Experiment Number : 01 **Date: 27.01.2026**

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?**

   JDK (Java Development Kit) is a software development kit used to develop, compile, and run Java programs.
   Why it is required:

- To write Java programs
- To compile Java source code using the javac compiler
- To run Java applications
- It provides tools like compiler, debugger, and libraries

  Without JDK, we cannot develop Java programs.

2. **Difference between JDK, JRE, and JVM.**

Feature	JDK	JRE	JVM
Full form	Java Development Kit	Java Runtime Environment	Java Virtual Machine
Purpose	Used to **develop** Java programs	Used to **run** Java programs	Executes Java bytecode
Contains	JRE + development tools	JVM + libraries	Only execution engine
Includes compiler	Yes (javac)	No	No
Used by	Developers	End users	Internally by JRE

3. **What is the purpose of the main() method in Java?**

The main() method is the entry point of a Java program.

Purpose:

- It is the method from where program execution starts
- JVM looks for main() method to start execution
- Without main() method, the program will not run

IN LAB EXERCISE

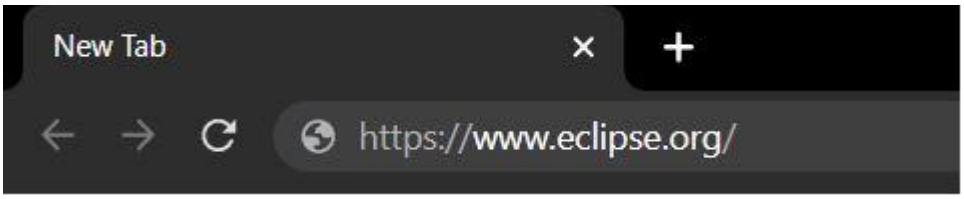
Objective:

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.

