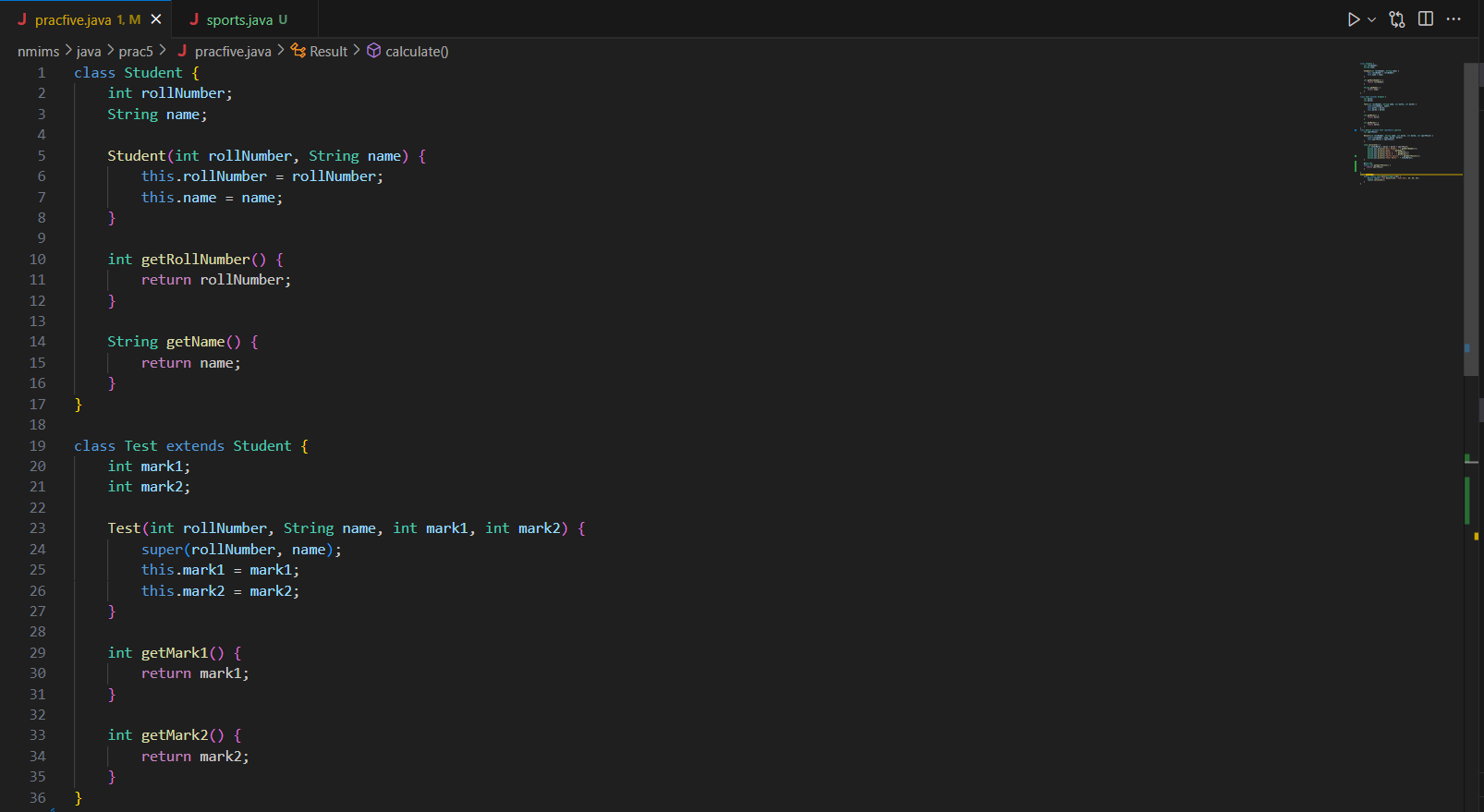
* Create class Student (roll number, name). Class Test (mark1, mark2) inherit student class. Create interface Sport (sport\_point) and method calculate(). Create class Result which extends Test and implements Sport and override the calculate method to find total\_marks as marks1+marks2+sport\_point. Calculate method should also print all the detail. Create an object of Result class in class containing main function and check it working.

