

Course Name:	Object Oriented Programming Methodology	Semester:	III
Date of Performance:	20 / 10 / 2023	Batch No:	B2
Faculty Name:	Prof.	Roll No:	16010122151
Faculty Sign & Date:		Grade/Marks:	___/25

Experiment No: 9
Title: Java Packages

Aim and Objective of the Experiment:
<p>Create a Package Engineering which has two classes as Student and Marks.</p> <p>Accept (n) student details like roll_no, Subject_name, Student_name, calculate total marks in the class Student.</p> <p>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.</p> <p>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.</p>

COs to be achieved:
CO4: Explore the interface, exceptions, multithreading, packages.

Tools used:
JDK, VScode / Eclipse

Theory:
<p>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.</p> <p>They are used for the following tasks –</p> <ul style="list-style-type: none"> • 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.

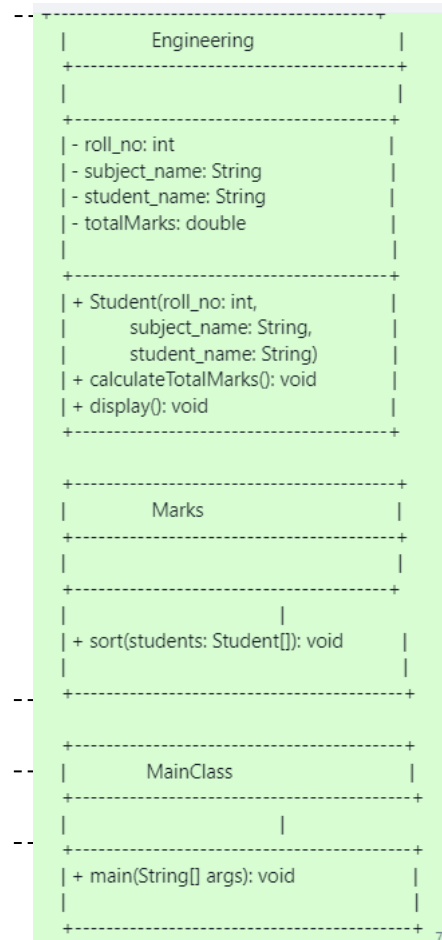
net – This package contains classes that support networking operations.

User-defined Packages in Java

```
1. package First;
2.
3. public class MyClass
4. {
5.     public void getNames(String name)
6.     {
7.         System.out.println(name);
8.     }
9.
10. }
```

```
1. package First;
2. import First.MyClass;
3. public class MyClass1 {
4.     public static void main(String args[])
5.     {
6.         // Initializing the String variable with a value
7.         String name = "Welcome";
8.         // Creating an instance of class MyClass in the package.
9.         MyClass obj = new MyClass();
10.     obj.getNames(name);
11. }
12. }
```

Class Diagram:



Algorithm for Student and Marks Classes:

1. Create a package named Engineering.
2. Define the Student class inside the Engineering package:
 - Define instance variables for student details: rollNo, subjectName, studentName, and totalMarks.
 - Create a parameterized constructor to initialize these variables.
 - Create a method calculateTotalMarks() to calculate the total marks (assuming you have the subject-wise marks).
 - Create a display() method to display the student's details.

Define the Marks class inside the Engineering package:

- Create a method sort(students: Student[]) to sort the students based on their total marks using a sorting algorithm (e.g., Bubble Sort).

4. Outside the Engineering package (in a separate class):

- Accept the number of students n and their details (i.e., rollNo, subjectName, studentName) from the user.
- Create an array of Student objects to store these details.
- Calculate the total marks for each student using the calculateTotalMarks() method.

Code:

```
1 // Student.java (inside Engineering package)
2 package Engineering;
3
4 import java.util.Scanner;
5
6 public class Student {
7     int rollNo;
8     String subjectName;
9     String studentName;
10    double marks1, marks2, marks3;
11    double totalMarks;
12
13    public void getDetails() {
14        Scanner input = new Scanner(System.in);
15        System.out.print("Enter Roll Number: ");
16        rollNo = input.nextInt();
17        input.nextLine(); // Consume the newline character
18        System.out.print("Enter Subject Name: ");
19        subjectName = input.nextLine();
20        System.out.print("Enter Student Name: ");
21        studentName = input.nextLine();
22        System.out.print("Enter Marks for Subject 1: ");
23        marks1 = input.nextDouble();
24        System.out.print("Enter Marks for Subject 2: ");
25        marks2 = input.nextDouble();
26        System.out.print("Enter Marks for Subject 3: ");
27        marks3 = input.nextDouble();
28        totalMarks = marks1 + marks2 + marks3;
29    }
30
31    public void display() {
32        System.out.println("Roll Number: " + rollNo);
33        System.out.println("Subject Name: " + subjectName);
34        System.out.println("Student Name: " + studentName);
35        System.out.println("Total Marks: " + totalMarks);
36    }
37 }
```

```
// Marks.java (inside Engineering package)
package Engineering;

public class Marks {
    public static void sort(Student[] students) {
        for (int i = 0; i < students.length - 1; i++) {
            for (int j = i + 1; j < students.length; j++) {
                if (students[i].totalMarks > students[j].totalMarks) {
                    Student temp = students[i];
                    students[i] = students[j];
                    students[j] = temp;
                }
            }
        }
    }
}

// MainClass.java
import Engineering.Student;
import Engineering.Marks;

public class MainClass {
    public static void main(String[] args) {
        int n = 3; // Number of students
        Student[] students = new Student[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            System.out.println("Enter Student " + (i + 1) + " details:");
            students[i].getDetails();
        }

        Marks.sort(students); // Sort students by total marks

        System.out.println("\nStudent Records (Sorted by Total Marks):");
        for (int i = 0; i < n; i++) {
            students[i].display();
            System.out.println();
        }
    }
}
```

