Experiment Number : 01 **Date: 22.01.26**

Installation of Java and Simple Java Programs

Aim:

To install Java Development Kit (JDK), configure the environment, and write simple Java programs including Hello World.

PRE LAB EXERCISE

QUESTIONS

1. **What is JDK and why is it required?**
   The JDK (Java Development Kit) is a complete software package used for developing Java applications. It contains all the tools required to write, compile, debug, and run Java programs, including the compiler (javac), the Java Runtime Environment (JRE), the Java Virtual Machine (JVM), and other development utilities. The JDK is required because without it, a programmer cannot compile source code into bytecode or develop Java applications. It provides the full environment needed for Java program development.
2. **Difference between JDK, JRE, and JVM.**
   The JVM (Java Virtual Machine) is the core component responsible for executing Java bytecode and providing platform independence. The JRE (Java Runtime Environment) consists of the JVM along with core class libraries and supporting files required to run Java programs, but it does not include development tools. The JDK (Java Development Kit) includes the JRE plus development tools such as the compiler and debugger, making it suitable for creating and compiling Java programs. In short, JVM runs the code, JRE provides the environment to run it, and JDK provides the environment to develop it.
3. **What is the purpose of the main() method in Java?**
   The main() method is the entry point of a Java application. When a Java program is executed, the JVM first looks for the public static void main(String[] args) method and starts program execution from there. It is defined as static so that it can be called without creating an object, public so that it is accessible to the JVM, and void because it does not return any value. Without the main() method, a normal Java program cannot start execution.

IN LAB EXERCISE

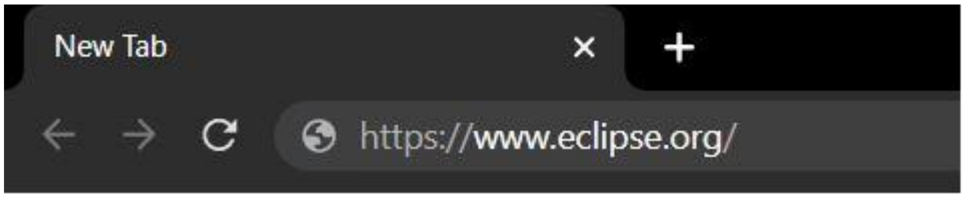
Objective:

To verify Java installation and execute a basic Java program.

INSTALLATION STEPS:

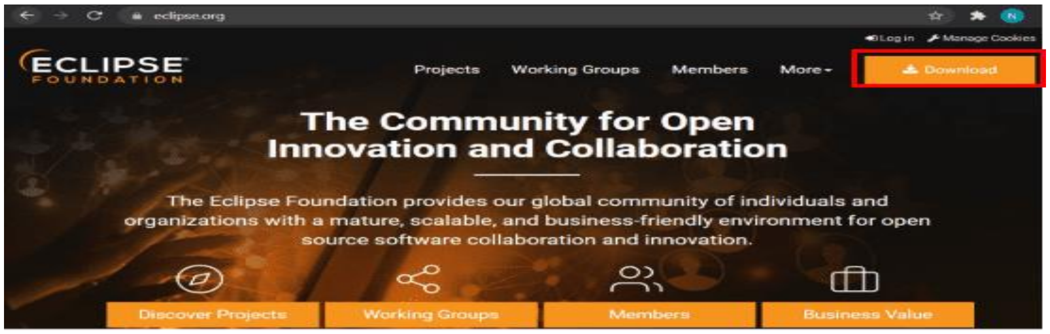
STEP 1: Open Browser

- Open your browser and go to the official URL Eclipse Downloads page.



STEP 2: Download Eclipse Installer

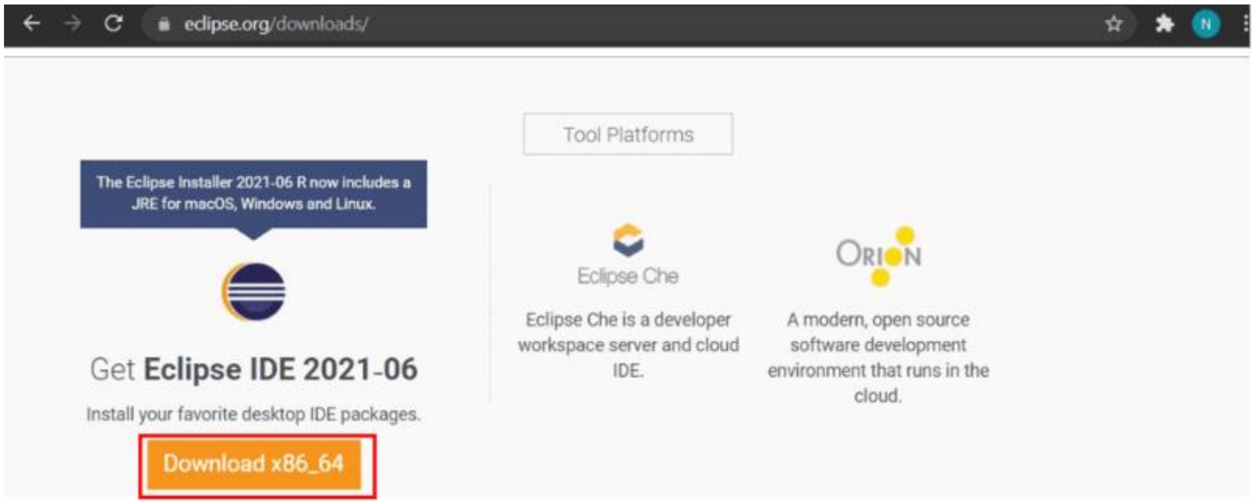
- Then, click on the "Download" button to download Eclipse IDE.



The Eclipse Foundation provides our global community of individuals and organizations with a mature, scalable, and business-friendly environment for open source software collaboration and innovation.

STEP 3: Download EXE

- Now, click on the "Download x86_64" button.



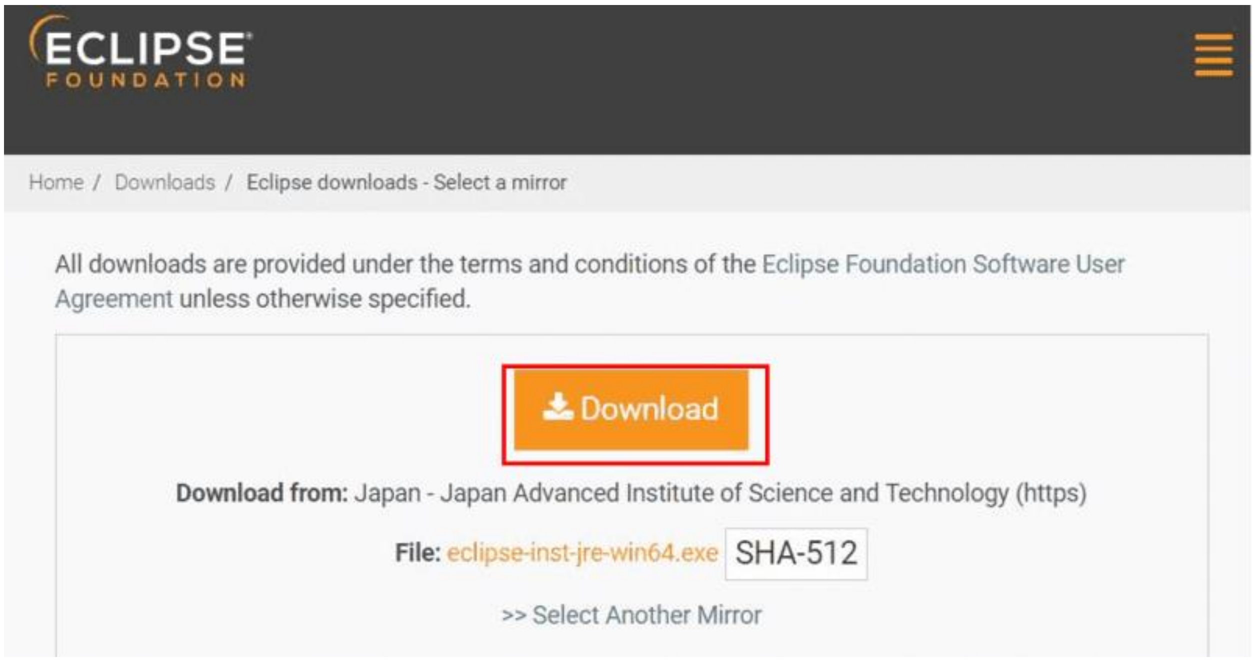
The Eclipse Installer 2021-06 R now includes a JRE for macOS, Windows and Linux.

Install your favorite desktop IDE packages.