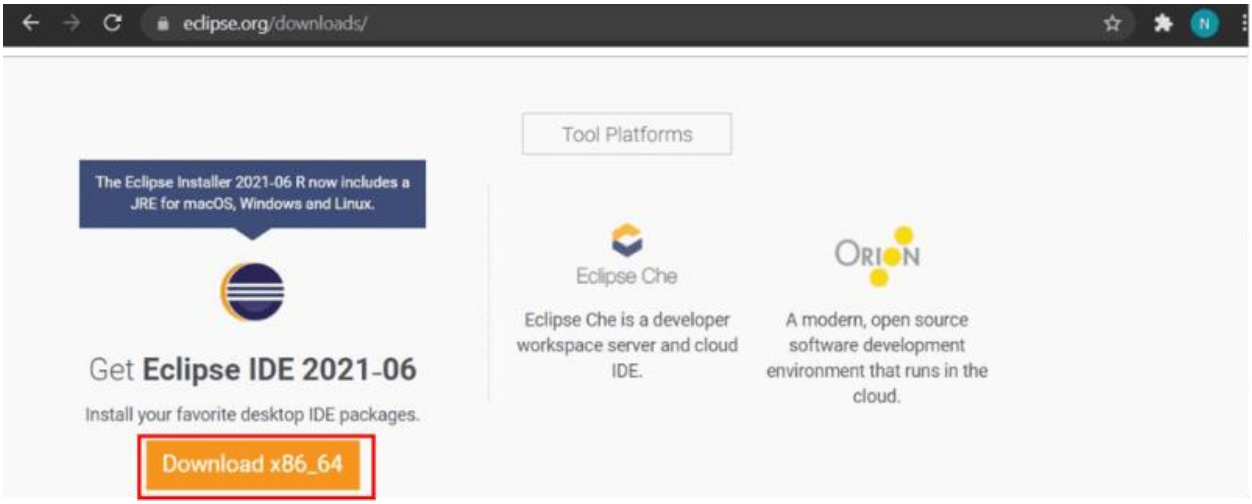


STEP 2: Download Eclipse Installer

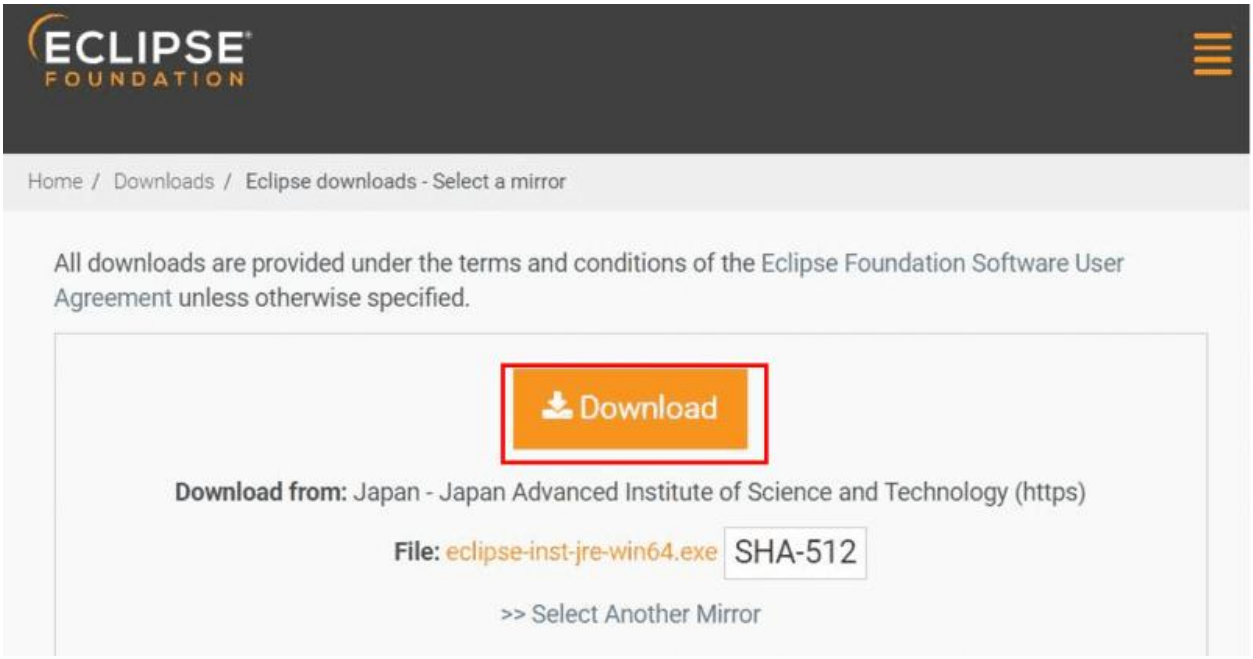
- Then, click on the "Download" button to download Eclipse IDE.

STEP 3: Download EXE

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

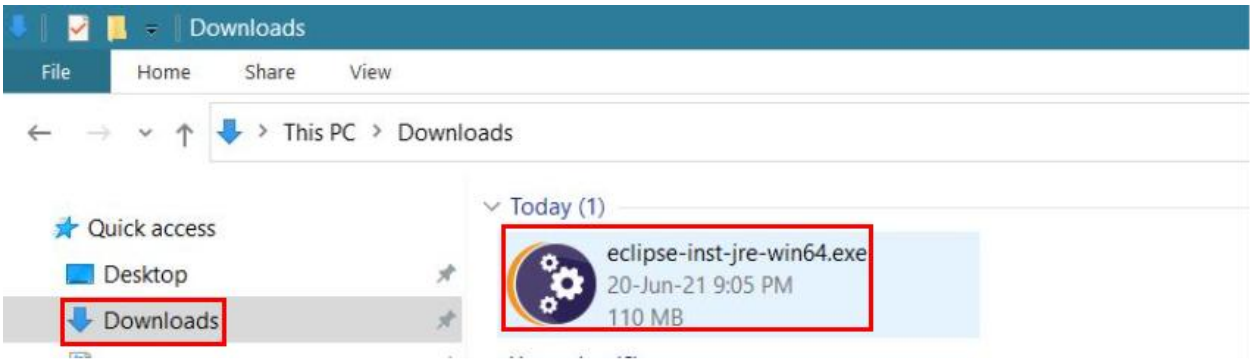


STEP 4: Then click on the "Download" button. After clicking on the download button the .exe file for the eclipse will be downloaded.



