University of Petroleum and Energy Studies

Session: 2024-2025

Semester: IV

BTech Computer Science

(OOPS – Lab-1)

Introduction to Java Environment

Submitted by

Daksh Mehrotra
Batch: 2 CCVT

Sap Id: 500125960

Roll no: R2142231932

Submitted to

Satyam Sir

Q1. Explore and understand the role of JDK, JRE and JVM.

A1. **1. JVM (Java Virtual Machine):**

- **Role**: The JVM is the runtime environment that executes Java bytecode. It abstracts the underlying hardware and operating system, enabling Java's "write once, run anywhere" capability.

- **Key Responsibilities**:
  - Converts bytecode into machine-specific code using the Just-In-Time (JIT) compiler.
  - Handles memory management (via the garbage collector).
  - Ensures runtime error handling.
  - Provides a secure execution environment by verifying the bytecode.

- **Lifecycle**:
  - Class loader loads .class files (bytecode).
  - Bytecode is verified for security and correctness.
  - Bytecode is interpreted or compiled into native machine code for execution.

2. JRE (Java Runtime Environment)

- **Role**: JRE provides the libraries and components required to run Java applications. It includes the JVM and supporting libraries but does not have development tools.

- **Components**:
  - **JVM**: Executes Java programs.
  - **Core Libraries**: Provide essential functionality like I/O, networking, and utilities.
  - **Other Runtime Components**: Includes classes for graphics, database connectivity, etc.
- **Use**: When you just want to run Java applications, you only need the JRE installed.

3. JDK (Java Development Kit):

- **Role**: JDK is the full development kit needed for creating, compiling, and debugging Java programs. It includes the JRE, development tools, and additional resources.

- **Components**:
    - **JRE**: To run compiled Java applications.
    - **Development Tools**:
        - javac: Java compiler for converting source code (.java) into bytecode (.class).
        - java: Launcher tool to run Java applications.
        - javadoc: Documentation generator.
        - jdb: Debugger.

    - **Libraries**: APIs and class libraries necessary for development.
- **Use**: If you're developing Java programs, you need the JDK installed.

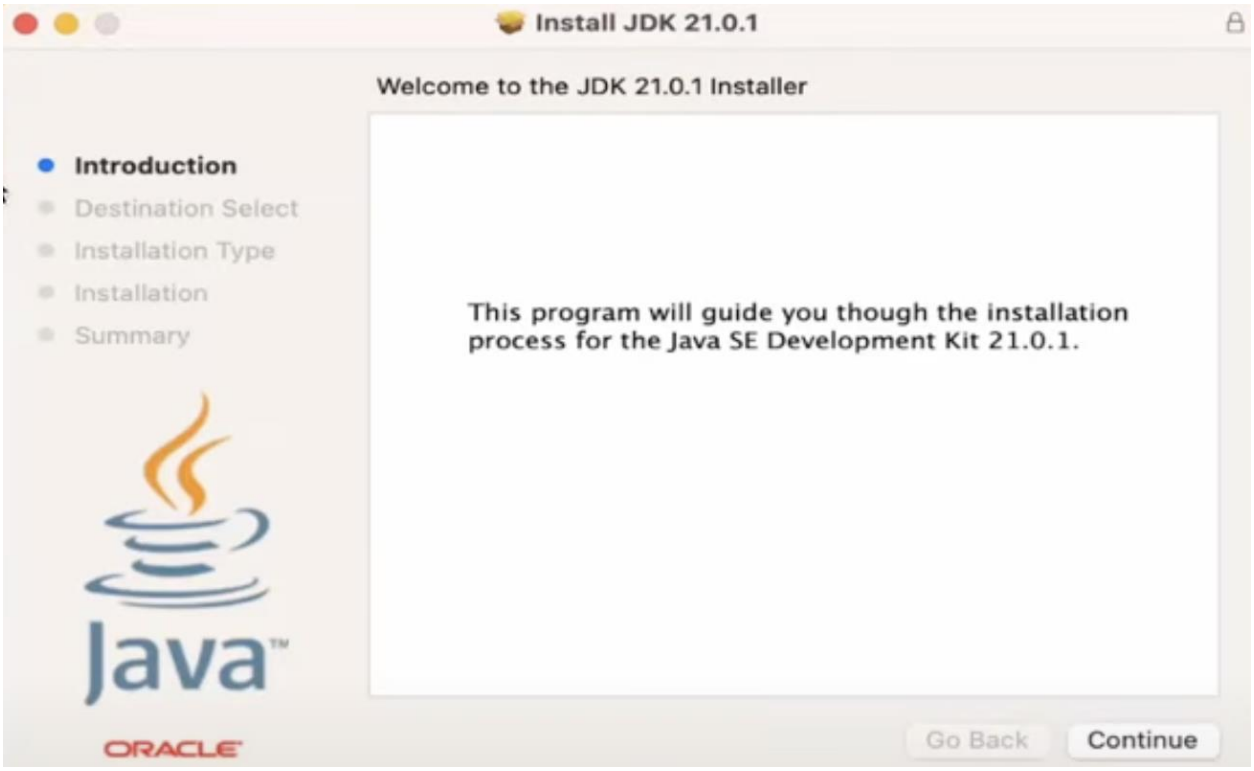
Q2. Install latest available JDK and verify the Java Environment.
A2.