PART B

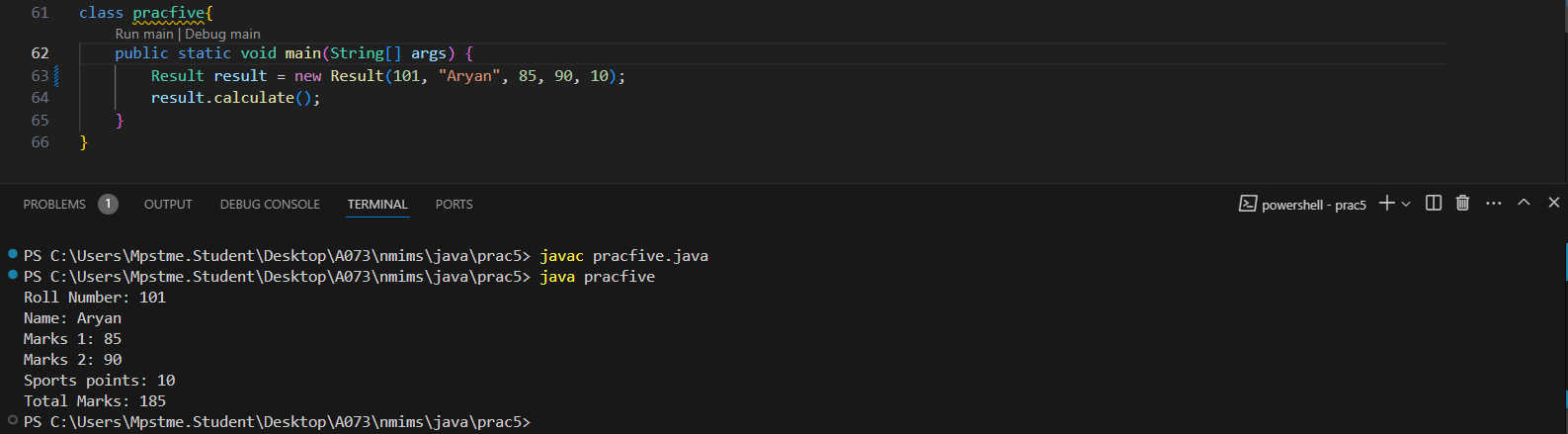
Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the portal at the end of the practical.

|  |  |
| --- | --- |
| Roll No. A073 | Name: Aryan Srivastava |
| Class : A | Batch : B3 |
| Date of Experiment: | Date of Submission |
| Grade |  |

**B.1 Software Code written by student:**

****

**B.2 Input and Output:**

****

**B.3 Question of Curiosity:**

1. State advantages of using packages.

**Organization:** Groups related classes and interfaces together, making the codebase more manageable.

**Namespace Management:** Helps avoid name conflicts by providing a unique namespace.

**Access Control:** Provides access control with different visibility levels (e.g., public, protected, default, private).

**Reusability:** Promotes reuse of code across different projects.

1. Explain how multiple inheritance is implemented in java using Interfaces.

Java supports multiple inheritance through interfaces by allowing a class to implement multiple interfaces. A class can implement several interfaces, thereby inheriting the abstract methods from all of them. For example:

interface A {

void methodA();

}

interface B {

void methodB();

}

class C implements A, B {

public void methodA() {

// implementation

}

public void methodB() {

// implementation

}

}

1. Differentiate between abstract class and interface

**Abstract Class:**

* Can have both abstract and concrete methods.
* Can have instance variables with access modifiers.
* Supports constructors.
* A class can extend only one abstract class due to single inheritance.

**Interface:**

* Can only have abstract methods (Java 8+ allows default and static methods).
* Cannot have instance variables (only constants, which are public static final).
* Cannot have constructors.
* A class can implement multiple interfaces.

**B.4 Conclusion:**

**Learnt about packages and interfaces.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*