Output:

```
Enter Student 1 details:
Enter Roll Number: 101
Enter Subject Name: Mathematics
Enter Student Name: Alice
Enter Marks for Subject 1: 95
Enter Marks for Subject 2: 88
Enter Marks for Subject 3: 92
```

```
Enter Student 2 details:
Enter Roll Number: 102
Enter Subject Name: Science
Enter Student Name: Bob
Enter Marks for Subject 1: 85
Enter Marks for Subject 2: 92
Enter Marks for Subject 3: 88
```

```
Enter Student 3 details:
Enter Roll Number: 103
Enter Subject Name: History
```

```
Enter Student 3 details:
Enter Roll Number: 103
Enter Subject Name: History
Enter Student Name: Carol
Enter Marks for Subject 1: 78
Enter Marks for Subject 2: 84
Enter Marks for Subject 3: 90
```

Student Records (Sorted by Total Marks):

```
Roll Number: 103
Subject Name: History
Student Name: Carol
Total Marks: 252
```

```
Roll Number: 102
Subject Name: Science
Student Name: Bob
Total Marks: 265
```

```
Roll Number: 101
Subject Name: Mathematics
```

Roll Number: 101
Subject Name: Mathematics
Student Name: Alice
Total Marks: 275

Post Lab Subjective/Objective type Questions:

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

In Java, a package is a way to organize and group related classes and interfaces. A package is a namespace that contains a collection of classes, interfaces, and sub-packages. It provides a way to modularize your code, making it easier to manage and maintain large projects. Here's the significance and key points regarding Java packages:

1. **Organization and Modularity:** Packages provide a way to organize your classes and other code elements. They allow you to group related classes together, making it easier to locate and

manage the components of your application. This helps in maintaining the modularity of your code.

2. **Name Clashes:** Packages prevent naming conflicts. For example, you can have a class named **Employee** in one package and another class with the same name in a different package without any conflict. This is because the fully qualified name of a class includes the package name.
3. **Access Control:** Java uses access modifiers like **public**, **protected**, and **private** to control access to classes and their members. Packages add another level of control. Classes marked as "package-private" can be accessed only by other classes in the same package, which restricts external access.
4. **Reusability:** You can reuse classes from other packages by importing them. This promotes code reuse and reduces redundancy. Java's standard library, for example, uses packages for various APIs that can be reused across different applications.
5. **Security:** Packages provide security by restricting access to certain classes and methods. Classes marked as **private** can't be accessed from outside the class, and classes marked as **public** are accessible to all other classes.
6. **Java Standard Library:** Java itself is organized using packages. The standard Java library is divided into various packages like **java.util**, **java.io**, **java.net**, etc. This modular organization simplifies the understanding and use of the library.
7. **Third-Party Libraries:** When you work with third-party libraries or frameworks, they are typically packaged. Packages help you easily integrate and use external code components.
8. **Scalability:** For larger applications, packages make it easier to scale your codebase. You can organize classes into packages according to functionality or layers (e.g., data access, business logic, user interface), making your codebase more maintainable.
9. **Readability:** Well-structured packages enhance code readability. Developers can quickly grasp the structure of your application, making collaboration more efficient.
10. **Java Naming Conventions:** Java packages should follow naming conventions, which include using lowercase letters for package names to distinguish them from class and interface names.

Packages are a fundamental concept in Java, and they help in managing the complexity of Java applications, ensuring code organization, and supporting reusability and maintainability.

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

No, importing a package in Java does not import its sub-packages automatically. When you import a package, it makes only the classes and interfaces within that specific package available for use in your code. Sub-packages need to be imported separately if you want to use their classes and interfaces.

For example, let's say you have a package structure like this:

```
myPackage
|-- subPackage1
| |-- ClassA
|-- subPackage2
| |-- ClassB
```

If you want to use **ClassA** from **subPackage1**, you need to import it explicitly in your Java code

like this:

```
import myPackage.subPackage1.ClassA;
```

Importing **myPackage** alone will not provide access to classes in **subPackage1** or **subPackage2**.

You need to import each sub-package or class individually when necessary.

However, there is a shortcut to import all the classes in a specific package, and that is to use a wildcard *. For example:

```
import myPackage.subPackage1.*;
```

This imports all classes in **subPackage1**. But keep in mind that it still doesn't import classes from other sub-packages, like **subPackage2**.

Conclusion:

We learnt the applications of Java Packages.

Signature of faculty in-charge with Date: