

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 is a package of tools and libraries provided by Oracle (and other vendors) that allows developers to write, compile, debug, and run Java programs.

2. Difference between JDK, JRE, and JVM.

JVM (Java Virtual Machine)

JVM is the runtime engine that executes Java bytecode. It converts bytecode into machine-specific instructions and enables platform independence.

JRE (Java Runtime Environment)

JRE provides the environment required to run Java programs. It contains the JVM and standard Java libraries but does not include development tools.

JDK (Java Development Kit)

JDK is used to develop Java applications. It includes JRE along with development tools like the compiler, debugger, and other utilities.

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

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

The JVM starts program execution from the main() method and controls the flow of execution.

IN LAB EXERCISE

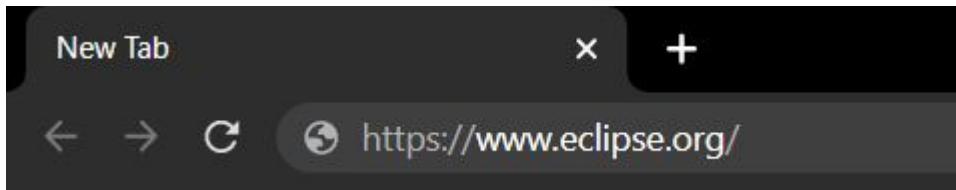
Objective:

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](https://www.eclipse.org/) Eclipse Downloads page.