STEP 5: Open Download EXE

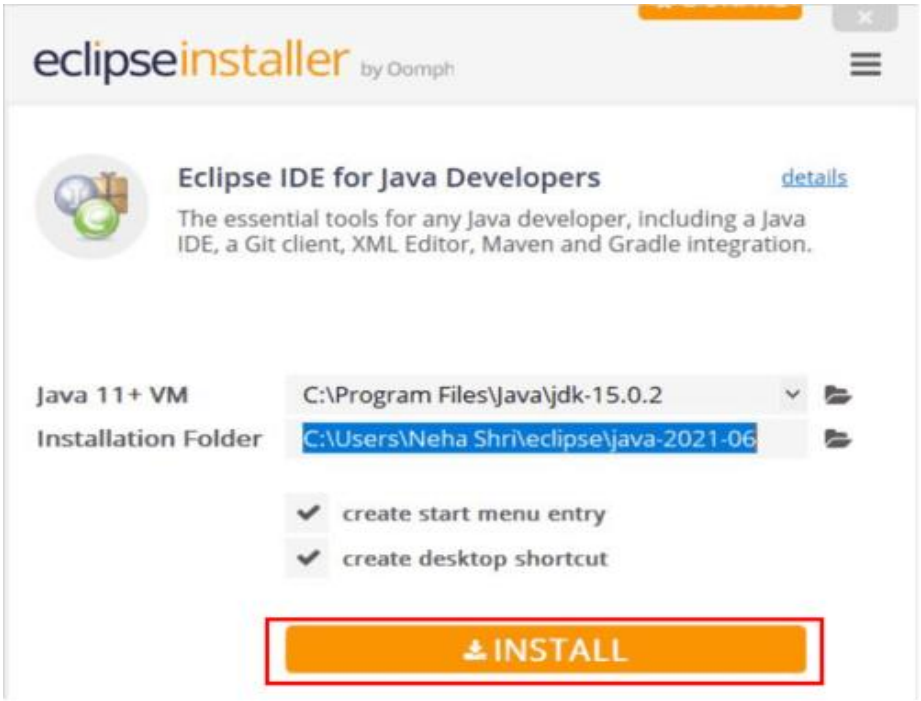
- Now go to File Explorer and click on "Downloads" after that click on the "*eclipse-inst-jre-win64.exe*" file for installing Eclipse IDE**.**



STEP 6: Install Eclipse

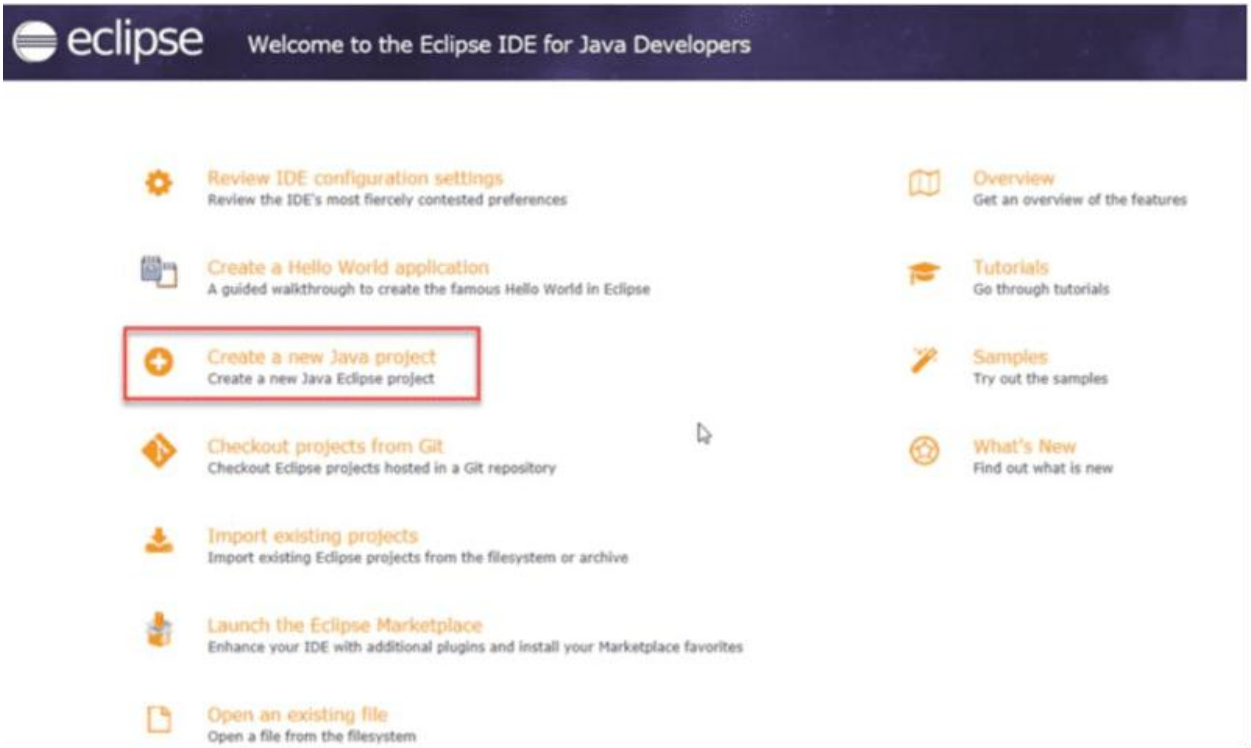
- Then, click on "Eclipse IDE for Java Developers".

