Experiment Number : 01 **Date:**

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 stands for Java Development Kit, a software package essential for developing Java applications. It includes tools like compilers and debuggers, making it required for writing, compiling, and running Java code.

2. Difference between JDK, JRE, and JVM.
   - JDK: JDK (Java Development Kit) is a software development kit containing tools like the javac compiler for developing, compiling, and debugging Java applications.
   - JRE: JRE (Java Runtime Environment) provides the libraries and components needed to run Java applications, including the JVM, but lacks development tools.
   - JVM: JVM (Java Virtual Machine) is an abstract machine that executes Java bytecode, enabling platform independence by converting it to machine code.
3. What is the purpose of the main() method in Java?

   The main() method serves as Java's entry point for program execution, defined as public static void main(String[] args). JVM starts running the program from here upon launch.

IN LAB EXERCISE

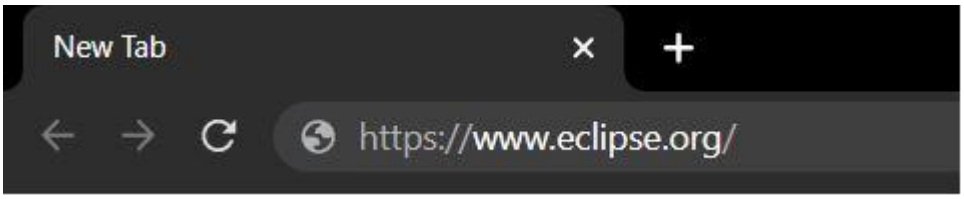
Objective:

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.