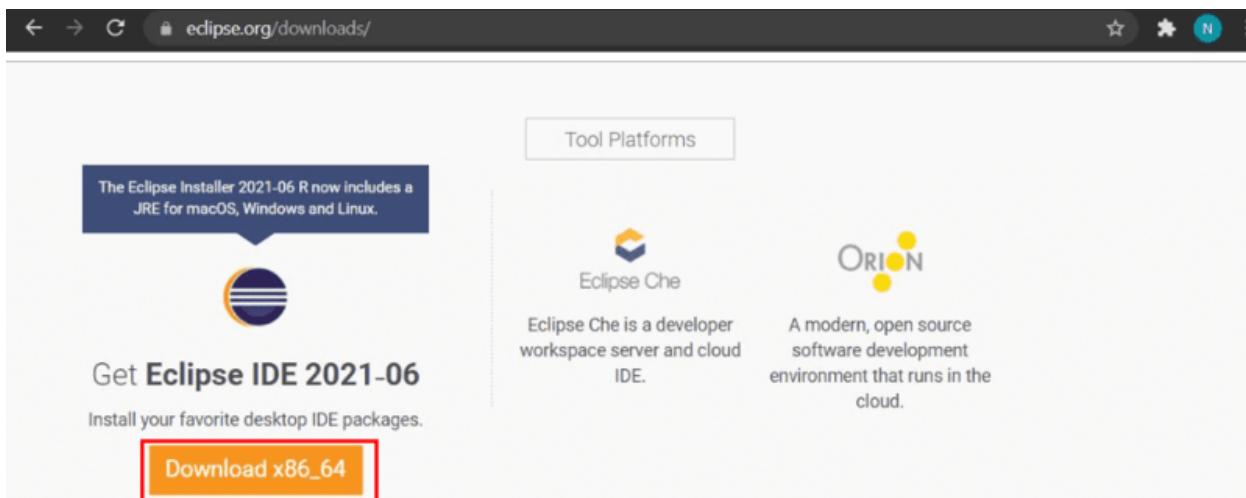
STEP 2: Download Eclipse Installer

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

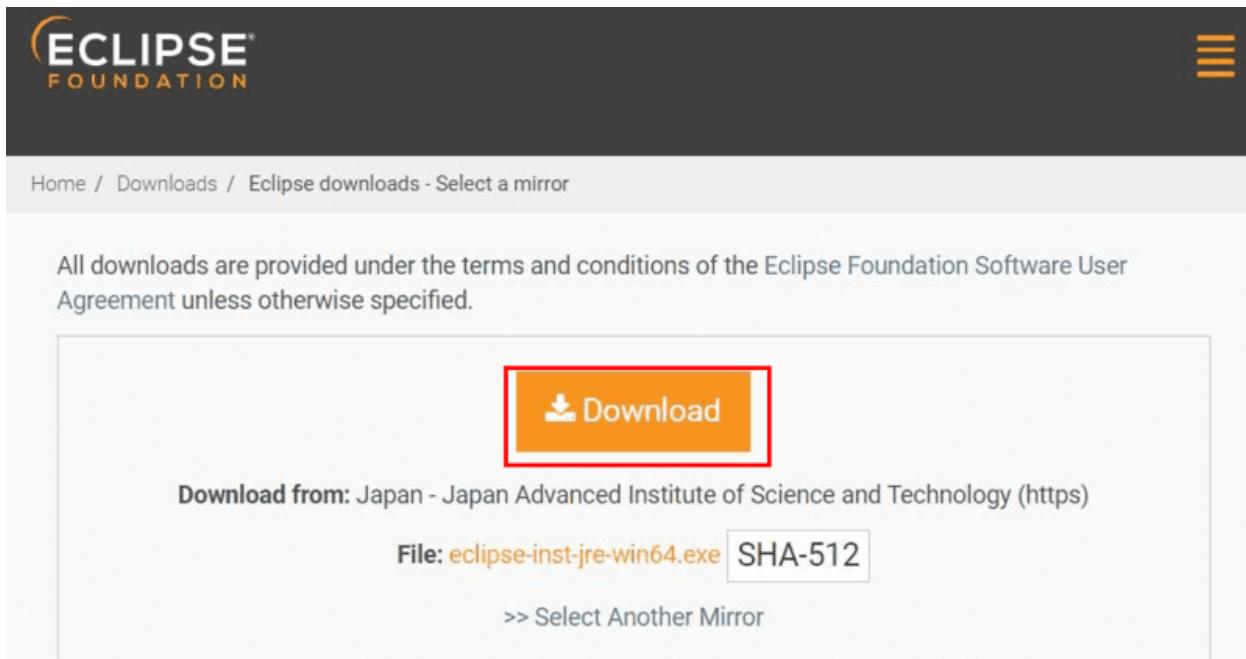


STEP 3: Download EXE

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



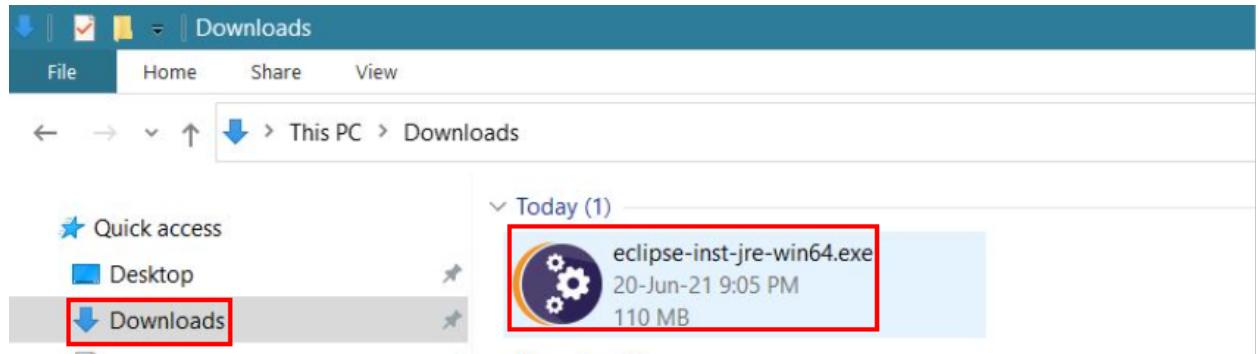
STEP 4: Then click on the "Download" button. After clicking on the download button the .exe file for the eclipse will be downloaded.



The screenshot shows the Eclipse Foundation Downloads page. At the top, the Eclipse Foundation logo is visible. Below it, a navigation bar includes 'Home', 'Downloads', and 'Eclipse downloads - Select a mirror'. A note states: 'All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.' A large orange 'Download' button is prominently displayed. Below it, the download information is shown: 'Download from: Japan - Japan Advanced Institute of Science and Technology (https)', 'File: eclipse-inst-jre-win64.exe', 'SHA-512', and a link to '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.