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



Step 8: Create New Project

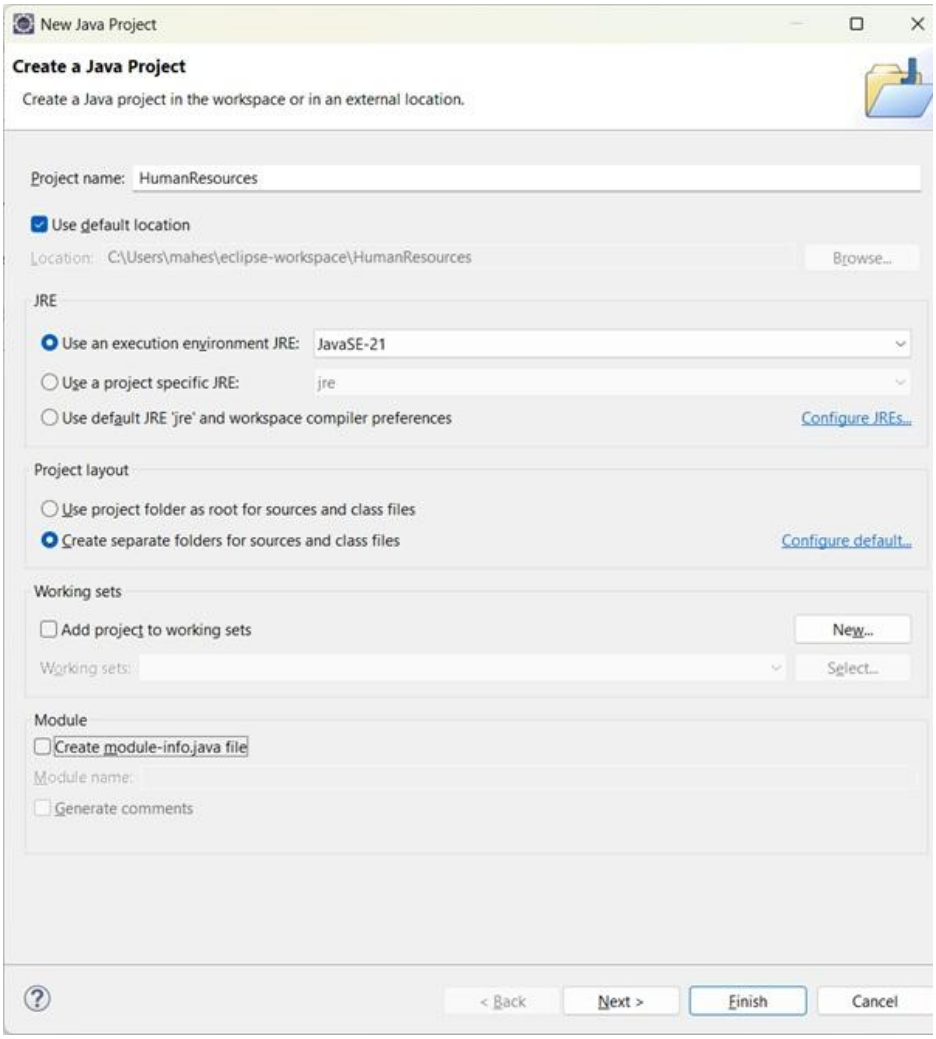
Now click on "Create a new Java project".