Eclipse Che is a developer workspace server and cloud IDE.

A modern, open source software development environment that runs in the cloud.

STEP 4: Then click on the "Download" button. After clicking on the download button the .exe file for the eclipse will be downloaded.



Home / Downloads / Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

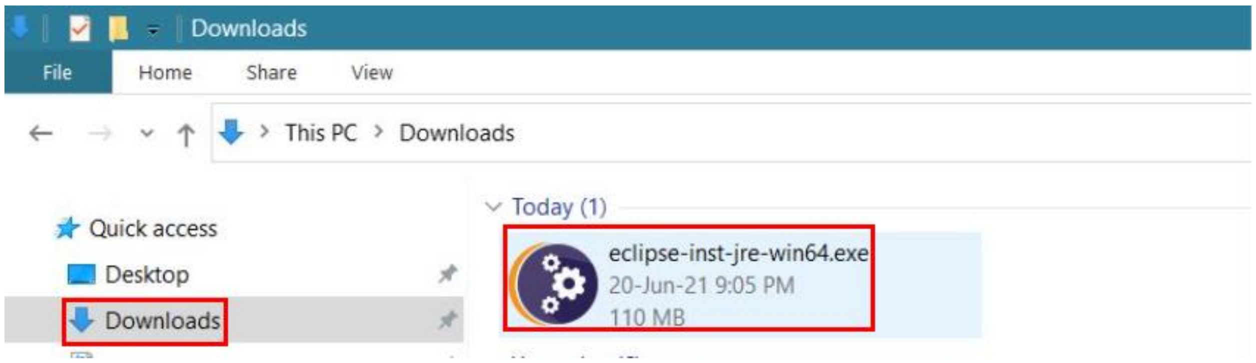
Download from: Japan - Japan Advanced Institute of Science and Technology (https)

File: eclipse-inst-jre-win64.exe SHA-512

>> Select Another Mirror

STEP 5: Open Download EXE

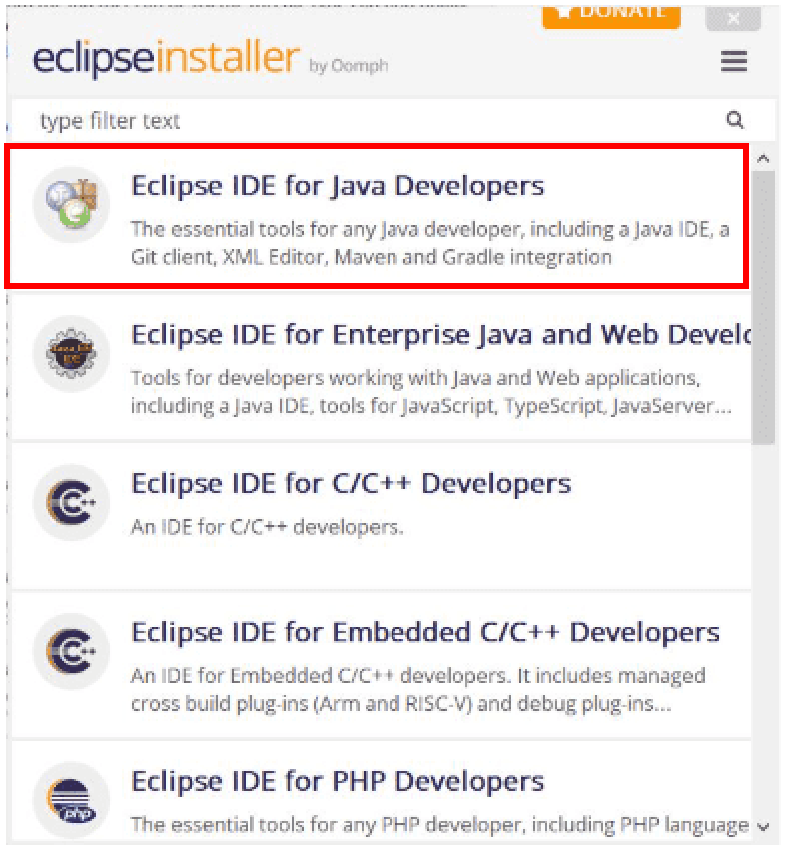
- Now go to File Explorer and click on "Downloads" after that click on the "*eclipse-inst-jre-win64.exe*" file for installing Eclipse IDE**.**



eclipse-inst-jre-win64.exe
20-Jun-21 9:05 PM
110 MB

STEP 6: Install Eclipse

- Then, click on "Eclipse IDE for Java Developers".



Eclipse IDE for Java Developers
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration

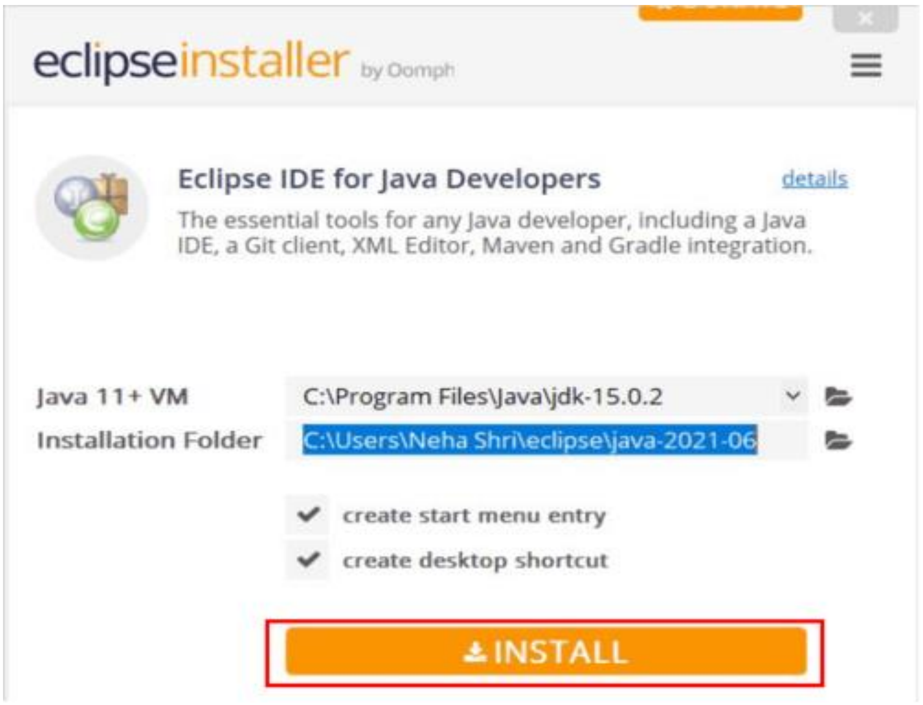
Eclipse IDE for Enterprise Java and Web Develc
Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer...

Eclipse IDE for C/C++ Developers
An IDE for C/C++ developers.

Eclipse IDE for Embedded C/C++ Developers
An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC-V) and debug plug-ins...

Eclipse IDE for PHP Developers
The essential tools for any PHP developer, including PHP language

STEP 7: Then, click on the "Install" button.


eclipseinstaller by Oomph

Eclipse IDE for Java Developers details

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.