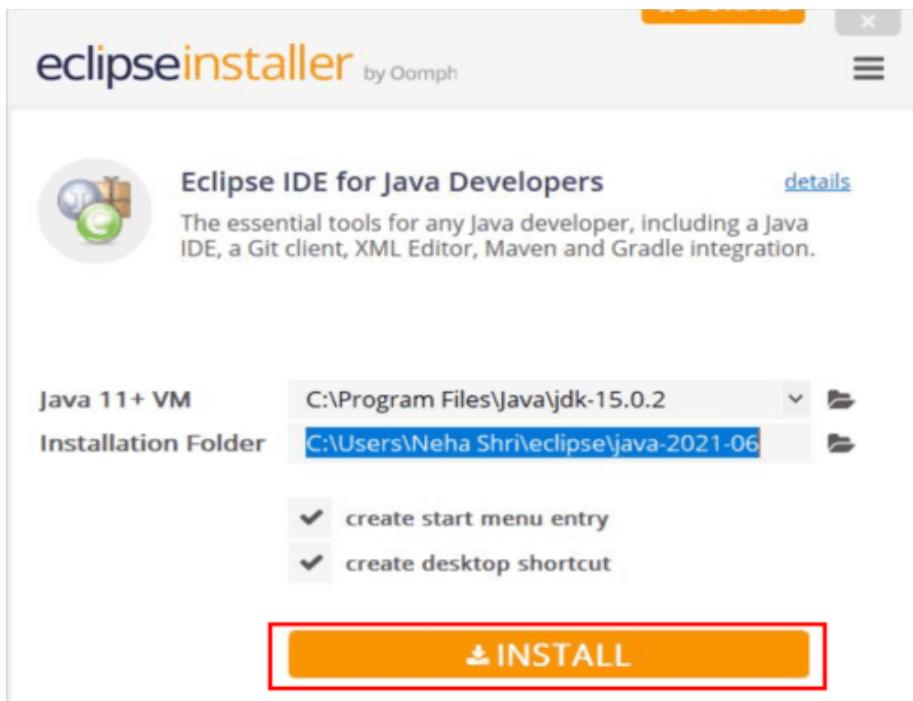


STEP 6: Install Eclipse

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

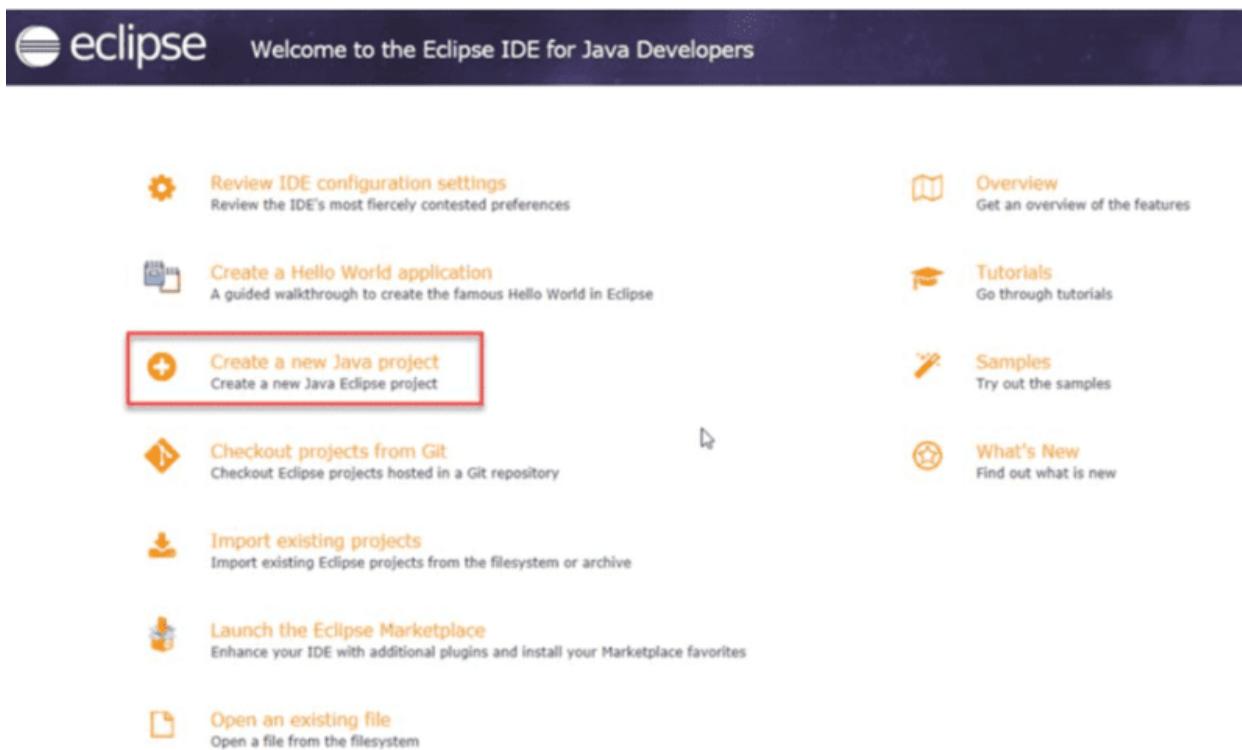


STEP 7: Then, click on the "Install" button.



