

Santhosh TMA

24BCS247

CSE-A1

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 package used to develop Java programs. It includes the compiler, libraries, and tools needed to write, compile, and run Java applications.

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

JDK: Used for developing Java programs; includes JRE and development tools.

JRE: Used to run Java programs; includes JVM and libraries.

JVM: Executes Java bytecode and makes Java platform independent.

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

The **main()** method is the starting point of a Java program. Program execution begins from this method.

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.



The Eclipse Foundation
<https://www.eclipse.org> › downloads

Eclipse Downloads | The Eclipse Foundation

The Eclipse Foundation is a non-profit organization that hosts the Eclipse IDE.

STEP 2: Download Eclipse Installer

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

Eclipse IDE for Java Developers



353 MB

444,586 DOWNLOADS

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

STEP 3: Download EXE

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

Windows: [AArch64](#) | [x86_64](#)

macOS: [AArch64](#) | [x86_64](#)

.inux: [AArch64](#) | [riscv64](#) | [x86_64](#)

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

 Download

Download from: Japan - Yamagata University ([https](https://))

eclipse-java-2025-12-R-macosx-cocoa-aarch64.dmg

SHA-512

STEP 6: Install Eclipse

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

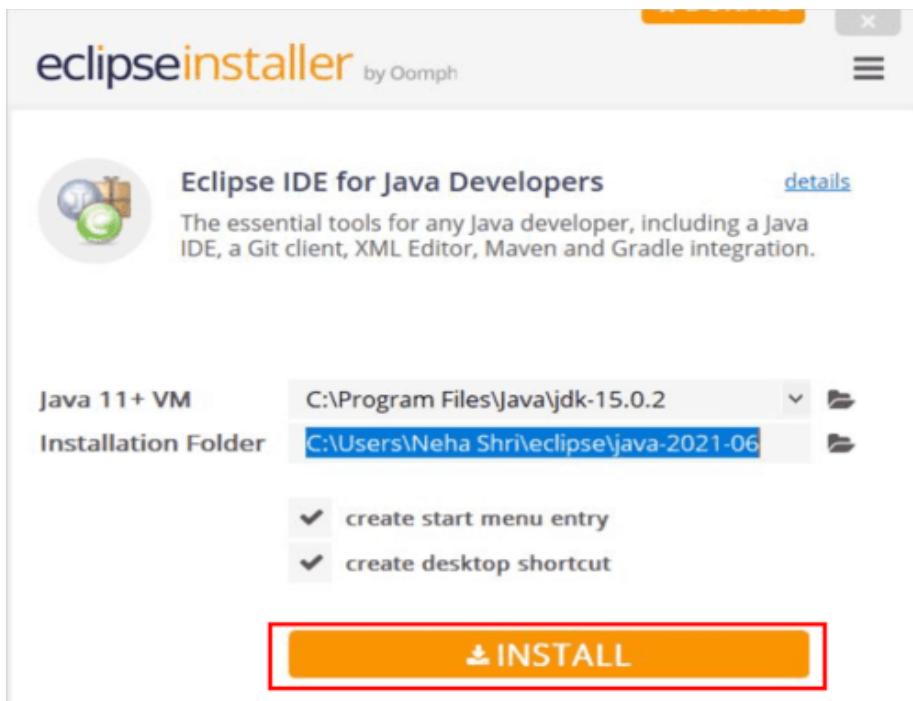
Eclipse IDE for Java Developers



353 MB 444,586 DOWNLOADS

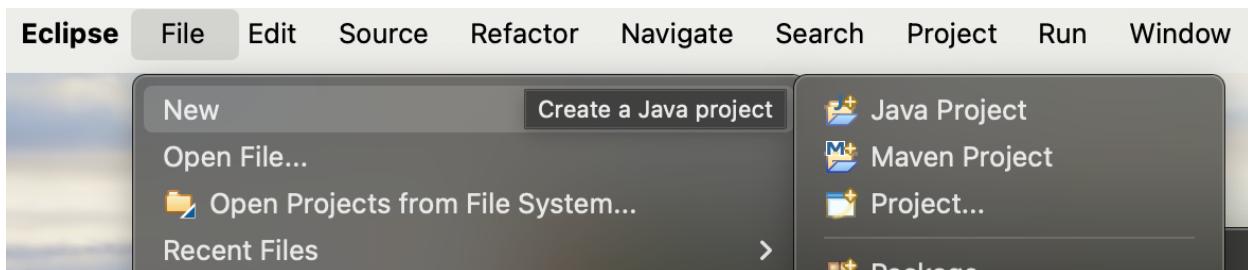
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration

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



Step 8: Create New Project

Now click on "Create a new Java project".

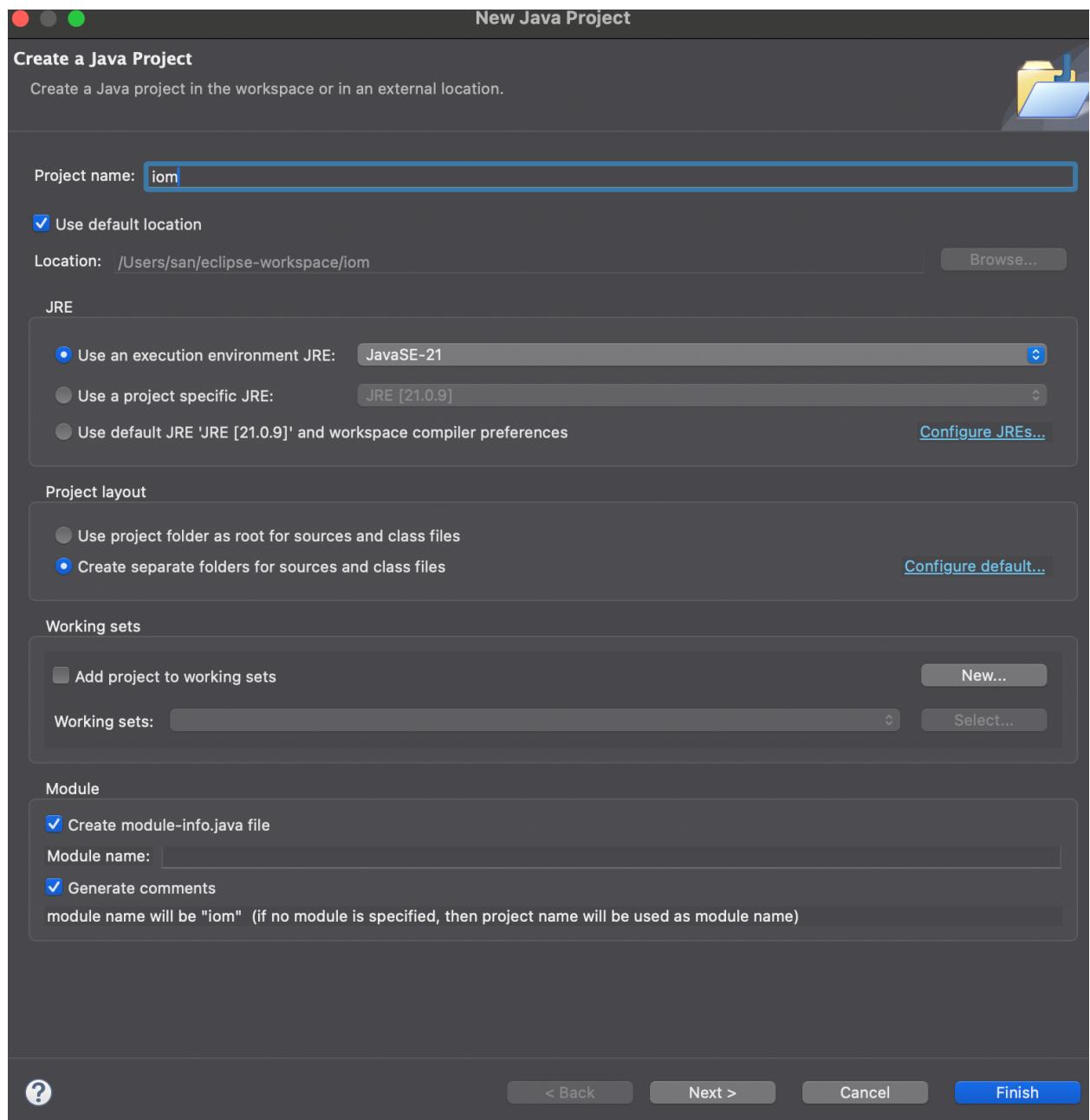


STEP 9: Create a new java project

- 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.



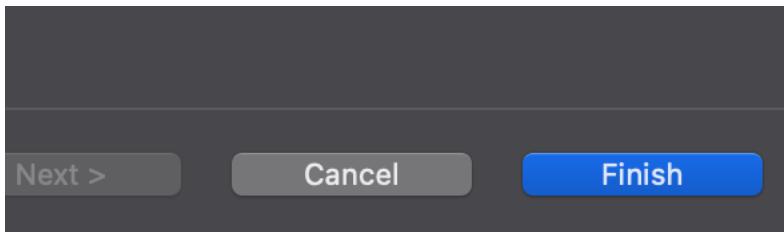
STEP 11: Create a new java package

- 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.

New Java Class

Java Class

Create a new Java class.