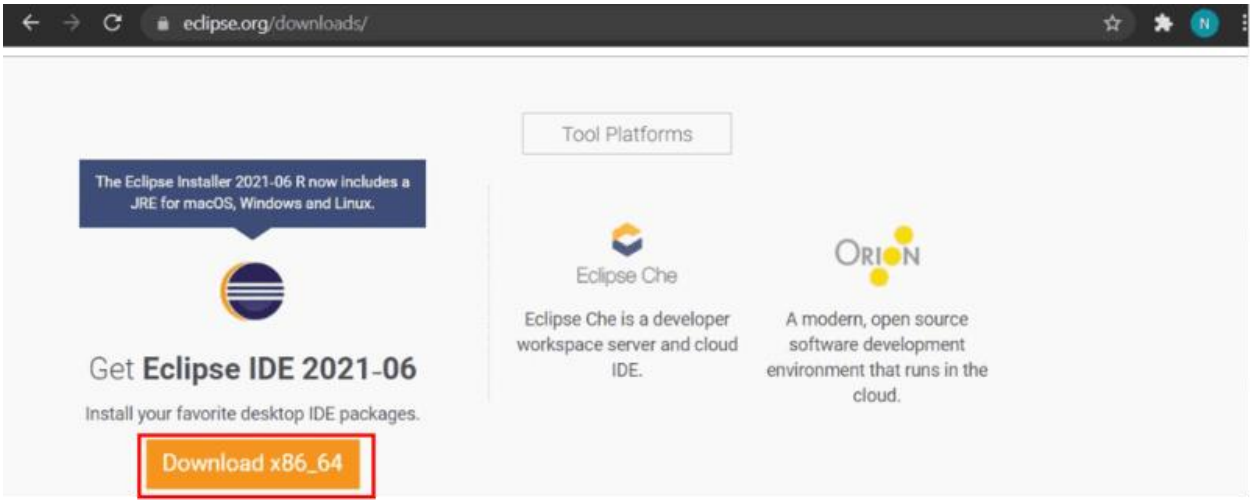


STEP 2: Download Eclipse Installer

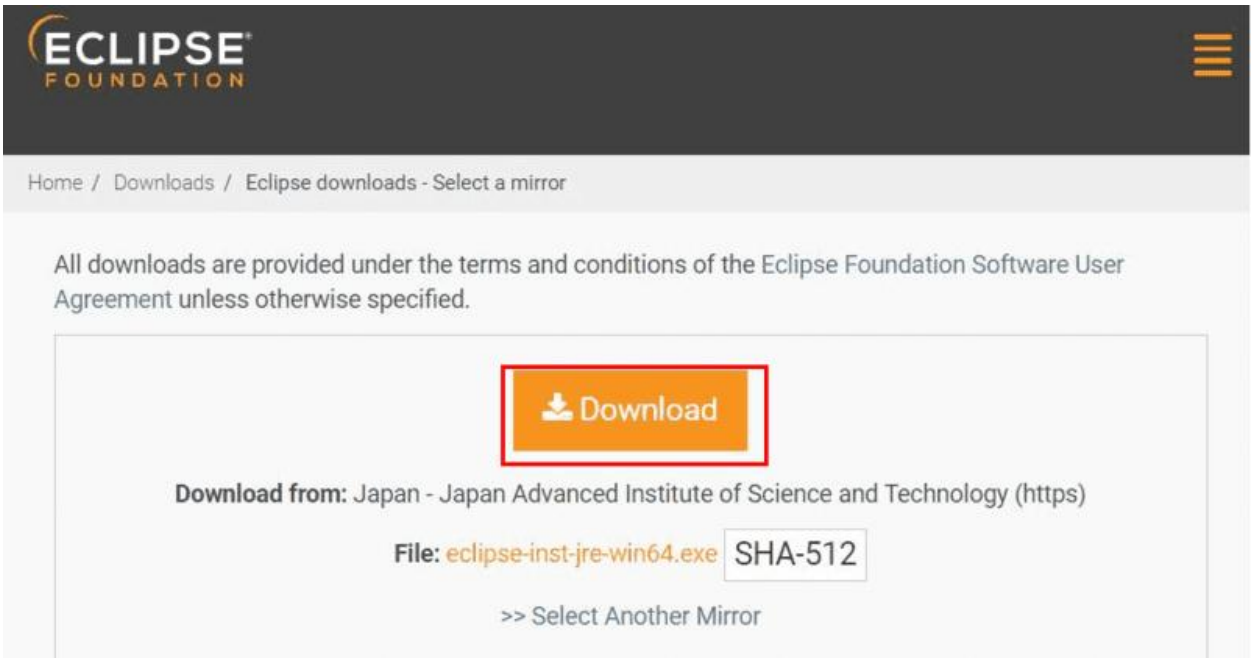
- Then, click on the "Download" button to download Eclipse IDE.

STEP 3: Download EXE

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

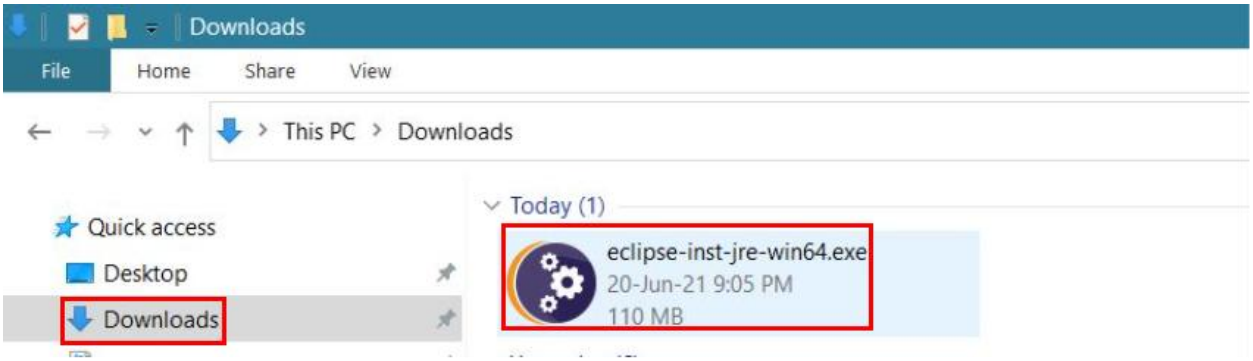


STEP 4: Then click on the "Download" button. After clicking on the download button the .exe file for the eclipse will be downloaded.



STEP 5: Open Download EXE

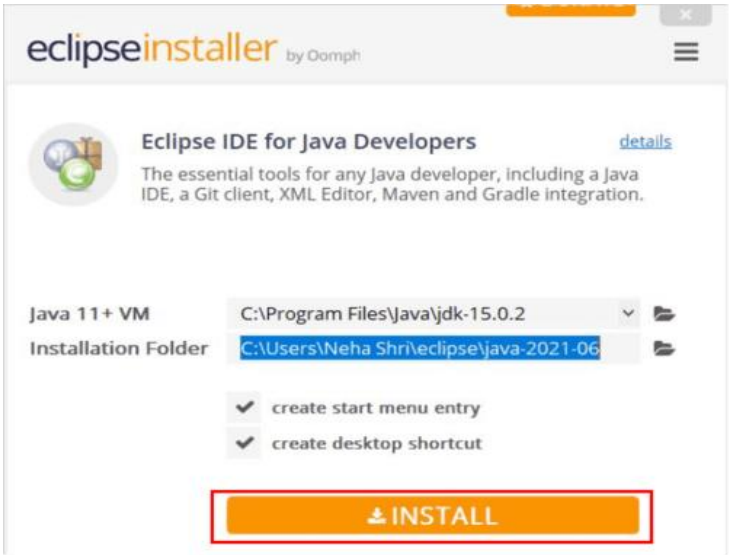
- Now go to File Explorer and click on "Downloads" after that click on the "*eclipse-inst-jre-win64.exe*" file for installing Eclipse IDE**.**



STEP 6: Install Eclipse

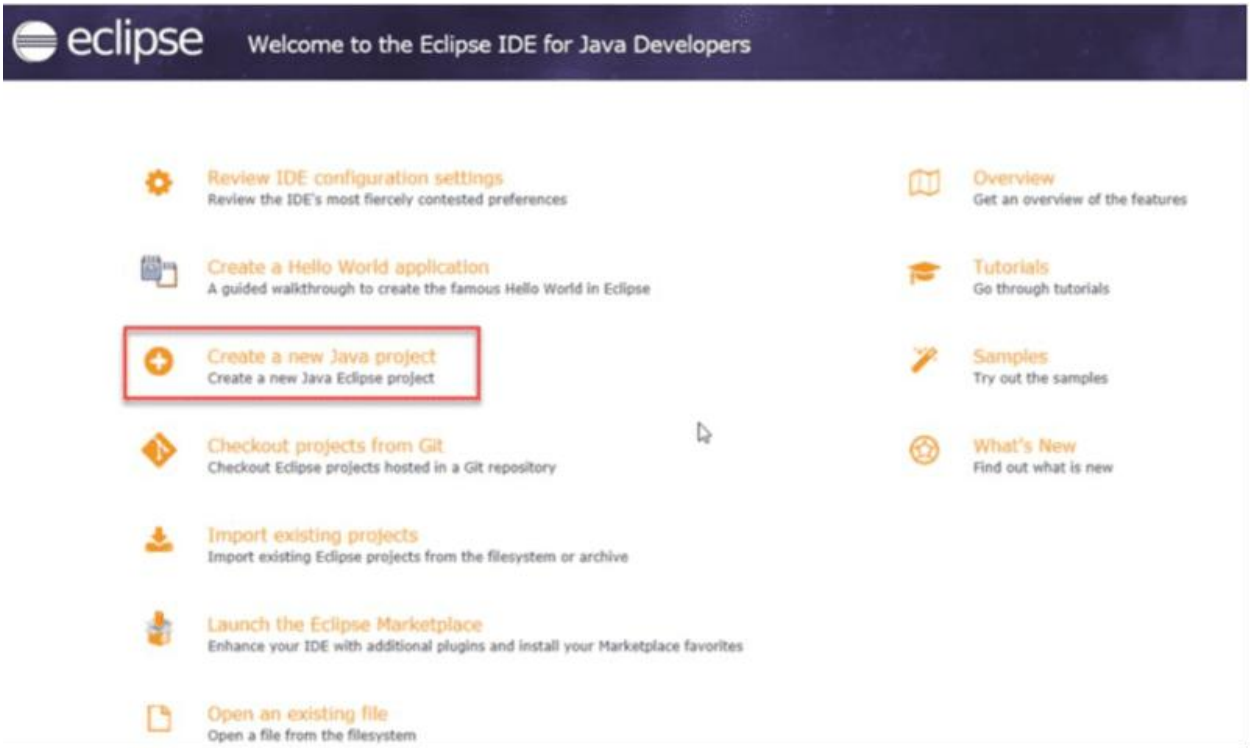
- Then, click on "Eclipse IDE for Java Developers".