Step 8: Create New Project

Now click on "Create a new Java project".



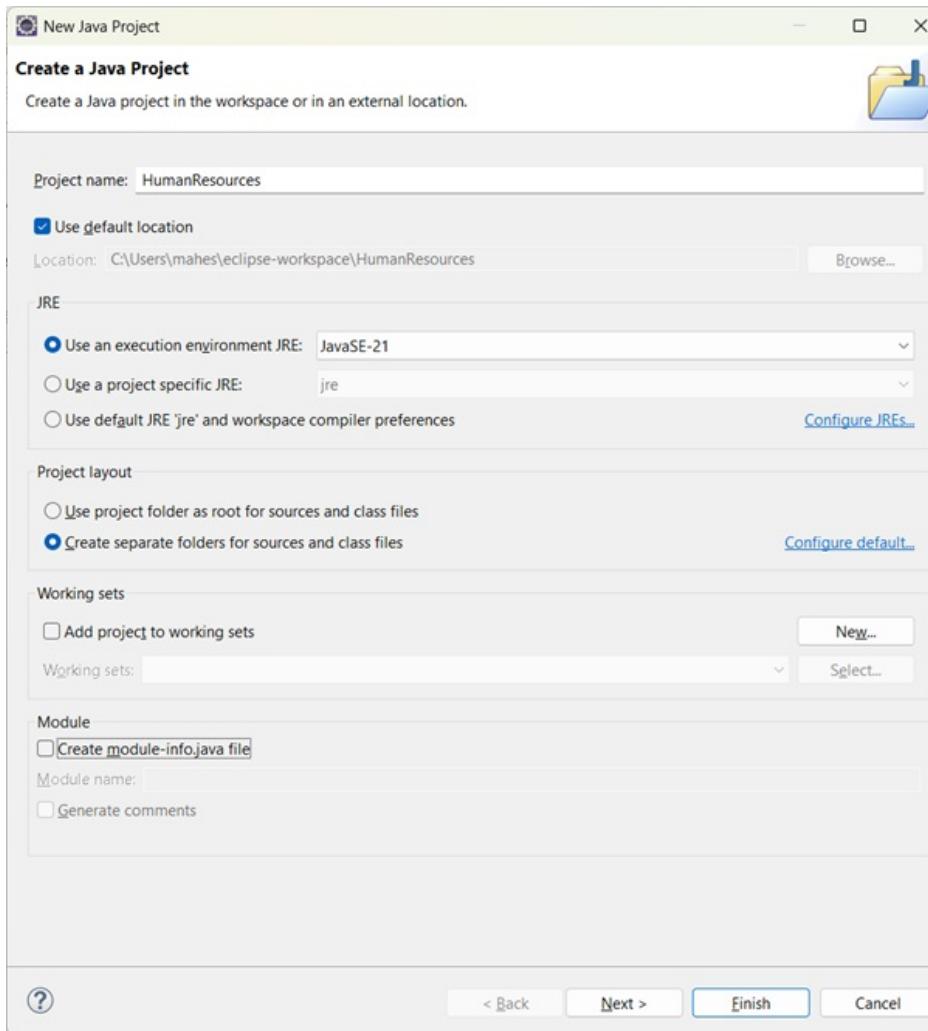
STEP 9: Create a new java project

- 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.

