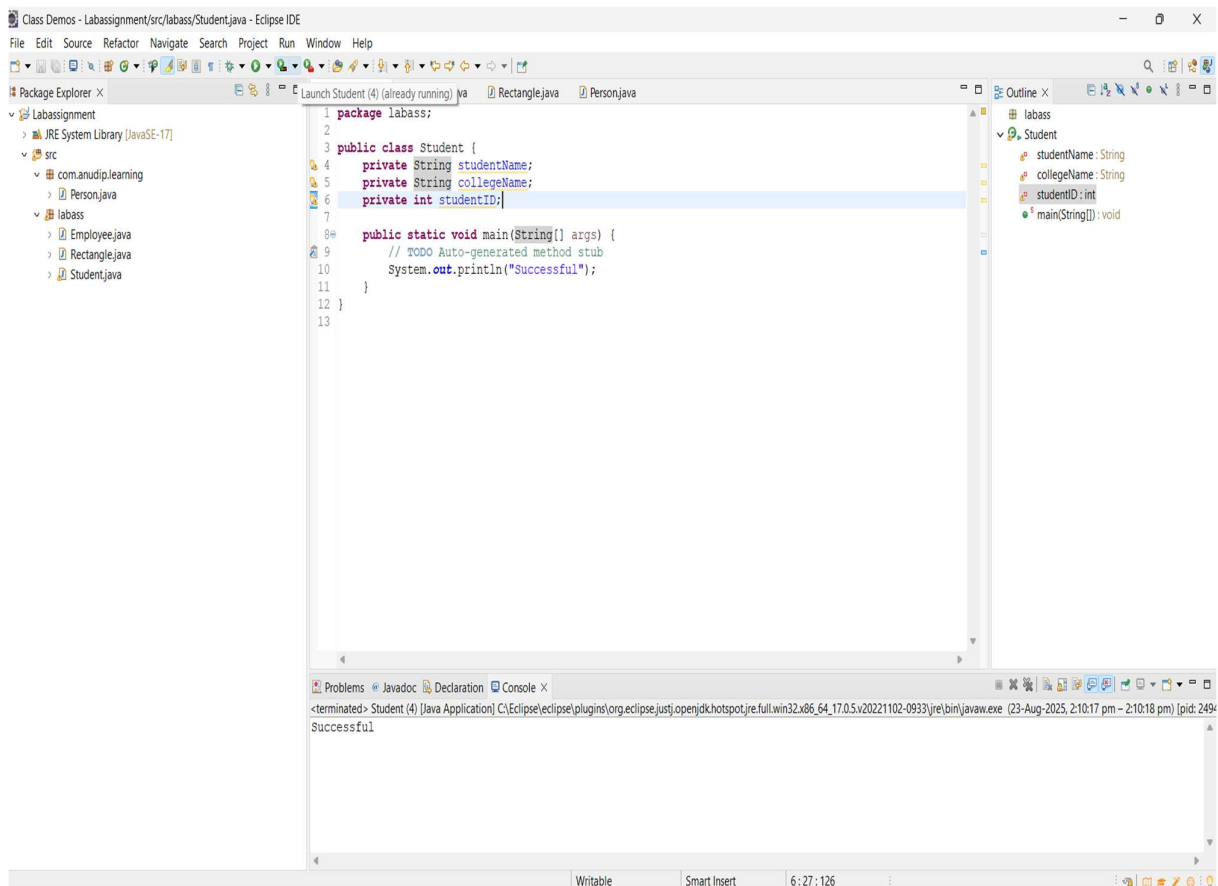


Assignment

Assignment 1.

- Create a class Student in Student.java then add member variables studentName, collegeName of type String
- Add a member variable studentID of type int.
- Make all the member variables as private.
- Add a main method. And print a message "Successful".
- Compile the class
- Run the class



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows the project structure: Labassignment > src > labass > Student.java. The main editor displays the code for Student.java:

```
1 package labass;
2
3 public class Student {
4     private String studentName;
5     private String collegeName;
6     private int studentID;
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10        System.out.println("Successful");
11    }
12 }
13
```

The Outline view on the right shows the class structure: Student with attributes studentName: String, collegeName: String, studentID: int, and a main method.