Java 11+ VM: C:\Program Files\Java\jdk-15.0.2

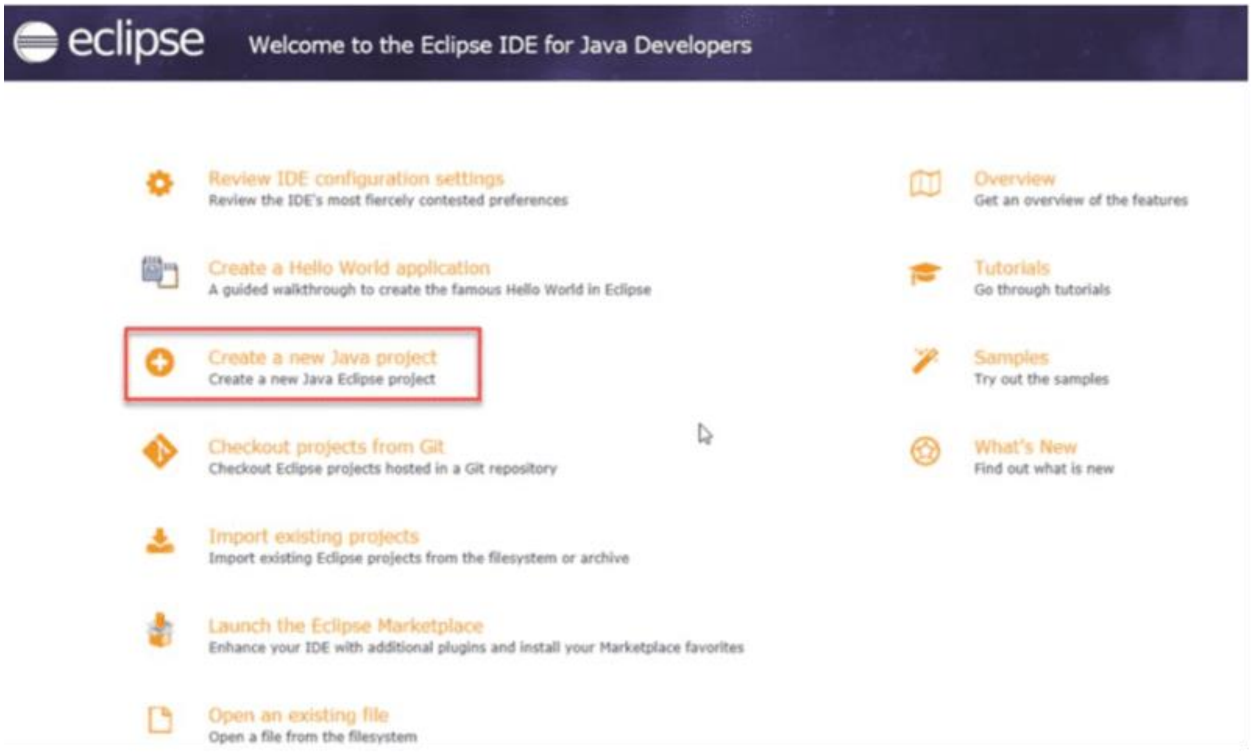
Installation Folder: C:\Users\Neha Shri\eclipse\java-2021-06

✓ create start menu entry

✓ create desktop shortcut

Step 8: Create New Project

Now click on "Create a new Java project".


eclipse Welcome to the Eclipse IDE for Java Developers

Review IDE configuration settings
Review the IDE's most fiercely contested preferences

Create a Hello World application
A guided walkthrough to create the famous Hello World in Eclipse

Create a new Java project
Create a new Java Eclipse project

Checkout projects from Git
Checkout Eclipse projects hosted in a Git repository

Import existing projects
Import existing Eclipse projects from the filesystem or archive

Launch the Eclipse Marketplace
Enhance your IDE with additional plugins and install your Marketplace favorites

Open an existing file
Open a file from the filesystem

Overview
Get an overview of the features

Tutorials
Go through tutorials

Samples
Try out the samples

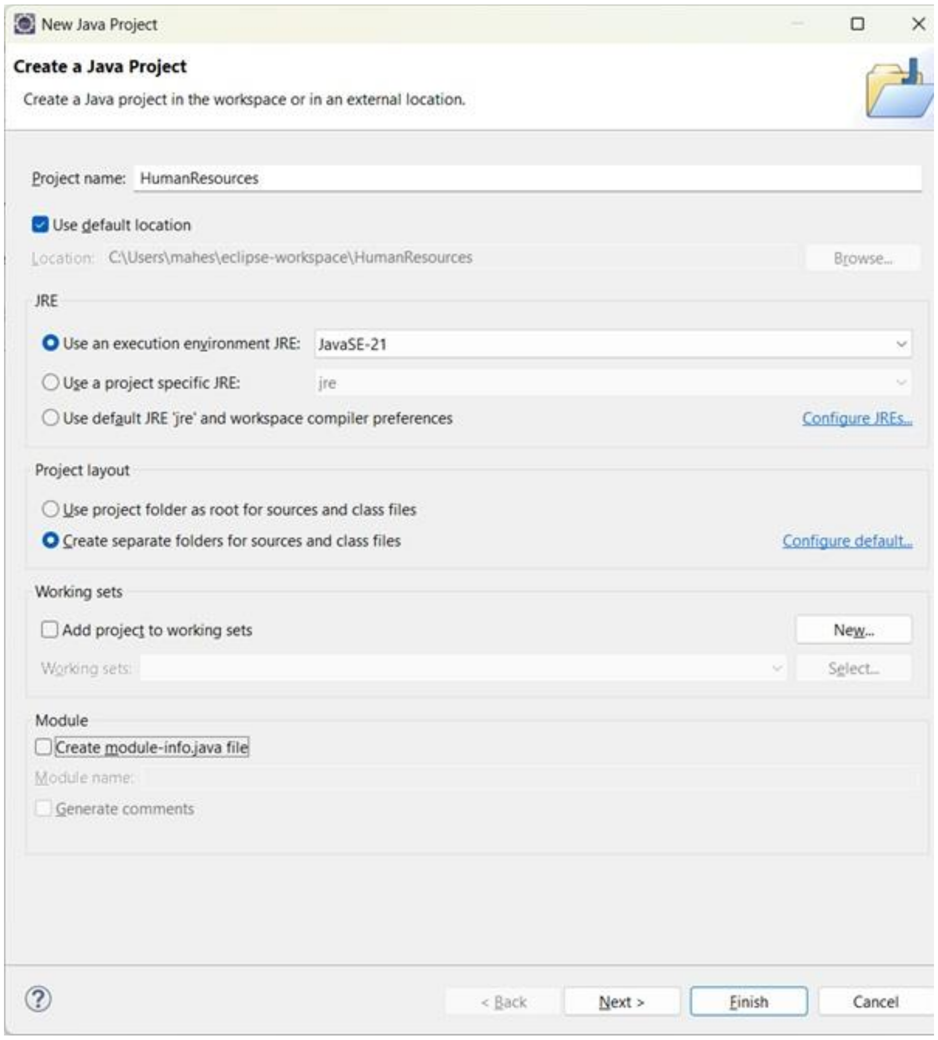
What's New
Find out what is new

STEP 9:Create a new java project

- By clicking on the File menu and choosing New → Java Project.

- By right clicking anywhere in the Project Explorer and selecting New → Java Project.
- By clicking on the New button ( ) in the Tool bar and selecting Java Project.

STEP 10: Enter the Project Name

- Select the Java Runtime Environment (JRE) or leave it at the default
- Select the Project Layout which determines whether there would be a separate folder for the source codes and class files. The recommended option is to create separate folders for sources and class files.
- 

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: HumanResources

[✓] Use default location

Location: C:\Users\mahes\eclipse-workspace\HumanResources

JRE

[✓ Use an execution environment JRE: JavaSE-21]

Use a project specific JRE: jre

Use default JRE 'jre' and workspace compiler preferences

Project layout

Use project folder as root for sources and class files

[✓ Create separate folders for sources and class files]

Working sets

Add project to working sets

Working sets:

Module

Create module-info.java file

Module name:

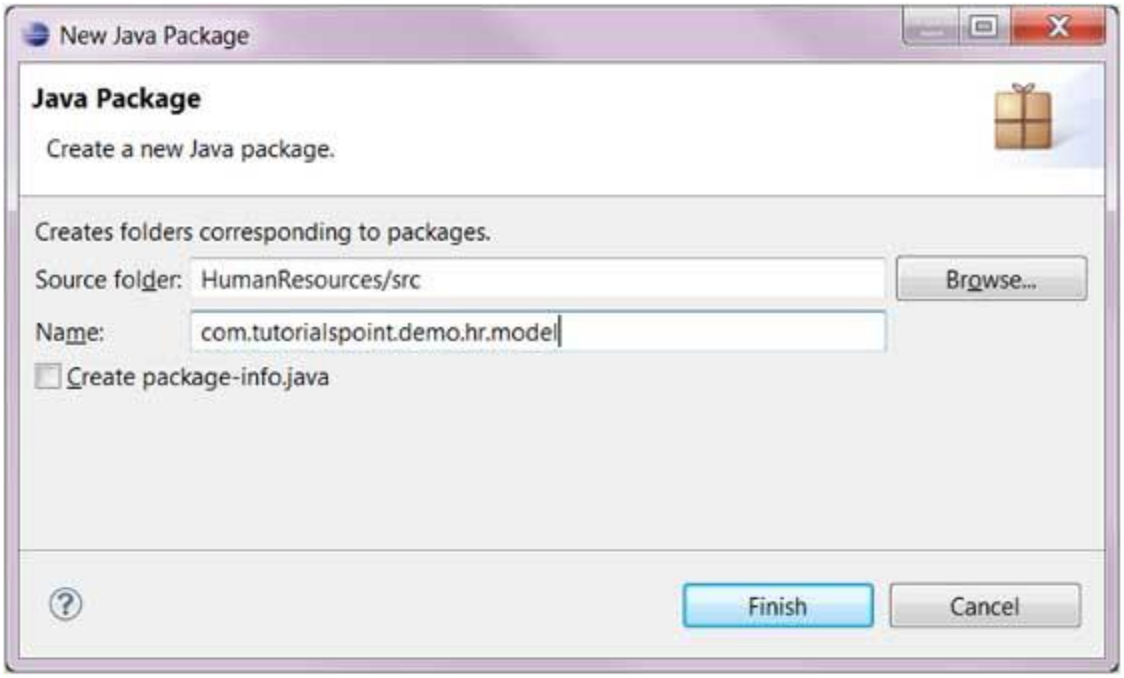
Generate comments

STEP 11: Create a new java package

- By clicking on the File menu and selecting New → Package.
- By right click in the package explorer and selecting New → Package.
- By clicking on the package icon which is in the tool bar( ).