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



Step 8: Create New Project

Now click on "Create a new Java project".

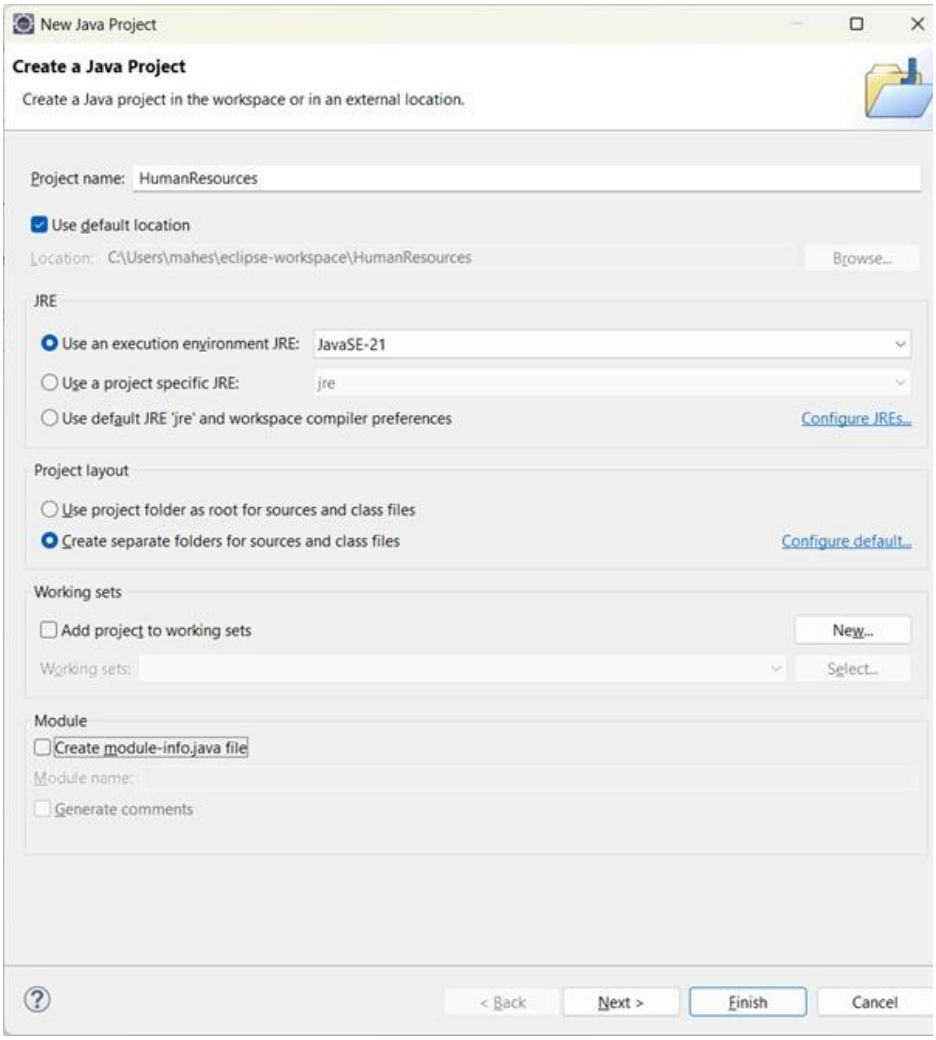


STEP 9:Create a new java project

- 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.

