**Batch: A3 Roll No.: 16010122074**

**Experiment / assignment / tutorial No. 9**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **TITLE :Java Packages** |

**AIM:** Create a **Package Engineering** which has two classes as **Student and Marks**. Accept (n) student details like roll\_no, Subject\_name, Student\_name,calculate total marks in the class Student Write **display () method** to display details and **sort () method** to sort the students records as per increasing order of the total marks. The function **sort must be statically defined to invoke it without referring to any object**. Both the functions are written in the Marks class.

Create a main class which will use a package to display all the records of the student in the increasing order of their total marks.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**CO4:** Explore the interface, exceptions, multithreading, packages.

**Books/ Journals/ Websites referred:**

1. Ralph Bravaco , Shai Simoson , “Java Programming From the Group Up” Tata McGraw-Hill.

2.Grady Booch, Object Oriented Analysis and Design .

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

**Java Packages:**

A package in Java is a group of similar types of classes, interfaces, and sub-packages. They can be categorized into two categories, the built-in package ( java, lang, util, awt, javax, swing, net, io, sql et), and user-defined package.

They are used for the following tasks –

* To prevent the naming conflicts which can occur between the classes.
* Make the searching and locating of classes or enumerations or annotations much easier.
* Provide access control to the classes.
* Used for data encapsulation.

**Advantages of Java Package:**

* A Java package is mainly used for the categorization of classes and interfaces so that we can maintain them easily.
* They always provide access protection
* Used to bundle classes and interfaces.
* With the help of packages, we can reuse the existing code
* By using the package, we can easily locate the classes related to it.
* Also, remove the naming collision.

**Built-in Packages in Java**

Built-in is a part of Java API and it offers a variety of packages are –

lang – Automatically imported and it contains language support classes.

io – Contains classes for input and output operations.

util – Contains utility classes for implementing data structures.

applet – This package contains classes that create applets.

awt – Contain classes that implement compounds for GUI.

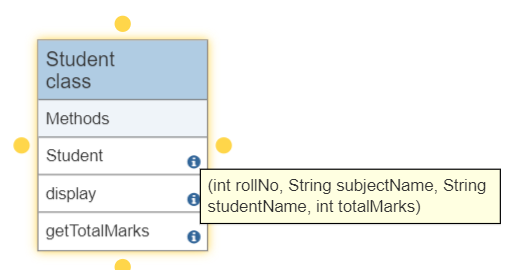
net – This package contains classes that support networking operations.

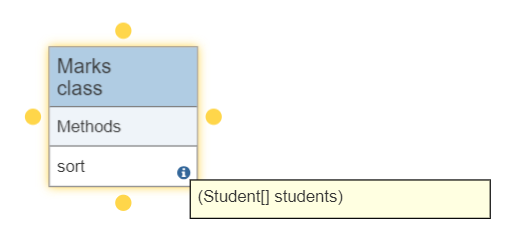
**User-defined Packages in Java**

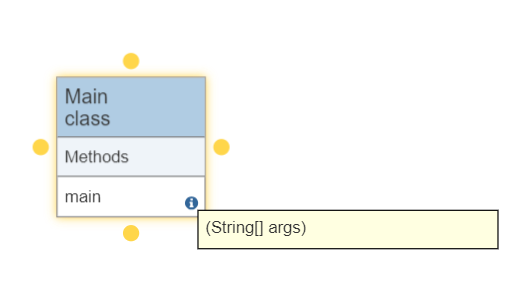
1. package First;
2. public class MyClass
3. {
4. public void **getNames**(String name)
5. {
6. System.out.**println**(name);
7. }
8. }
9. package First;
10. import First.MyClass;
11. public class MyClass1 {
12. public static void **main**(String args[])
13. {
14. // Initializing the String variable with a value
15. String name = "Welcome";
16. // Creating an instance of class MyClass in the package.
17. MyClass obj = new **MyClass**();
18. obj.**getNames**(name);
19. }
20. }

.

**Class Diagram:**







**Algorithm:**

1. Start the program.
2. Create a Scanner object to read user input.
3. Prompt the user to enter the number of students.
4. Create an array of Student objects based on the user-provided number of students.
5. For each student in the array:

* Prompt the user to enter the student's details (Roll No, Subject Name, Student Name, and Total Marks).
* Create a Student object with the provided details and store it in the array.

1. Display the original student records.
2. Use the Marks.sort method to sort the students based on their total marks in ascending order.
3. Display the sorted student records.
4. Close the Scanner to release system resources.
5. End the program.

**Implementation details:**

**STUDENT CLASS**

package PackageEngineering;

public class Student {

    private int rollNo;

    private String subjectName;

    private String studentName;

    private int totalMarks;

    public Student(int rollNo, String subjectName, String studentName, int totalMarks) {

        this.rollNo = rollNo;

        this.subjectName = subjectName;

        this.studentName = studentName;

        this.totalMarks = totalMarks;

    }

    public void display() {

        System.out.println("Roll No: " + rollNo);

        System.out.println("Subject Name: " + subjectName);

        System.out.println("Student Name: " + studentName);

        System.out.println("Total Marks: " + totalMarks);

        System.out.println("--------------------");

    }

    public int getTotalMarks() {

        return totalMarks;