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.
- 

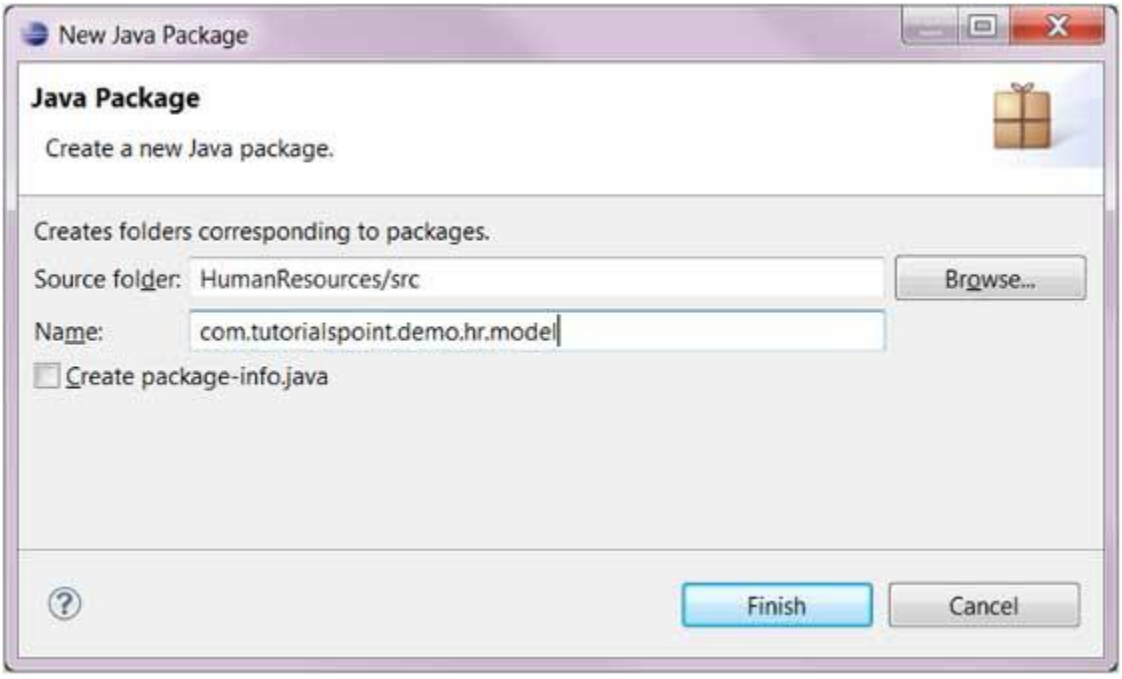
STEP 11: Create a new java package

- 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.

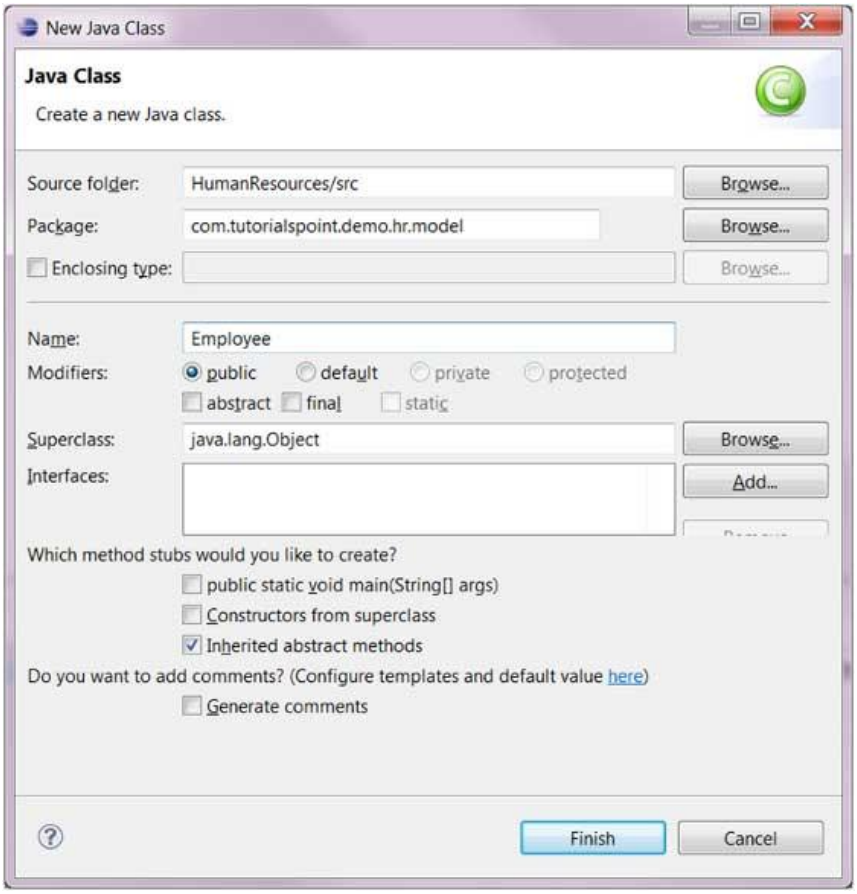


STEP 12:Create a New Java class.

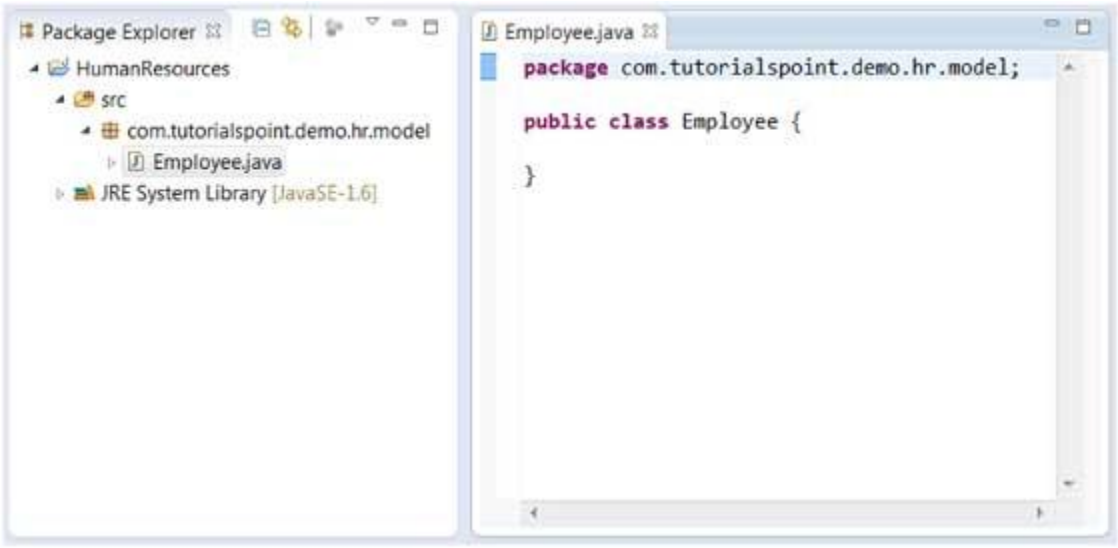
- By clicking on the File menu and selecting New → Class.
- By right clicking in the package explorer and selecting New → Class.
- By clicking on the class drop down button ( ) and selecting class ( ).

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.



STEP 14: Class created successfully.



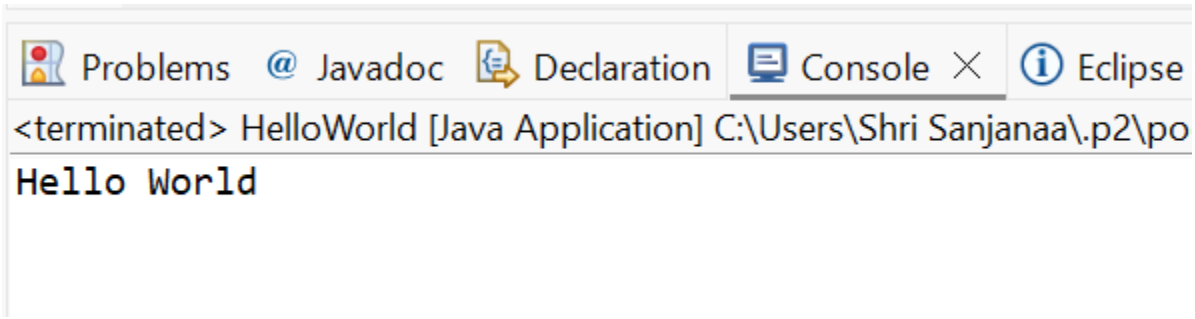
BASIC PROGRAMS:

Program 1: Hello World Program

Source Code:

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

Output:

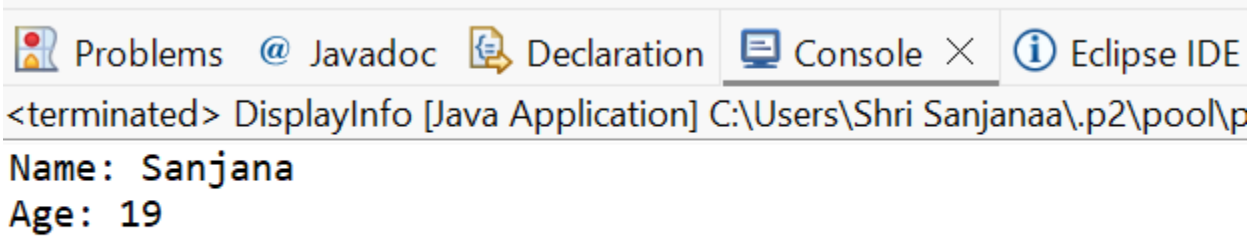


Program 2: Display Personal Details

Source Code:

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

Output:

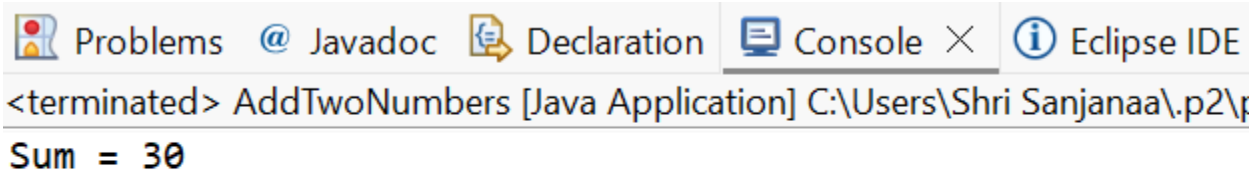


Program 3: Addition of Two Numbers

Source Code:

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

Output:



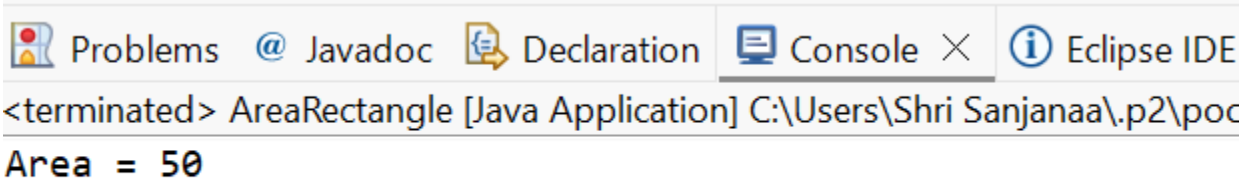
Program 4: Area of a Rectangle

Source Code:

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

}

Output:

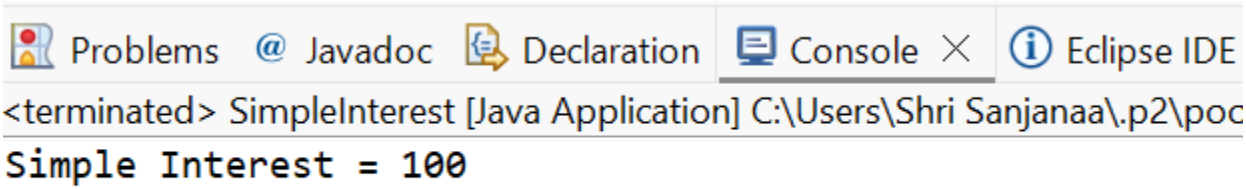


Program 5: Simple Interest Calculation

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:



POST LAB EXERCISE

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

```
class DisplayInfo {
public static void main(String[] args) {
System.out.println("Name: Sanjanaa");
```

```
System.out.println("Department: Computer Science Engineering");
}
}
```

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

```
class DisplayInfo {
public static void main(String[] args) {
System.out.print("Name: Sanjanaa ");
System.out.print("Department: Computer Science Engineering");
}
}
```

3. **What happens if `main()` is written without `static`?**

If main() is written without static, the program will not run.
Reason:

- JVM starts execution without creating an object
- Non-static methods need an object to be called
- So JVM cannot call main() method

4. **Why is Java called platform independent?**

Java is called platform independent because:

- Java source code is compiled into bytecode
- Bytecode is same for all operating systems
- JVM converts bytecode into machine code
- Different OS → Different JVM → Same program runs everywhere

This concept is called:
Write Once, Run Anywhere (WORA)

5. **Write a program to find the cube of a number.**

```
import java.util.Scanner;
class Cube {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter a number: ");
int num = sc.nextInt();
int cube = num * num * num;
System.out.println("Cube of the number is: " + cube);
}
}
```

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	