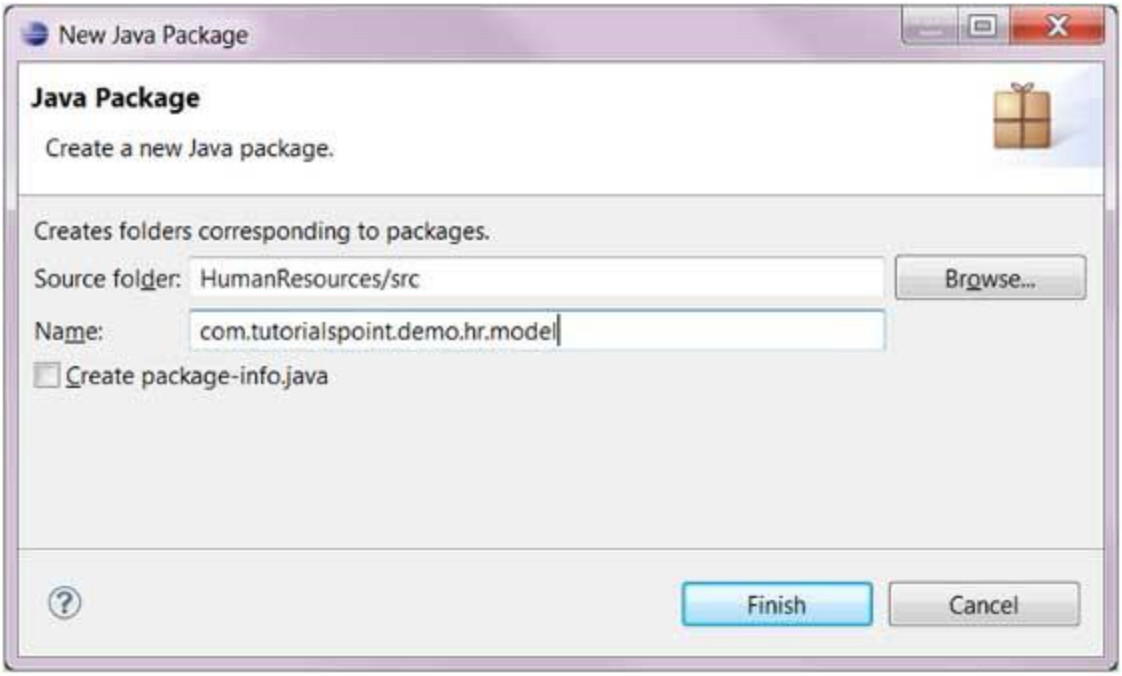


STEP 11: Create a new java package

- 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.

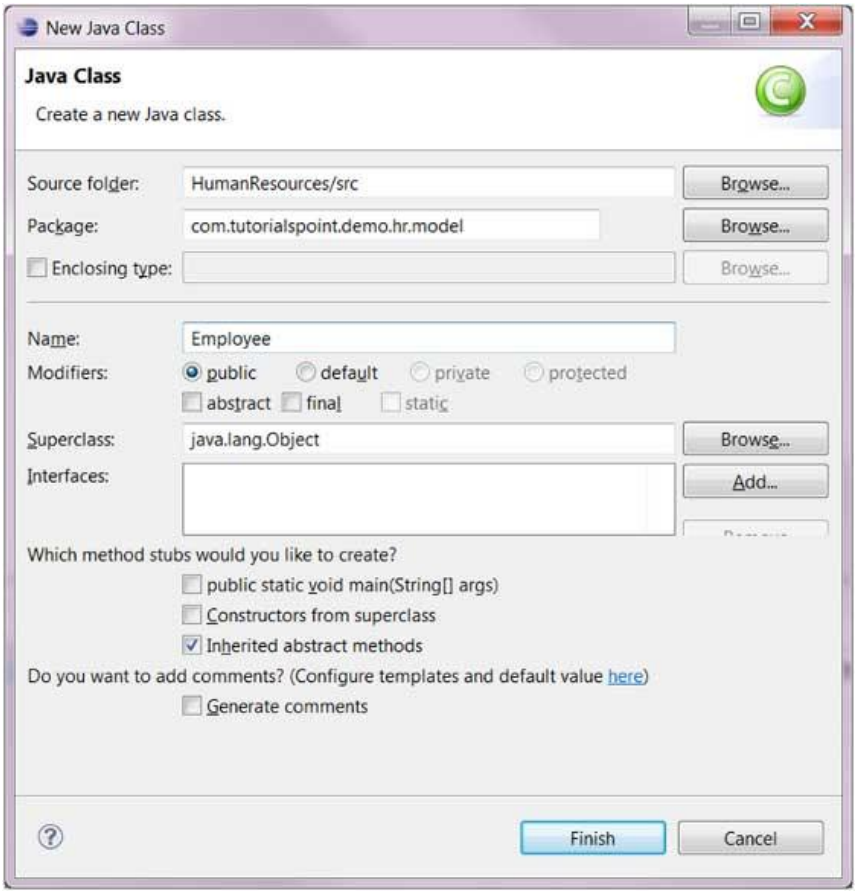


STEP 12: Create a New Java class.

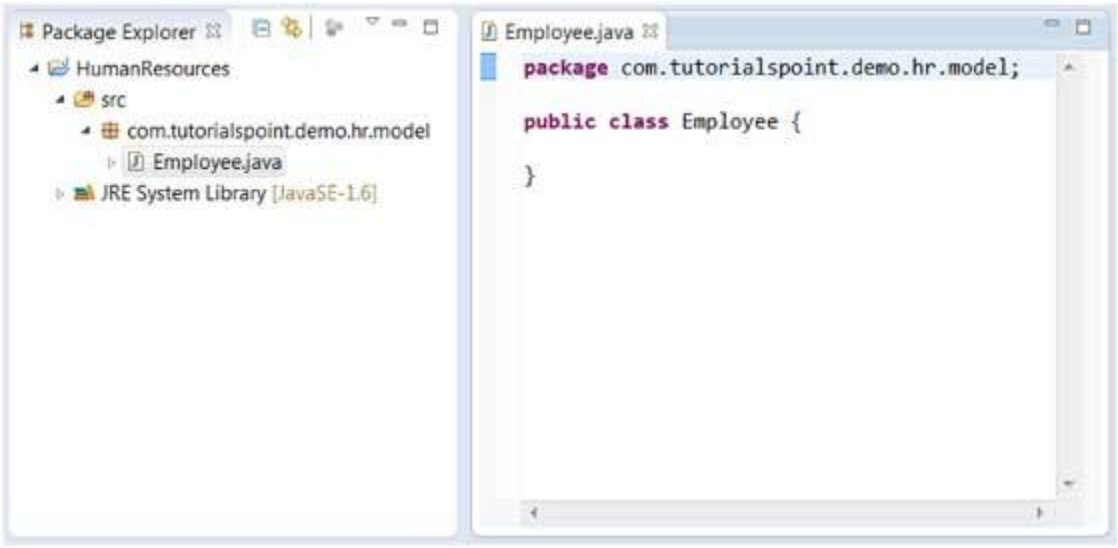
- By clicking on the File menu and selecting New → Class.
- By right clicking in the package explorer and selecting New → Class.
- By clicking on the class drop down button ( ) and selecting class ( ).

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.



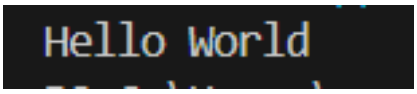
BASIC PROGRAMS:

Program 1: Hello World Program

Code:

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

Output:

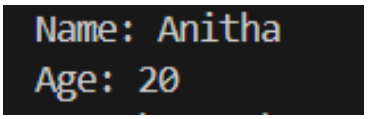


Program 2: Display Personal Details

Code:

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

Output:

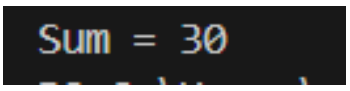


Program 3: Addition of Two Numbers

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

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

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

   Output:
```
Name: Naveena
Department: Computer Science Engineering
```

2. Modify the program to print the output in same line.
   Code:
```
public class Details {
         public static void main(String[] args) {
      System.out.print("Name: Naveena ");
      System.out.print("Department: Computer Science Engineering");
   }
}
```

   Output:
```
Name: Naveena 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:
   - The JVM starts execution from main()
   - Without static, JVM would need to create an object first
   - But JVM cannot create an object without calling main()

4. Why is Java called platform independent?
   Java is called **platform independent** because:
   - Java programs are compiled into **bytecode**
   - Bytecode can run on any system that has a **JVM (Java Virtual Machine)**
   - Same program runs on **Windows, Linux, and Mac** without modification

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

Code:

```
public class Cube {
        public static void main(String[] args) {
    int n = 10;
    int cube = n * n * n;
    System.out.println("Cube of " + n + " is: " + cube);
  }
}
```

Output:

```
Cube of 10 is: 1000
```

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		