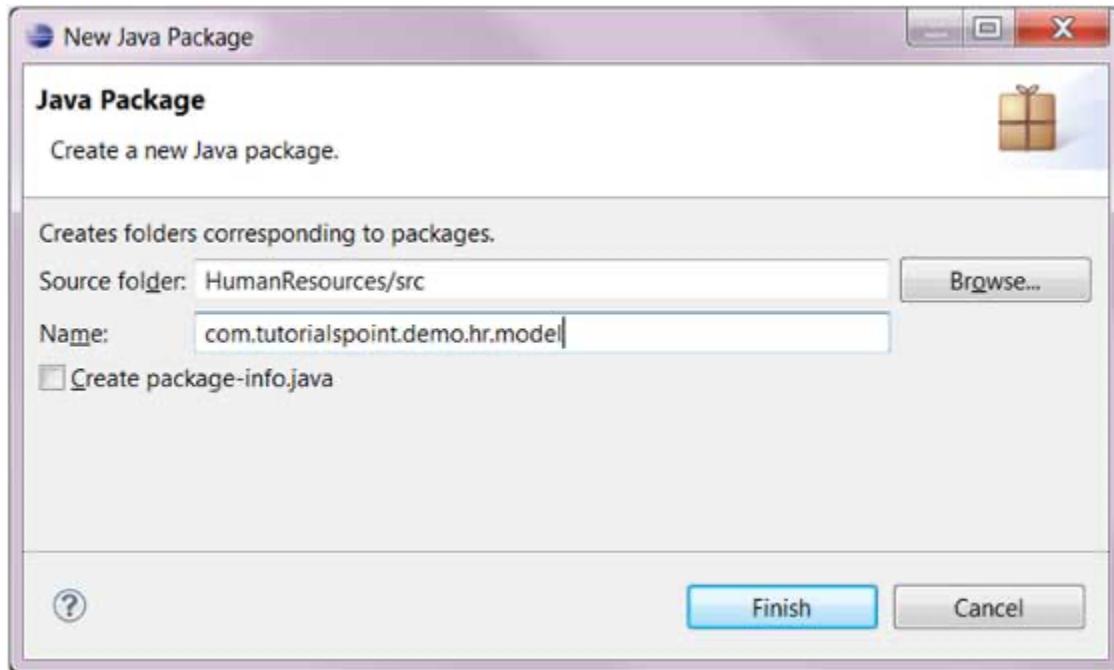


STEP 11: Create a new java package

- 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.