    }

}

**MARKS CLASS**

package PackageEngineering;

import java.util.Arrays;

public class Marks {

    public static void sort(Student[] students) {

        Arrays.sort(students, (a, b) -> Integer.compare(a.getTotalMarks(), b.getTotalMarks()));

    }

}

**MAIN CLASS**

import PackageEngineering.Student;

import PackageEngineering.Marks;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of students: ");

        int numStudents = scanner.nextInt();

        scanner.nextLine(); // Consume the newline character

        Student[] students = new Student[numStudents];

        for (int i = 0; i < numStudents; i++) {

            System.out.println("Enter details for Student " + (i + 1) + ":");

            System.out.print("Roll No: ");

            int rollNo = scanner.nextInt();

            scanner.nextLine(); // Consume the newline character

            System.out.print("Student Name: ");

            String studentName = scanner.nextLine();

            System.out.print("Subject Name: ");

            String subjectName = scanner.nextLine();

            System.out.print("Total Marks: ");

            int totalMarks = scanner.nextInt();

            scanner.nextLine(); // Consume the newline character

            students[i] = new Student(rollNo, subjectName, studentName, totalMarks);

        }

        System.out.println("Original Student Records:");

        for (Student student : students) {

            student.display();

        }

        Marks.sort(students);

        System.out.println("Student Records Sorted by Total Marks:");

        for (Student student : students) {

            student.display();

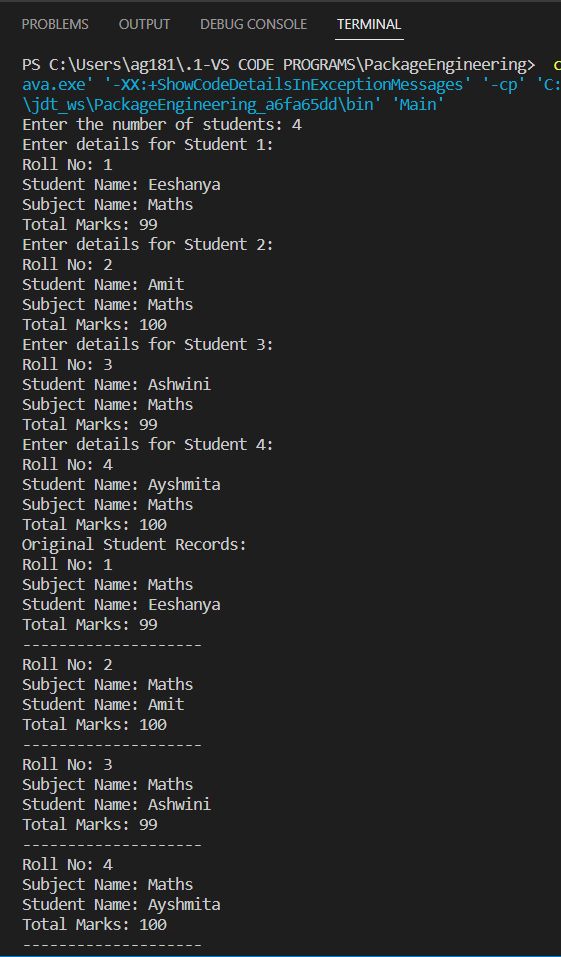
        }

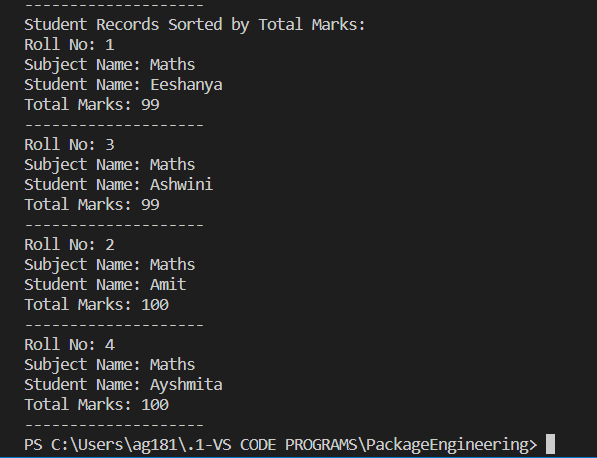
    scanner.close();

    }

}

**Output:**





**Conclusion:**

Through this experiment we learnt the concept of packages and implemented them using the Java language. We have successfully used user defined package Engineering to calculate, sort and display the total marks of the students and their other details. Packages helps us club classes into an entity in programs.

**Date: \_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

**Post Lab Descriptive Questions**

**Q.1   What are Java Packages? What's the significance of packages?**

A java package is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two form, built-in package and user-defined package.

Following are the advantages of using packages in Java −

* + Programmers can define their own packages to bundle a group of classes/interfaces, etc.
  + It is a good practice to group related classes implemented by you so that a programmer can easily determine that the classes, interfaces, enumerations, and annotations are related.
  + Since the package creates a new namespace there won't be any name conflicts with names in other packages.
  + Using packages, it is easier to provide access control
  + It is also easier to locate the related classes.

**Q.2 Does Importing a package imports its sub-packages as well in Java?**

No, you will have to import the sub-packages explicitly. An import declaration which ends with a package name and .\* imports all public classes/interfaces/annotations of the package only and nothing else. If you need classes from the child package too, you have to import them separately. If an import declaration contains a specific class, only that class will be imported. If the import declaration contains a specific class plus .\* (it's a static import, must be in the form of import static…), then all static fields and methods will be imported from that class and nothing more.