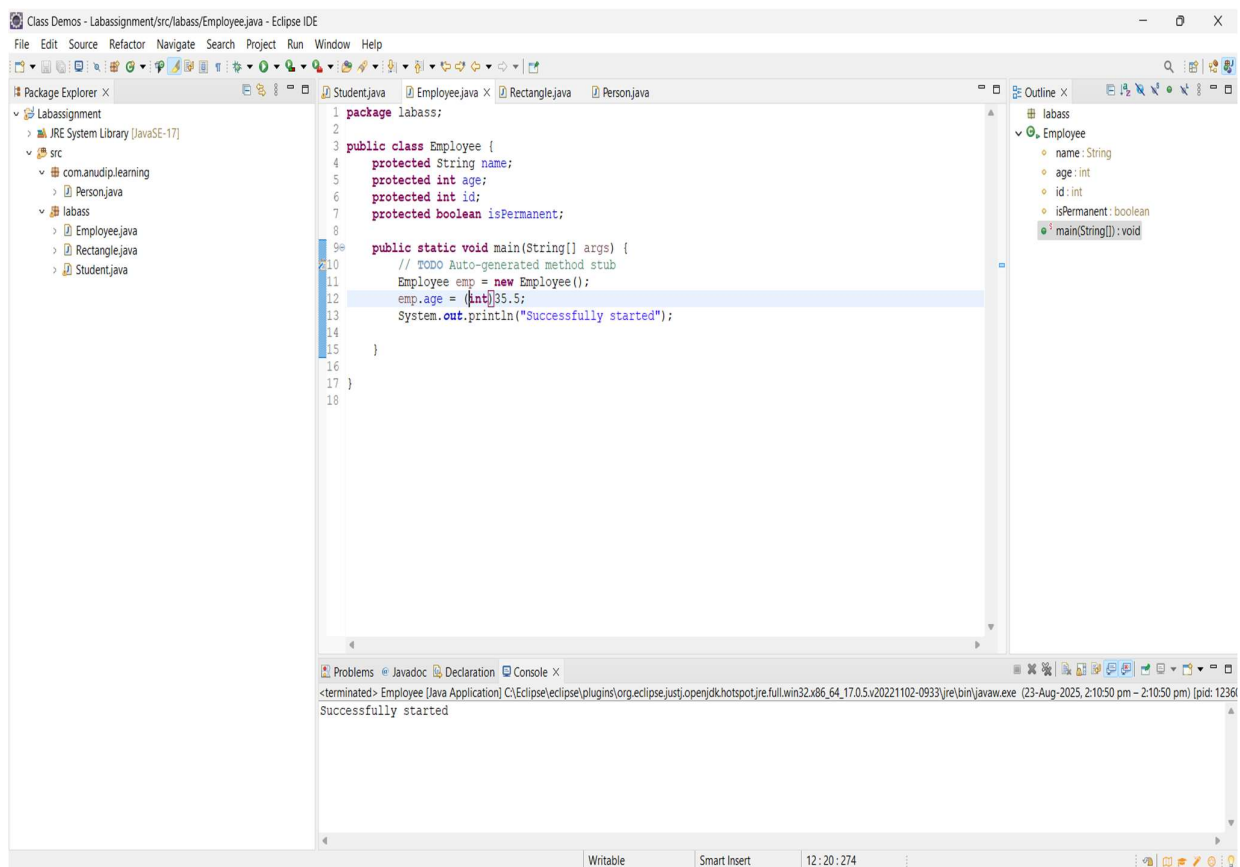
The Console view at the bottom shows the output of the program:

```
<terminated> Student (4) [Java Application] C:\Eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.5.v20221102-0933\jre\bin\javaw.exe (23-Aug-2025, 2:10:17 pm - 2:10:18 pm) [pid: 2494]
Successful
```

The status bar at the bottom indicates the file is writable, has smart insert enabled, and the cursor is at line 6, column 27.

Assignment-2.

- Create a new class Employee
- Add member variables: id and age of type int, name of type String and isPermanent of type boolean
- Now assign values 35.5 to age; See the error message.
- How can you avoid this error? Correct the error by casting.
- Make all the members protected
- Add a main method to it. Print message “Successfully started”.
- Compile the class.



Assignment-3.

- Create a class Person
- Add member variables name as String, age and salary as int
- Initialize the member variable along with declaration.
- Now put the previous Person class in a package com.anudip.learning
- Add a main method. Add a print message "Test Successful".
- Run the class after compilation.
- Modify the classpaths to see the error messages on the console.

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows the project structure: Labassignment > src > com.anudip.learning > Person.java. The main editor displays the code for Person.java:

```
1 package com.anudip.learning;
2
3 public class Person {
4     String Name = "Sravani";
5     int age = 21;
6     int salary = 50000;
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10        System.out.println("Test Successful");
11    }
12 }
13
14 }
15
```

The Outline view on the right shows the class structure: com.anudip.learning > Person, with members Name: String, age: int, salary: int, and main(String[]): void.

The Console view at the bottom shows the output of the program:

```
<terminated> Person (1) [Java Application] C:\Eclipse\ eclipse\plugins\org.eclipse.justi.openjdk hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (23-Aug-2025, 2:26:40 pm - 2:26:40 pm) [pid: 2192]
Test Successful
```

Assignment-4.

- Create a class Rectangle
- Add a member variable width and height of type double.
- Create an enum Color with values RED, GREEN, BLUE
- Create a member variable boxColor of type Color.
- Add a main method.
- In main method just print the enum Color.BLUE (You will notice that Java prints the enum name as it is.)
- Compile and run the class.