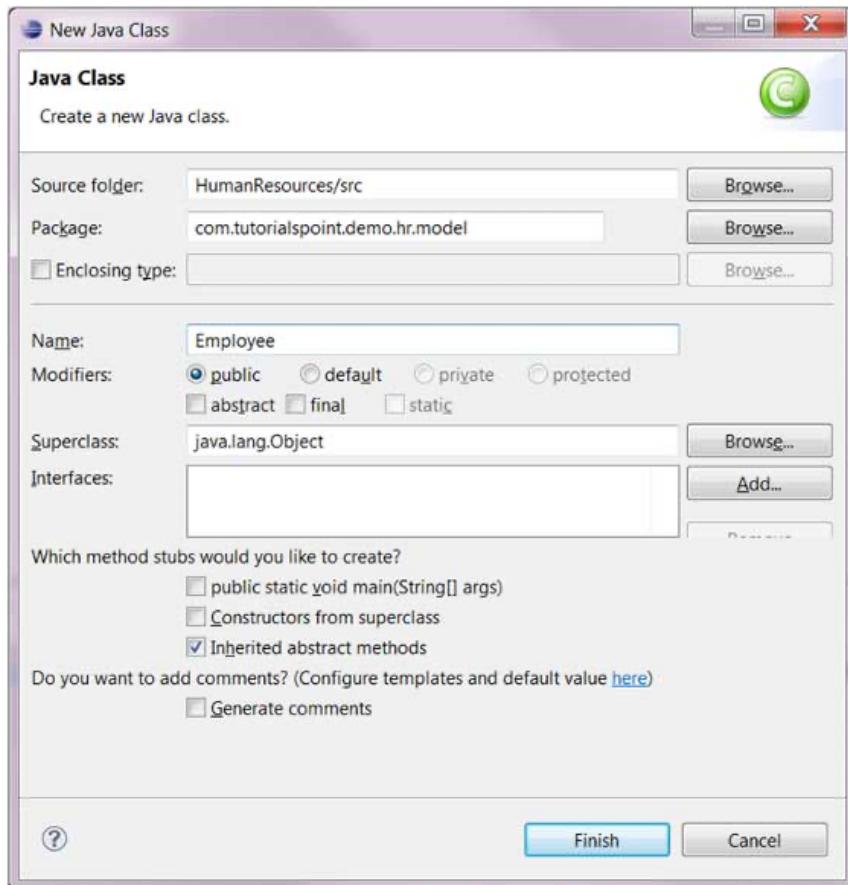


STEP 12: Create a New Java class.

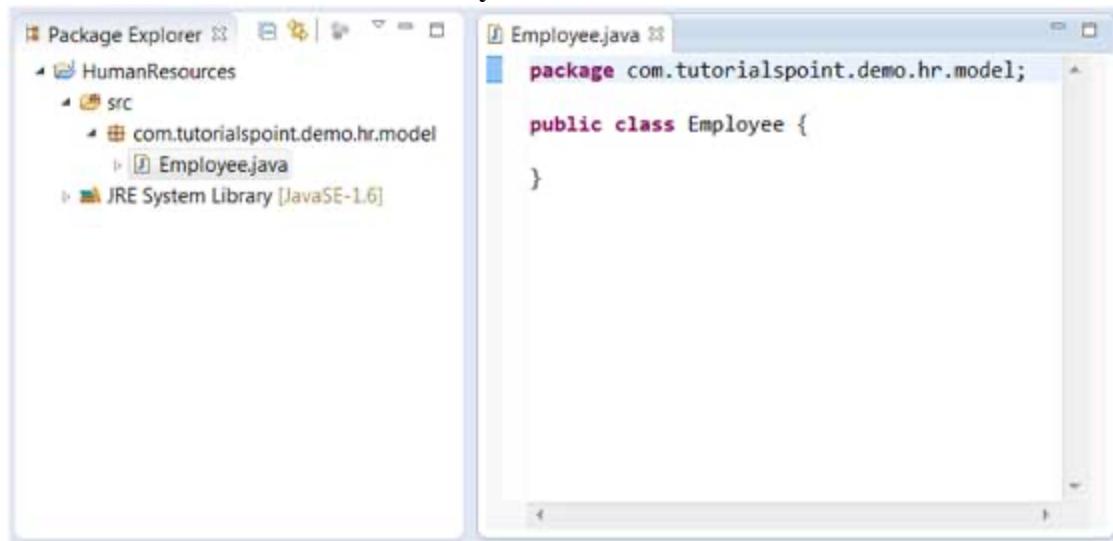
- By clicking on the File menu and selecting New → Class.
- By right clicking in the package explorer and selecting New → Class.
- By clicking on the class drop down button () and selecting class ().

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:

Hello World

```
Storage\0bf54446b42aa6e1dbffa55502133ecd\redhat.java\jdt_ws\JAVA\VS_2bcf11c9\bin HelloWorld  
Hello World  
parthipan@parthipan-MacBook-Air:~/JAVA/VS_2bcf11c9/bin$
```

Program 2: Display Personal Details

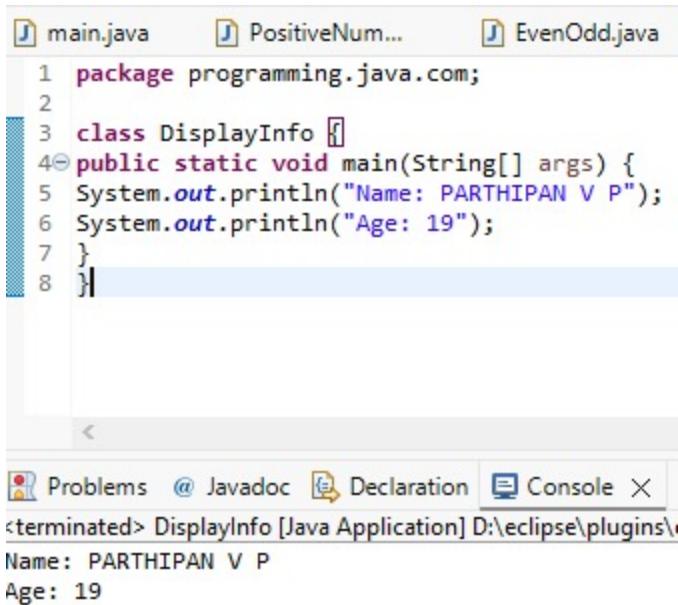
Source Code:

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

Output:

Name: : PARTHIPAN V P

Age: 19



The screenshot shows the Eclipse IDE interface. At the top, there are three tabs: 'main.java', 'PositiveNum...', and 'EvenOdd.java'. The 'main.java' tab is active, displaying the following Java code:

```
1 package programming.java.com;
2
3 class DisplayInfo {
4     public static void main(String[] args) {
5         System.out.println("Name: PARTHIPAN V P");
6         System.out.println("Age: 19");
7     }
8 }
```

Below the code editor is the 'Console' view, which shows the output of the program:

```
terminated> DisplayInfo [Java Application] D:\eclipse\plugins\Name: PARTHIPAN V P
Age: 19
```

Program 3: Addition of Two Numbers

Source Code:

```
class AddTwoNumbers {
    public static void main(String[] args) {
        int a = 6, b = 8;
        System.out.println("Sum = " + (a + b));
    }
}
```

Output:

Sum = 14



The screenshot shows a code editor with a tab for 'AddTwoNumbers.java'. The code is identical to the one in the previous section:

```
1 class AddTwoNumbers {
2     public static void main(String[] args) {
3         int a = 6, b = 8;
4         System.out.println("Sum = " + (a + b));
5     }
6 }
```

Next to the code editor is a toolbar with icons for copy, paste, share, and run. The 'Run' button is highlighted in blue. To the right of the code editor is a 'Output' panel showing the program's output:

```
Sum = 14
==== Code Execu
```

Program 4: Area of a Rectangle

Source Code:

```
class AreaRectangle {  
    public static void main(String[] args) {  
        int length = 10, breadth = 11;  
        System.out.println("Area = " + (length * breadth));  
    }  
}
```

Output:

Area=110



The screenshot shows a Java code editor interface. On the left, the file `AreaRectangle.java` is displayed with the following code:

```
1- class AreaRectangle {  
2-     public static void main(String[] args) {  
3     int length = 10, breadth = 11;  
4     System.out.println("Area = " + (length * breadth));  
5     }  
6 }
```

On the right, the **Output** panel shows the result of running the code: `Area = 110`. Below the output, a message indicates `== Code Executed`.

Program 5: Simple Interest Calculation**Source Code:**

```
class SimpleInterest {  
    public static void main(String[] args) {  
        int p = 1000;  
        int r = 7;  
        int t = 8;  
        int si = (p * r * t) / 100;  
        System.out.println("Simple Interest = " + si);  
    }  
}
```

Output:

```
1 ~ class SimpleInterest {  
2 ~   public static void main(String[] args) {  
3 ~     int p = 1000;  
4 ~     int r = 7;  
5 ~     int t = 8;  
6 ~     int si = (p * r * t) / 100;  
7 ~     System.out.println("Simple Interest = " + si);  
8 ~   }  
9 ~ }
```

Simple Interest = 560

== Code Execution Su

POST LAB EXERCISE

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

```
class StudentInfo {  
    public static void main(String[] args) {  
        System.out.println("Name: PARTHIPAN");  
        System.out.println("Department: Computer Science");  
    }  
}
```

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

```
class StudentInfo {  
    public static void main(String[] args)  
        System.out.println("Name: PARTHIPAN , Department: 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 will fail at runtime. The JVM cannot call the main() method because it requires an object of the class. Since the JVM does not create an object automatically, the program will not execute and results in a runtime error.

4. Why is Java called platform independent?

Java is called platform independent because it does not depend on any specific operating system. Java code is first converted into bytecode, which can run on any machine that has a JVM. Therefore, the same Java program can run on different platforms without any changes.

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

```
class Cube {  
    public static void main(String[] args) {  
        int num = 3;  
        int cube = num * num * num;  
        System.out.println("Cube of " + num + " 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		