STEP 11:

- Enter/confirm the source folder name.
- Enter the package name.
- Click on the Finish button.



Creates folders corresponding to packages.

Source folder: HumanResources/src

Name: com.tutorialspoint.demo.hr.model

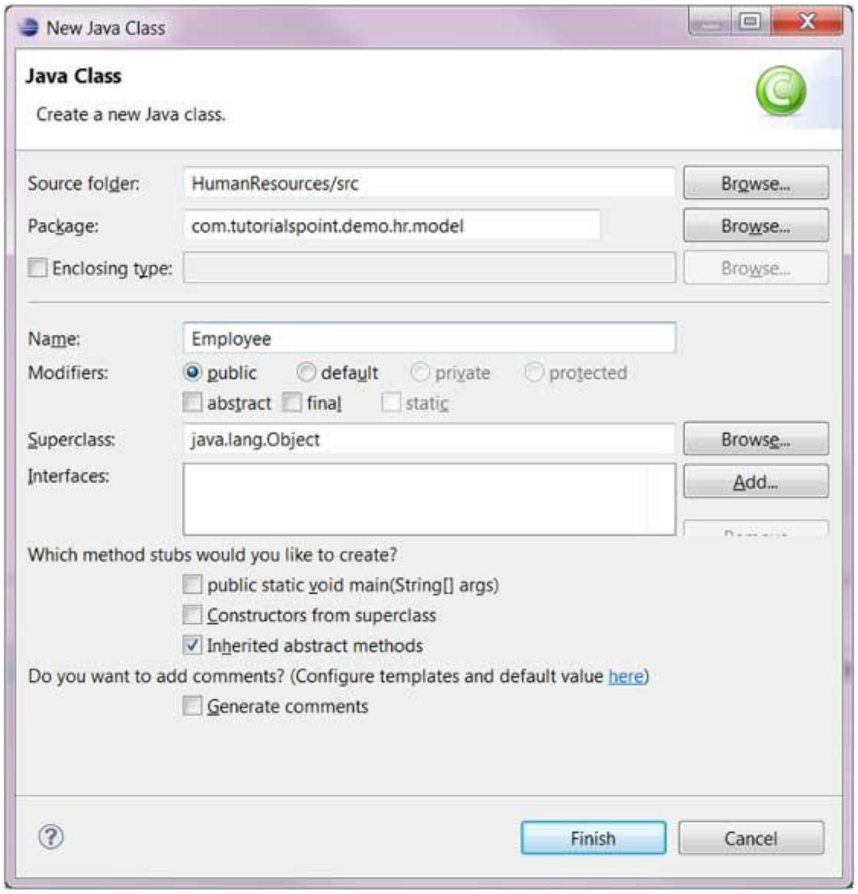
☐ Create package-info.java

STEP 12:Create a New Java class.

- By clicking on the File menu and selecting New → Class.
- By right clicking in the package explorer and selecting New → Class.
- By clicking on the class drop down button ( ) and selecting class ( ).

STEP 13:

- Ensure the source folder and package are correct.
- Enter the class name.
- Select the appropriate class modifier.
- Enter the super class name or click on the Browse button to search for an existing class.
- Click on the Add button to select the interfaces implemented by this class.
- Examine and modify the check boxes related to method stubs and comments.



New Java Class

Java Class

Create a new Java class.

Source folder: HumanResources/src

Package: com.tutorialspoint.demo.hr.model

☐ Enclosing type:

Name: Employee

Modifiers: [✓ public] default private protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

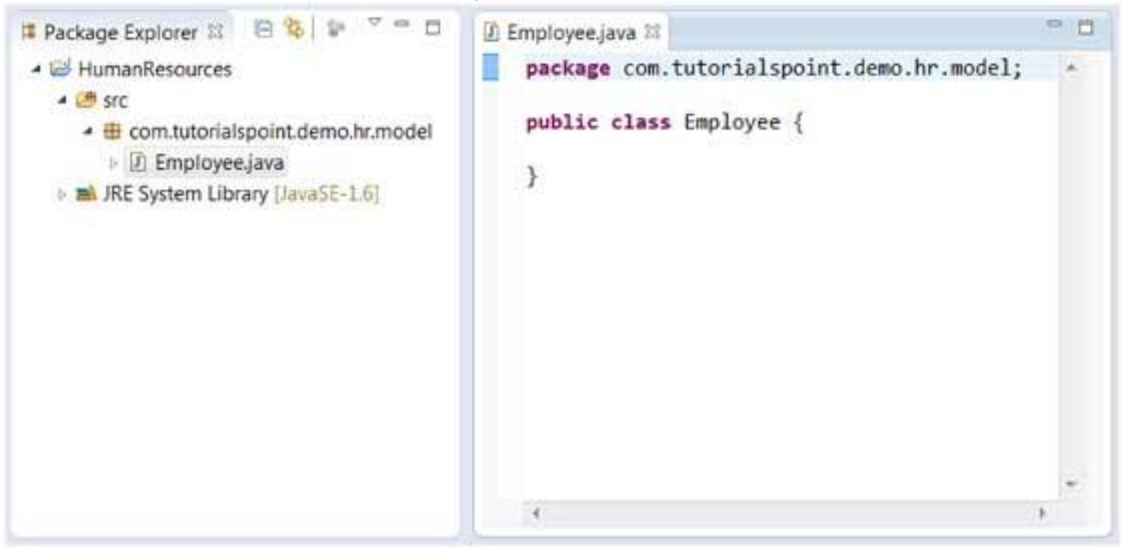
☐ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

STEP 14: Class created successfully.



Package Explorer
HumanResources
src
com.tutorialspoint.demo.hr.model
Employee.java
JRE System Library [JavaSE-1.6]
Employee.java

```
package com.tutorialspoint.demo.hr.model;

public class Employee {

}
```

BASIC PROGRAMS:

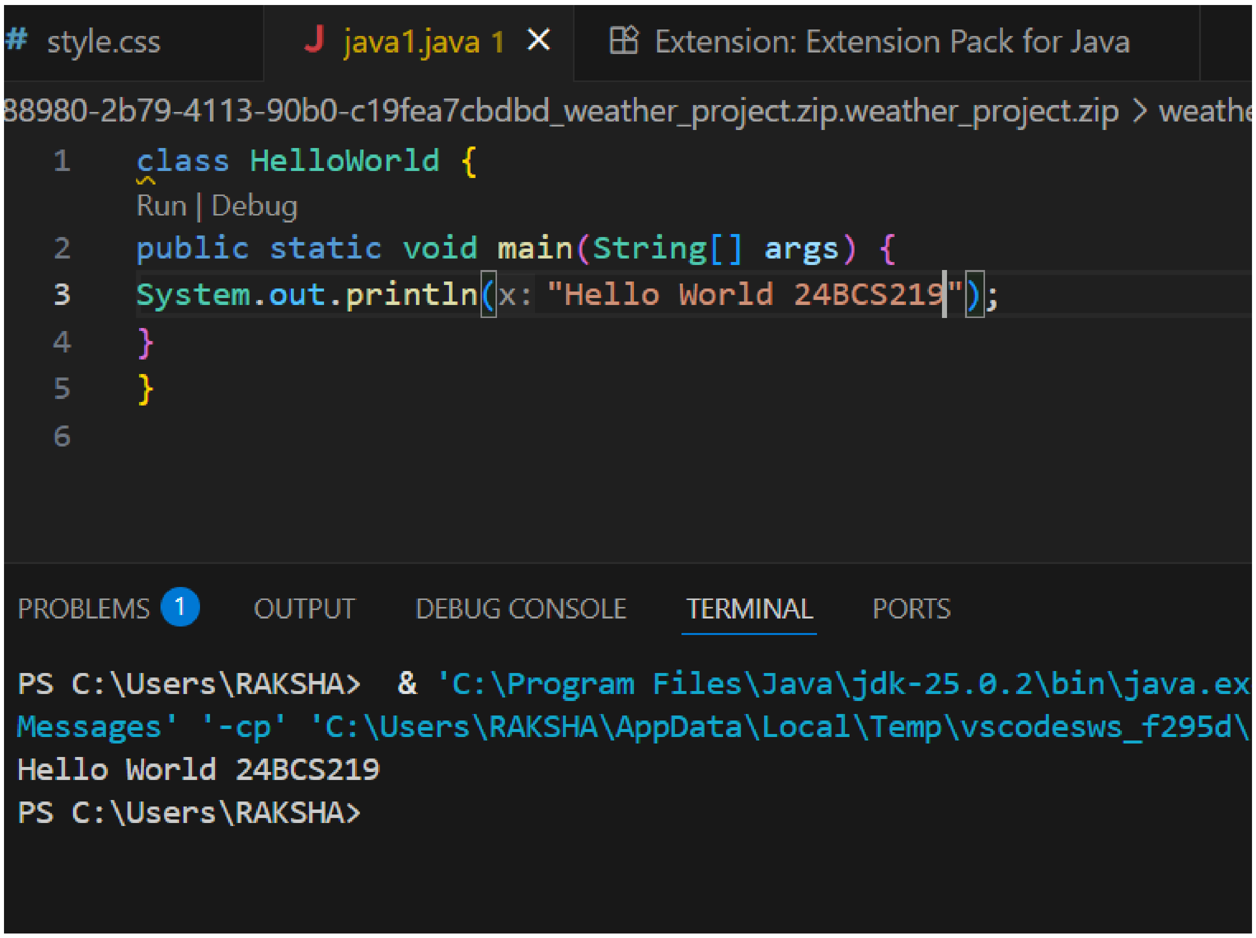
Program 1: Hello World Program

Source Code:

```
class HelloWorld {
public static void main(String[] args) {
System.out.println("Hello World");
}
}
```

Output:

```
Hello World
```



# style.css
J java1.java 1 ×
Extension: Extension Pack for Java
88980-2b79-4113-90b0-c19fea7cbdbd_weather_project.zip.weather_project.zip > weathe
Run | Debug
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```
class HelloWorld {
public static void main(String[] args) {
System.out.println(x: "Hello World 24BCS219");
}
}
```

```
PS C:\Users\RAKSHA>  & 'C:\Program Files\Java\jdk-25.0.2\bin\java.ex
Messages' '-cp' 'C:\Users\RAKSHA\AppData\Local\Temp\vscodesws_f295d\
Hello World 24BCS219
PS C:\Users\RAKSHA>
```

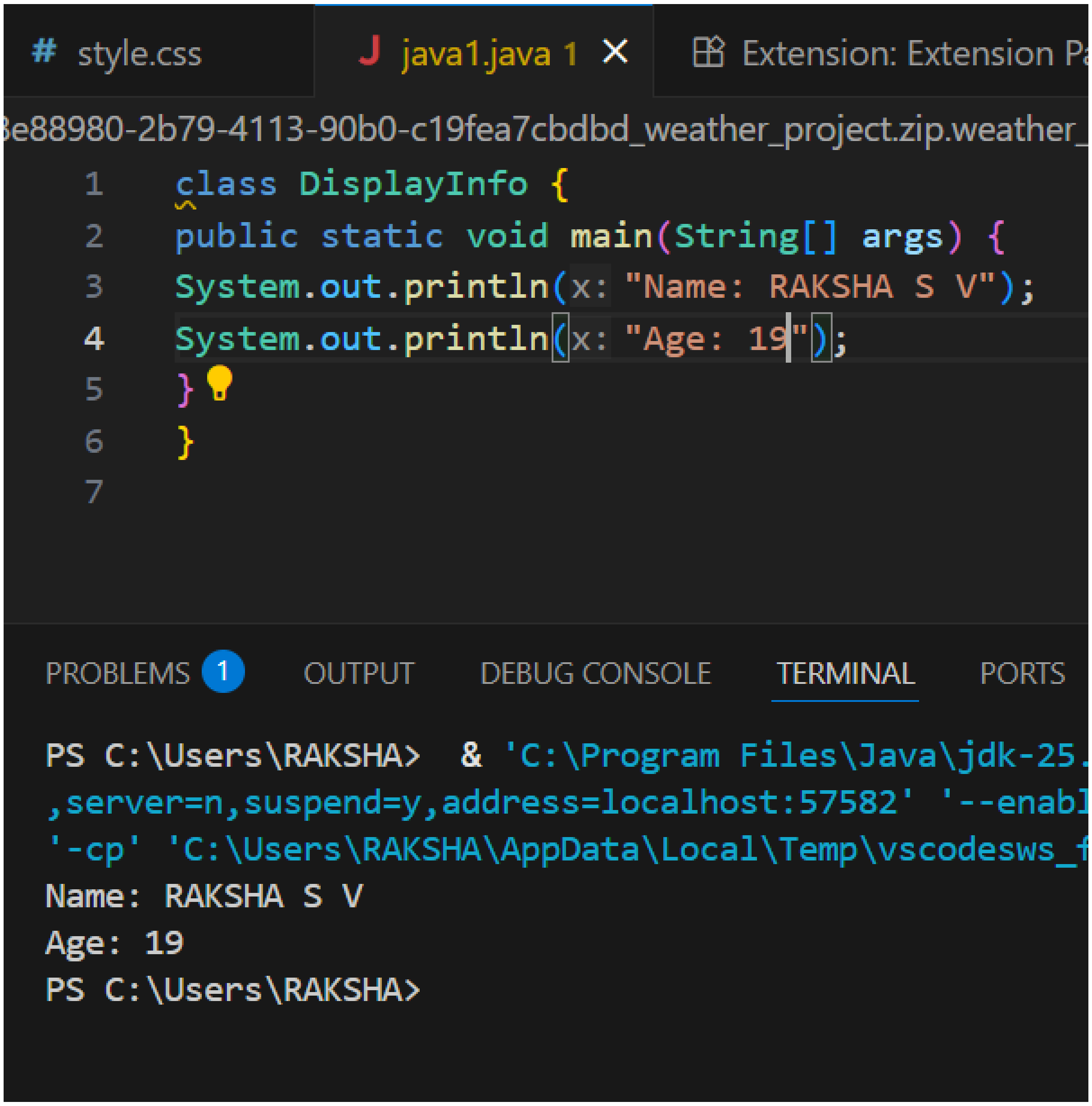
Program 2: Display Personal Details

Source Code:

```
class DisplayInfo {
```

```
public static void main(String[] args) {
System.out.println("Name: Anitha");
System.out.println("Age: 20");
}
}
```

Output:



```java
class DisplayInfo {
public static void main(String[] args) {
System.out.println(x: "Name: RAKSHA S V");
System.out.println(x: "Age: 19");
}
}
```

```
PS C:\Users\RAKSHA>  & 'C:\Program Files\Java\jdk-25.
,server=n,suspend=y,address=localhost:57582' '--enabl
'-cp' 'C:\Users\RAKSHA\AppData\Local\Temp\vscodesws_f
Name: RAKSHA S V
Age: 19
PS C:\Users\RAKSHA>
```

Name: RAKSHA S V

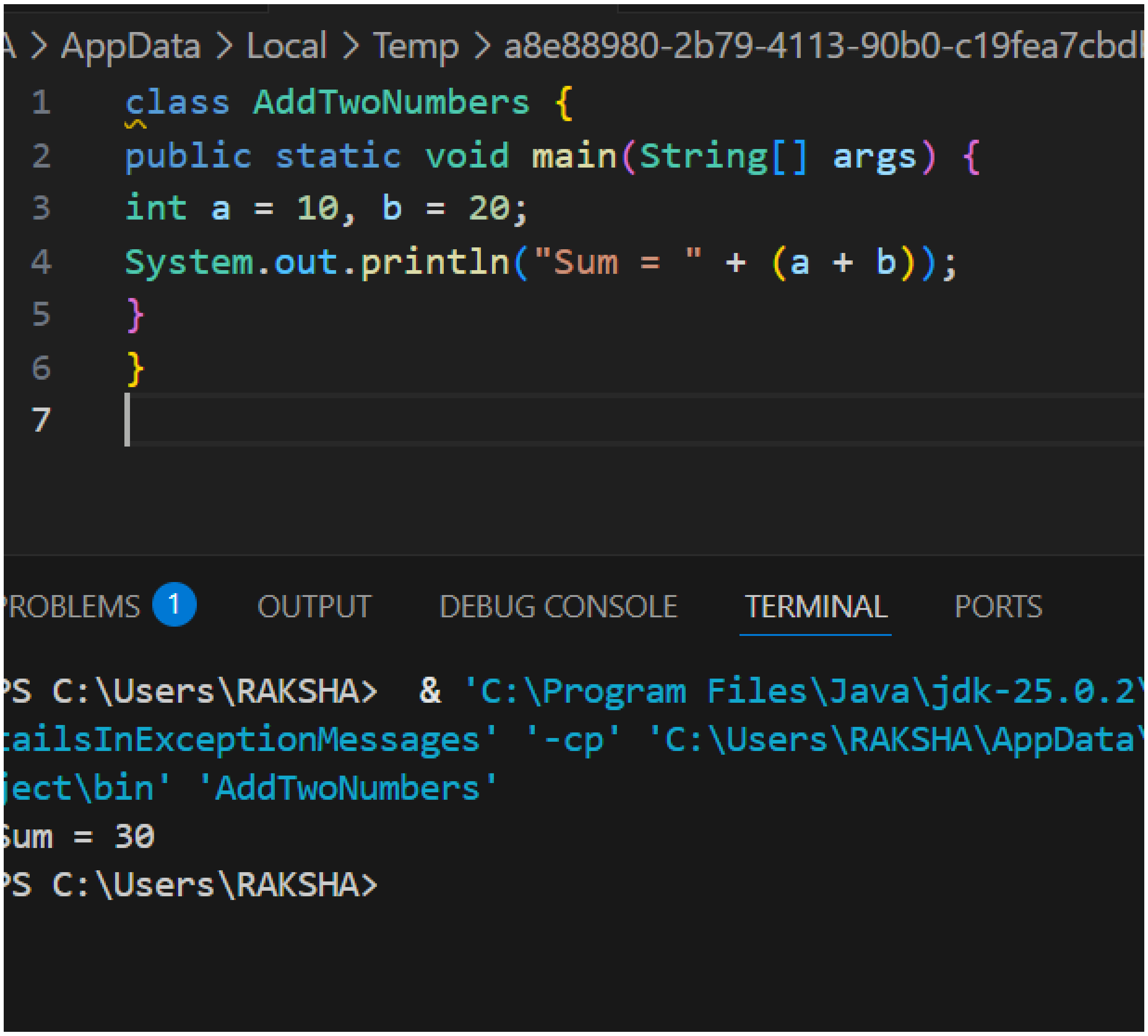
Age: 19

Program 3: Addition of Two Numbers

Source Code:

```
class AddTwoNumbers {
public static void main(String[] args) {
int a = 10, b = 20;
System.out.println("Sum = " + (a + b));
}
}
```

Output:



```
A > AppData > Local > Temp > a8e88980-2b79-4113-90b0-c19fea7cbdl
class AddTwoNumbers {
public static void main(String[] args) {
int a = 10, b = 20;
System.out.println("Sum = " + (a + b));
}
}
```

```
PS C:\Users\RAKSHA>  & 'C:\Program Files\Java\jdk-25.0.2\
tailsInExceptionMessages' '-cp' 'C:\Users\RAKSHA\AppData\
ject\bin' 'AddTwoNumbers'
Sum = 30
PS C:\Users\RAKSHA>
```

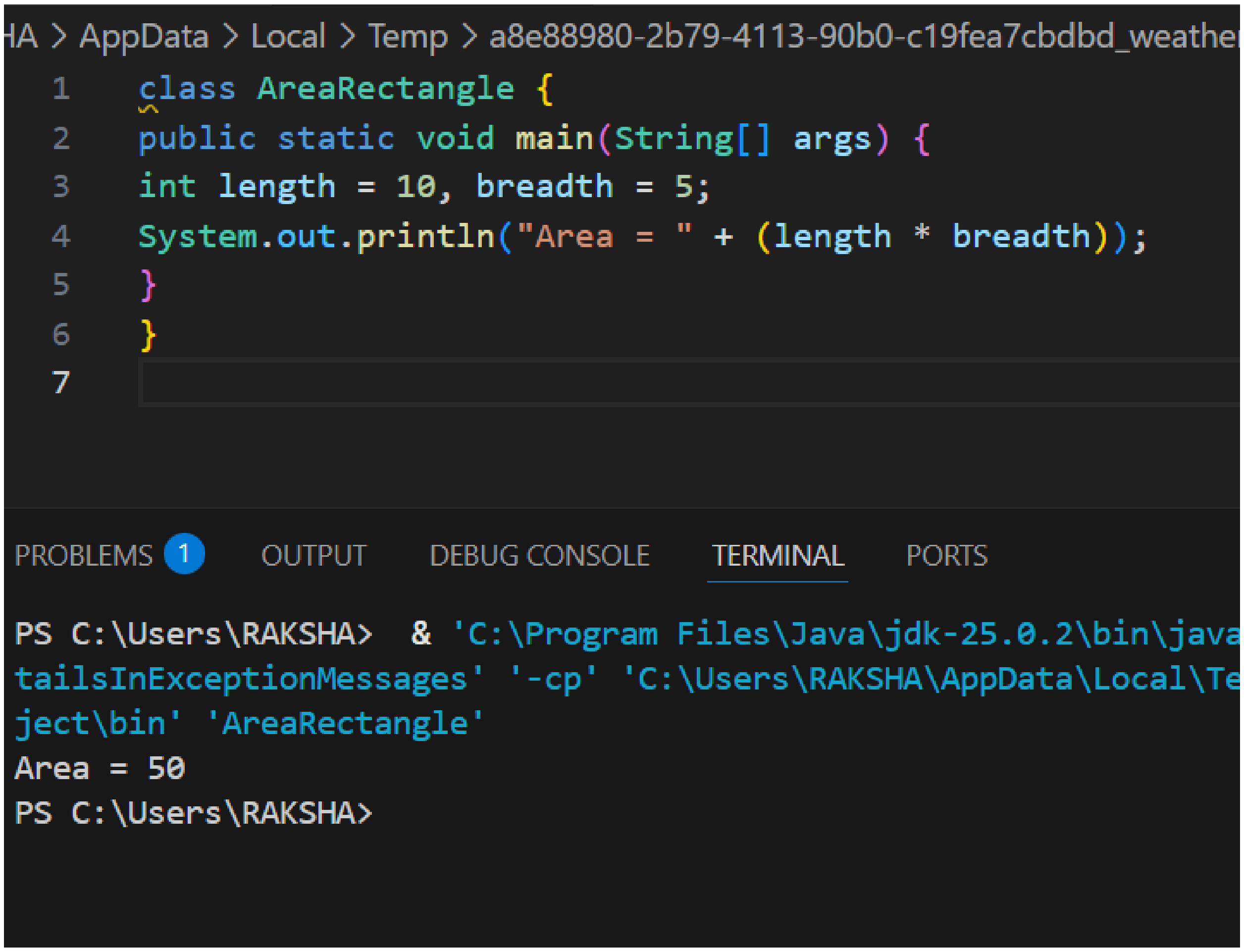
Sum = 30

Program 4: Area of a Rectangle

Source Code:

```
class AreaRectangle {
public static void main(String[] args) {
int length = 10, breadth = 5;
System.out.println("Area = " + (length * breadth));
}
}
```

Output:



```
HA > AppData > Local > Temp > a8e88980-2b79-4113-90b0-c19fea7cbdbd_weathe
class AreaRectangle {
public static void main(String[] args) {
int length = 10, breadth = 5;
System.out.println("Area = " + (length * breadth));
}
}
```

```
PS C:\Users\RAKSHA>  & 'C:\Program Files\Java\jdk-25.0.2\bin\java
tailsInExceptionMessages' '-cp' 'C:\Users\RAKSHA\AppData\Local\Te
ject\bin' 'AreaRectangle'
Area = 50
PS C:\Users\RAKSHA>
```

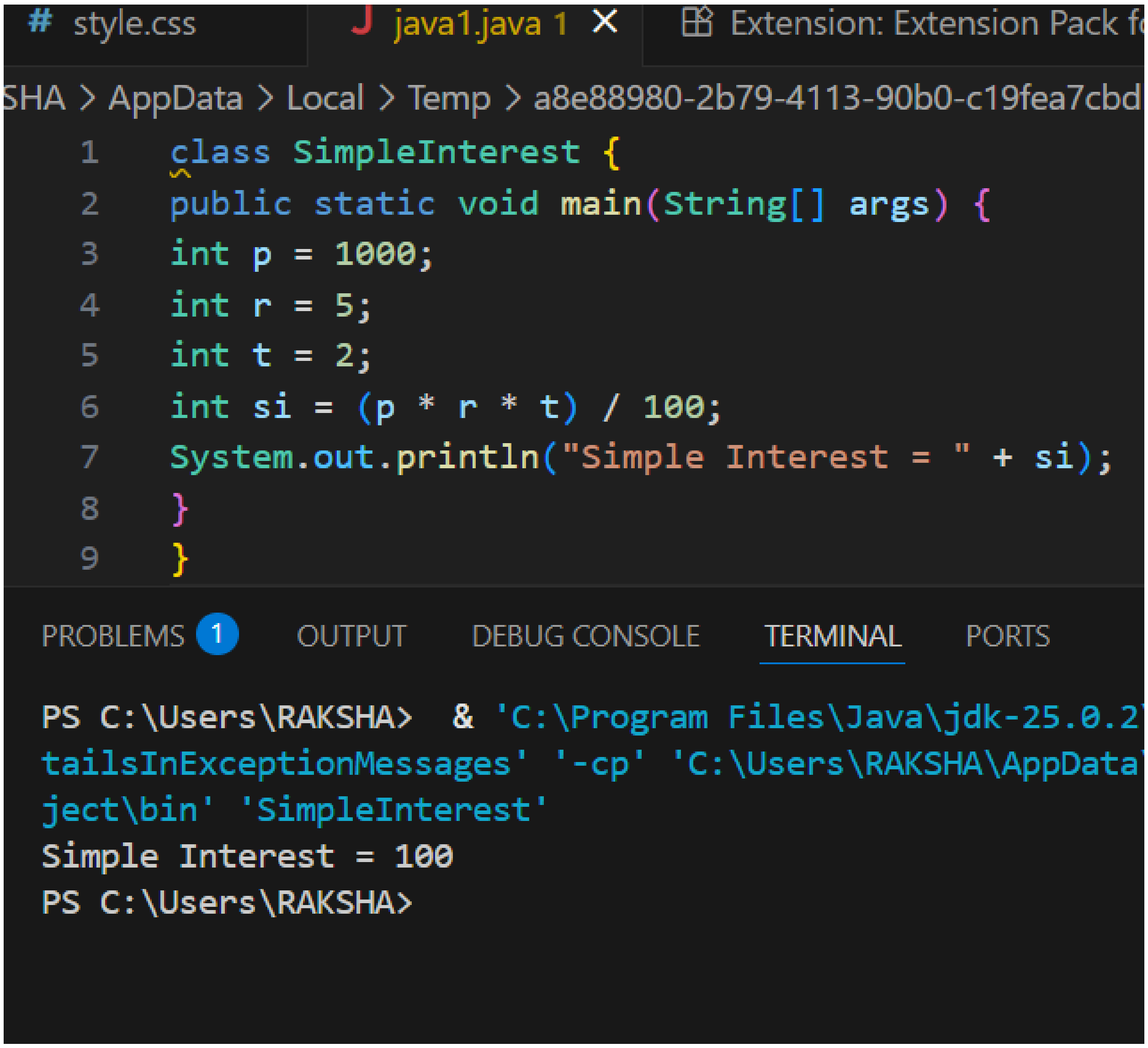
Area = 50

Program 5: Simple Interest Calculation

Source Code:

```
class SimpleInterest {
public static void main(String[] args) {
int p = 1000;
int r = 5;
int t = 2;
int si = (p * r * t) / 100;
System.out.println("Simple Interest = " + si);
}
}
```

Output:


style.css   java1.java 1 ×   Extension: Extension Pack f

SHA > AppData > Local > Temp > a8e88980-2b79-4113-90b0-c19fea7cbd

```java
class SimpleInterest {
public static void main(String[] args) {
int p = 1000;
int r = 5;
int t = 2;
int si = (p * r * t) / 100;
System.out.println("Simple Interest = " + si);
}
}
```

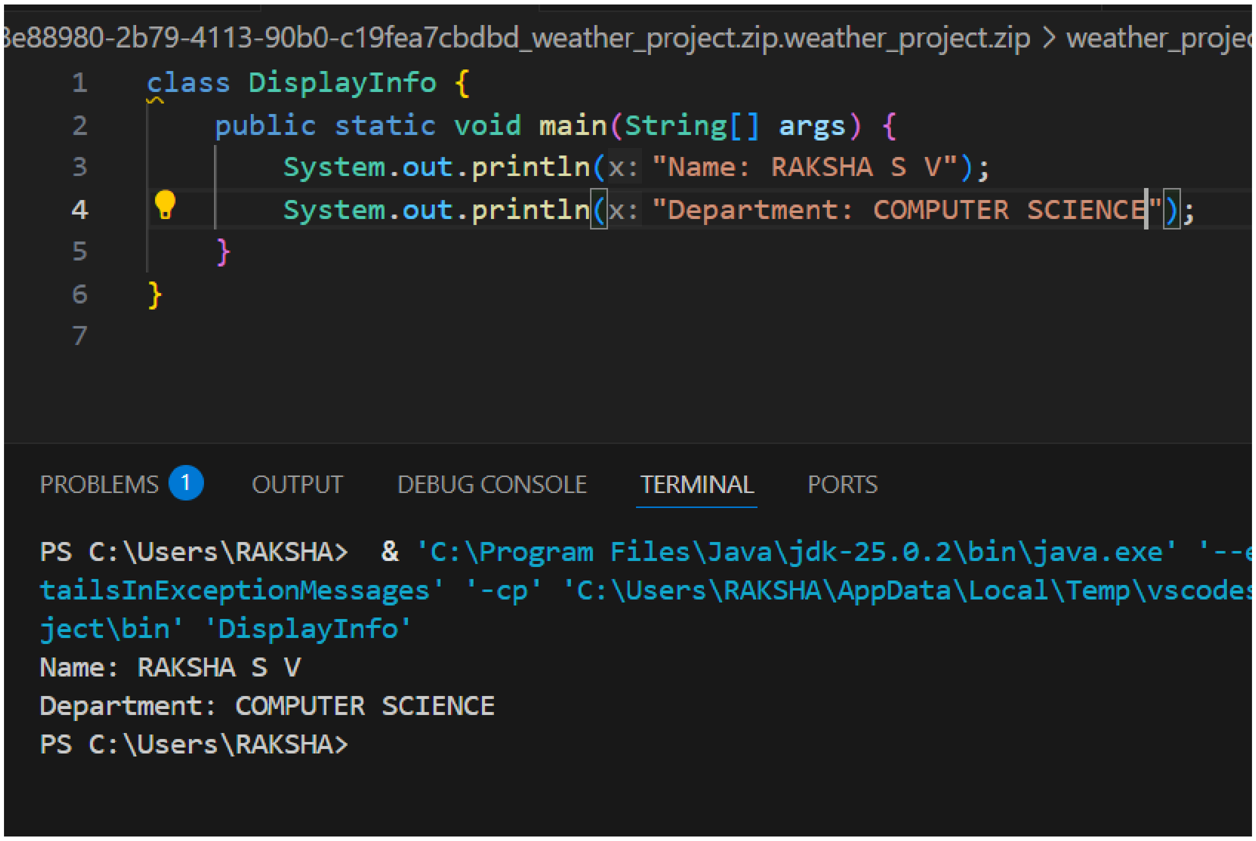
PROBLEMS 1   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\RAKSHA>  & 'C:\Program Files\Java\jdk-25.0.2
tailsInExceptionMessages' '-cp' 'C:\Users\RAKSHA\AppData
ject\bin' 'SimpleInterest'
Simple Interest = 100
PS C:\Users\RAKSHA>
```

Simple Interest = 100

POST LAB EXERCISE

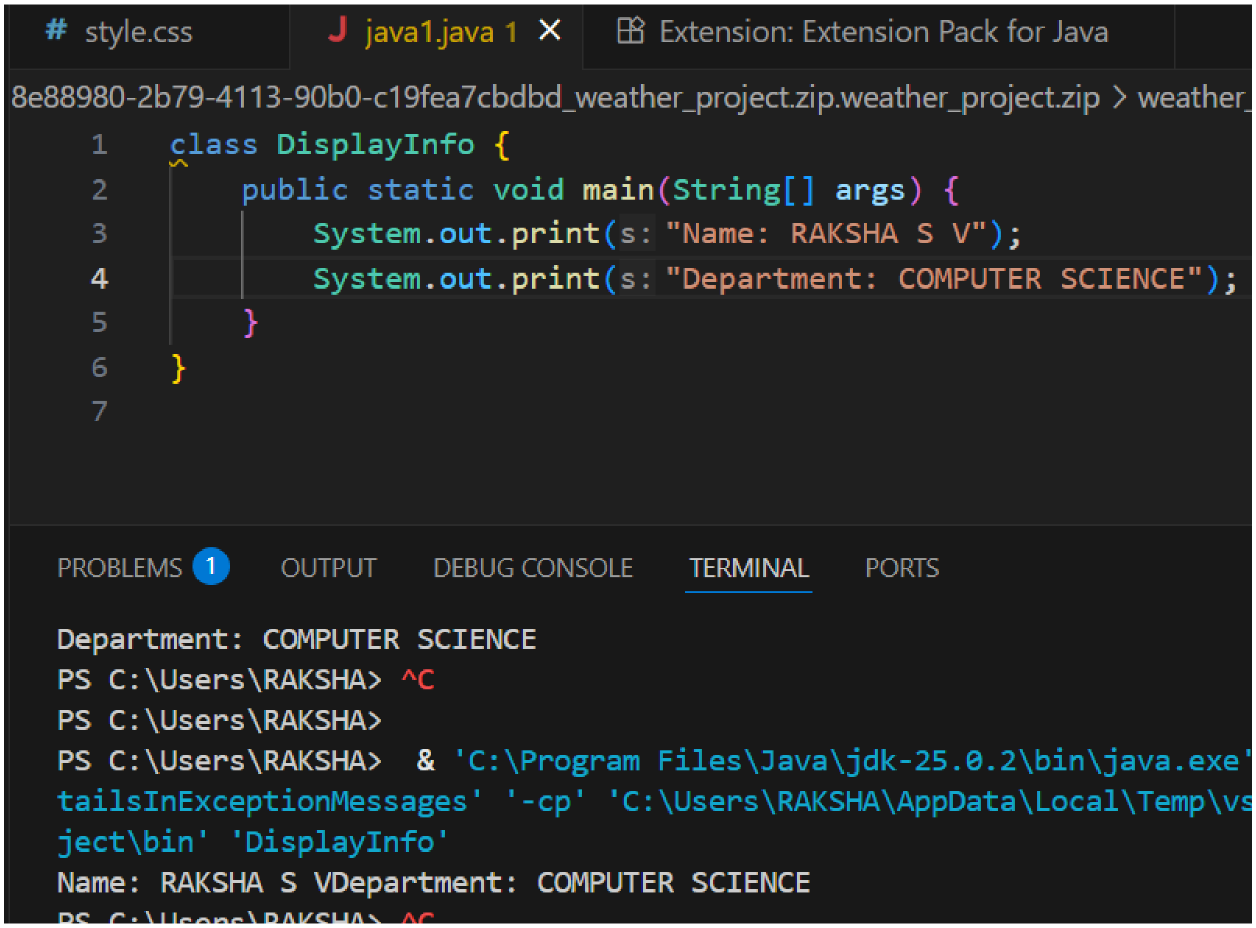
1. **Write a Java program to display your name and department.**



```
3e88980-2b79-4113-90b0-c19fea7cbdbd_weather_project.zip.weather_project.zip > weather_projec
class DisplayInfo {
    public static void main(String[] args) {
        System.out.println(x: "Name: RAKSHA S V");
        System.out.println(x: "Department: COMPUTER SCIENCE");
    }
}
```

```
PS C:\Users\RAKSHA>  & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '--e
tailsInExceptionMessages' '-cp' 'C:\Users\RAKSHA\AppData\Local\Temp\vscodes
ject\bin' 'DisplayInfo'
Name: RAKSHA S V
Department: COMPUTER SCIENCE
PS C:\Users\RAKSHA>
```

2. **Modify the program to print the output in same line.**

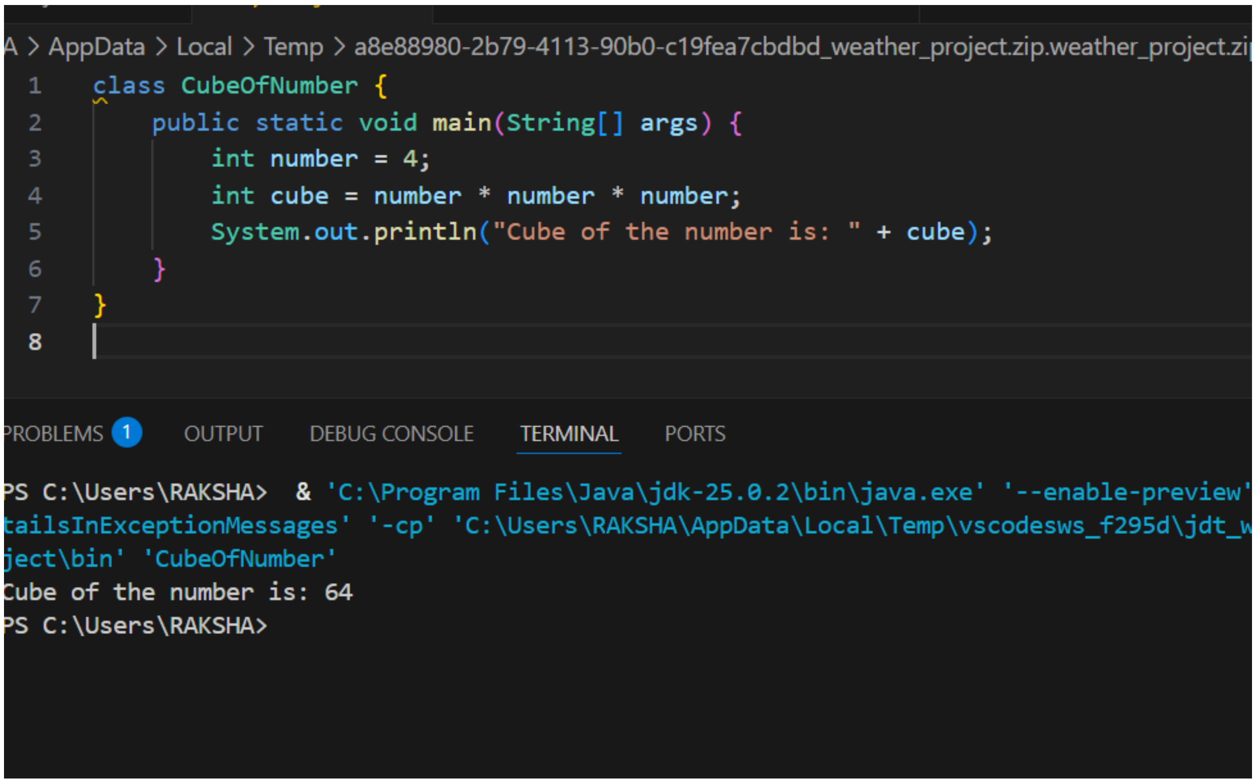


# style.css
java1.java 1
Extension: Extension Pack for Java
8e88980-2b79-4113-90b0-c19fea7cbdbd_weather_project.zip.weather_project.zip > weather_
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```java
class DisplayInfo {
    public static void main(String[] args) {
        System.out.print(s: "Name: RAKSHA S V");
        System.out.print(s: "Department: COMPUTER SCIENCE");
    }
}
```

```
Department: COMPUTER SCIENCE
PS C:\Users\RAKSHA> ^C
PS C:\Users\RAKSHA>
PS C:\Users\RAKSHA>  & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe'
tailsInExceptionMessages' '-cp' 'C:\Users\RAKSHA\AppData\Local\Temp\vs
ject\bin' 'DisplayInfo'
Name: RAKSHA S VDepartment: COMPUTER SCIENCE
```

3. **What happens if `main()` is written without `static`?**
   If the main() method is written without the static keyword, the program will compile successfully, but it will not execute. At runtime, the Java Virtual Machine will generate an error stating that the main method is not static. This is because the JVM invokes the main() method without creating an object of the class, and only static methods can be called directly using the class name. Therefore, the static keyword is mandatory in the main() method signature.
4. **Why is Java called platform independent?**
   Java is called platform independent because Java source code is compiled into an intermediate form known as bytecode. This bytecode is not specific to any operating system or hardware architecture. It is executed by the Java Virtual Machine (JVM), which is available for different platforms. As a result, the same Java program can run on any system that has a compatible JVM, supporting the principle "Write Once, Run Anywhere."
5. **Write a program to find the cube of a number.**



A > AppData > Local > Temp > a8e88980-2b79-4113-90b0-c19fea7cbdbd_weather_project.zip.weather_project.zi

```java
class CubeOfNumber {
    public static void main(String[] args) {
        int number = 4;
        int cube = number * number * number;
        System.out.println("Cube of the number is: " + cube);
    }
}

```

```
PS C:\Users\RAKSHA>  & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '--enable-preview'
tailsInExceptionMessages' '-cp' 'C:\Users\RAKSHA\AppData\Local\Temp\vscodesws_f295d\jdt_w
ject\bin' 'CubeOfNumber'
Cube of the number is: 64
PS C:\Users\RAKSHA>
```

Result:

Thus the Java IDE was successfully installed and a simple Java program was executed.

ASSESSMENT

Description	**Max Marks**	**Marks Awarded**
Pre Lab Exercise	**5**	
In Lab Exercise	**10**	
Post Lab Exercise	**5**	
Viva	**10**	
Total	**30**	
Faculty Signature		