Source folder: iom/src

Package: iom

Enclosing type:

Name:

Modifiers:

public package private protected
 abstract final static
 none sealed non-sealed final

Superclass: java.lang.Object

Interfaces:

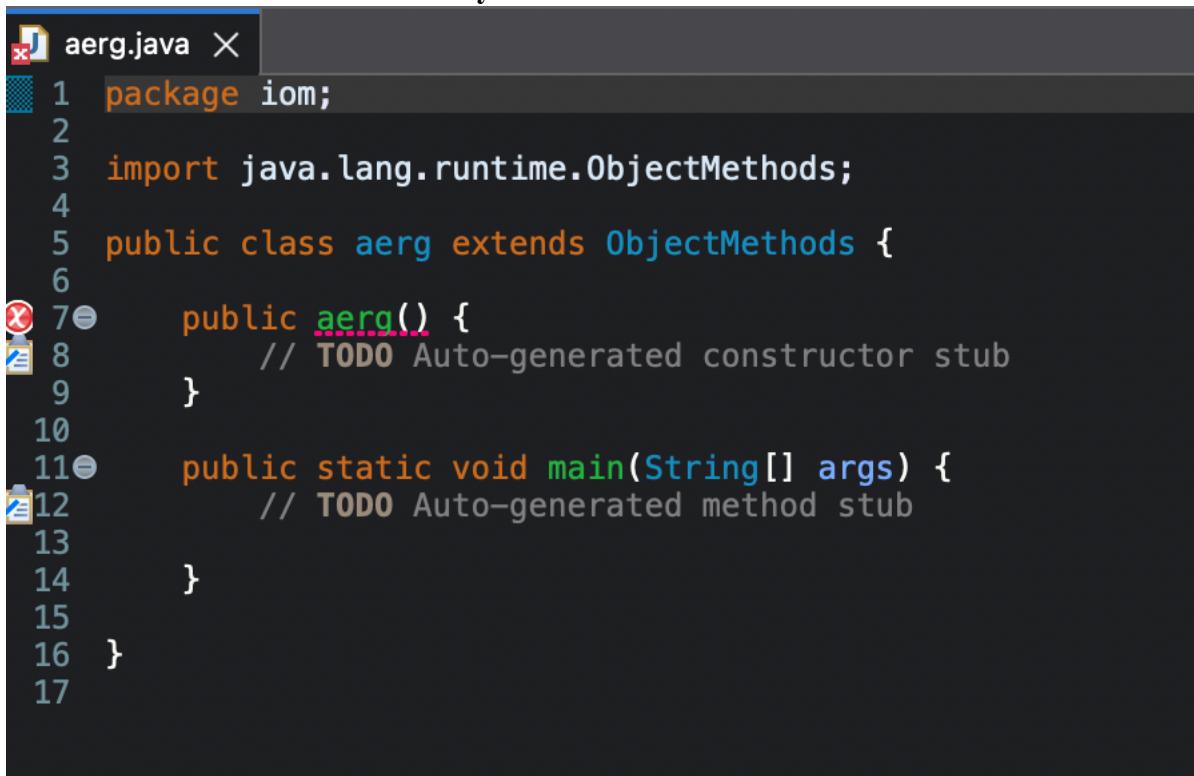
Which method stubs would you like to create?

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.



```
aerg.java X
1 package iom;
2
3 import java.lang.runtime.ObjectMethods;
4
5 public class aerg extends ObjectMethods {
6
7     public aerg() {
8         // TODO Auto-generated constructor stub
9     }
10
11    public static void main(String[] args) {
12        // TODO Auto-generated method stub
13    }
14
15
16}
17
```

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

```
Hello World
==== Code Execution Successful ====
```

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:

Name: Anitha

Age: 20

Name: Anitha

Age: 20

==== Code Execution Successful ===

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:

Sum = 30

```
Sum = 30
```

```
==== Code Execution Successful ===
```

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:

```
Area = 50
```

```
Area = 50
```

```
==== Code Execution Successful ===
```

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:

Simple Interest = 100

```
Simple Interest = 100  
  
==== Code Execution Successful ===
```

POST LAB EXERCISE

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

```
1 -> class Main {  
2 ->     public static void main(String[] args) {  
3 ->         System.out.println("Name: Santhosh TMA");  
4 ->         System.out.println("Department: Computer  
5 ->             Science");  
6 ->     }  
7 -> }  
8
```

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

```
1 - class Main {  
2 -     public static void main(String[] args) {  
3 -         System.out.print("Santhosh TMA ");  
4 -         System.out.print("Computer Science");  
5 -     }  
6 }  
7
```

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

The program will compile, but the JVM cannot call `main()` and throws a runtime error because `static` is required to run without creating an object.

4. Why is Java called platform independent?

Because Java programs compile into bytecode, which can run on any system that has a JVM (Write Once, Run Anywhere).

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

```
1 - class Main {  
2 -     public static void main(String[] args) {  
3 -         int n = 3;  
4 -         int cube = n * n * n;  
5 -         System.out.println(cube);  
6 -     }  
7 }  
8
```

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		