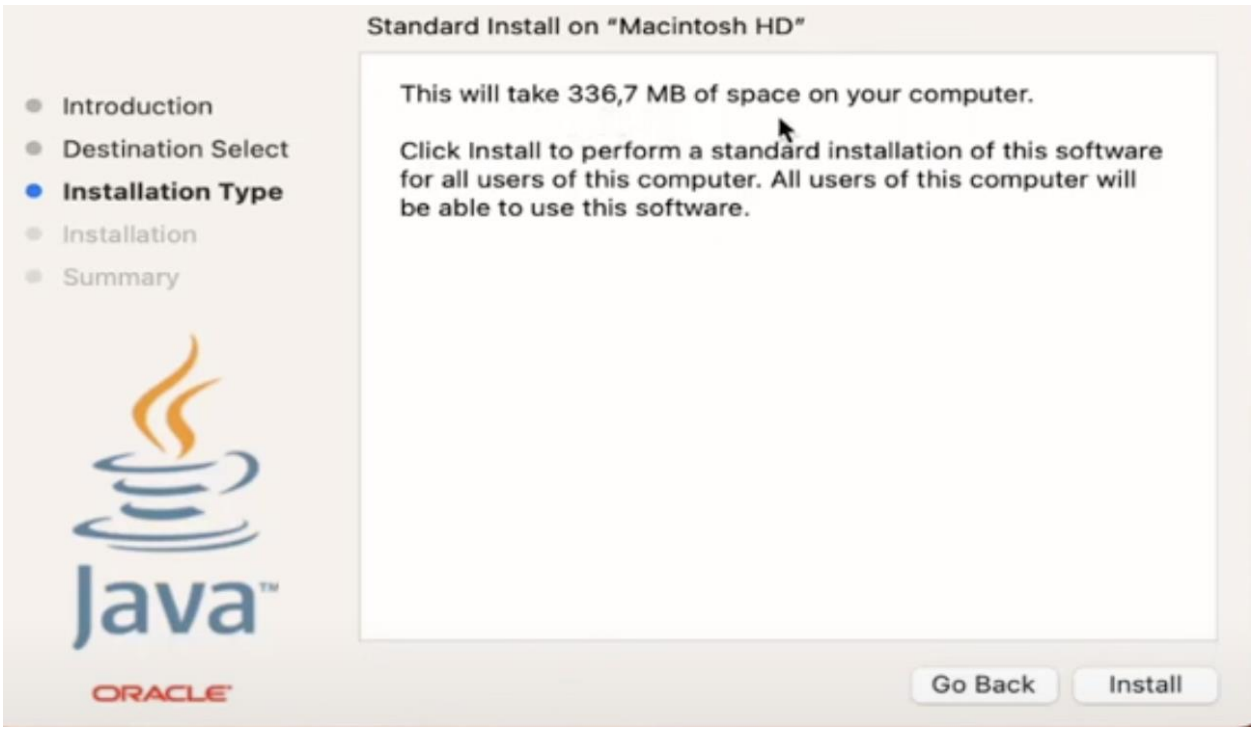
Linux **macOS** **Windows**

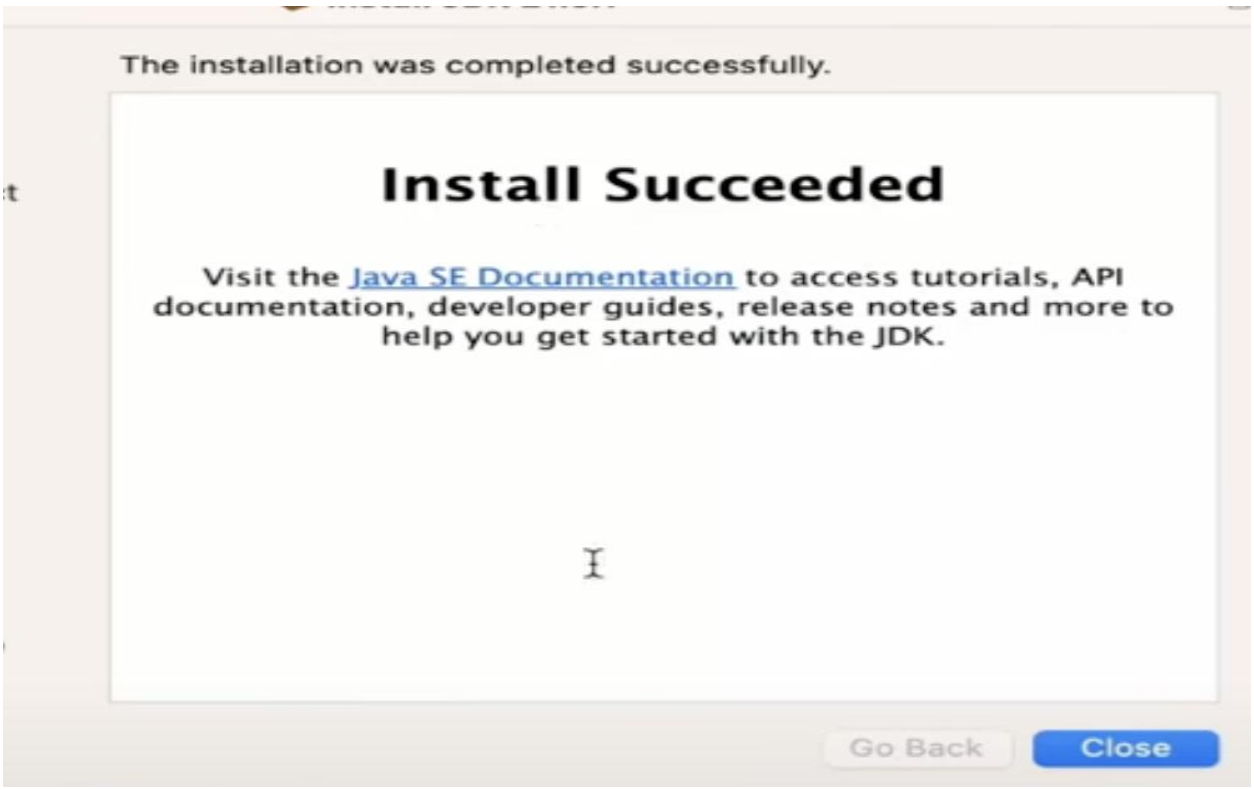
Product/file description	File size	Download
ARM64 Compressed Archive	226.27 MB	https://download.oracle.com/java/23/latest/jdk-23_macos-aarch64_bin.tar.gz (sha256)
ARM64 DMG Installer	225.76 MB	https://download.oracle.com/java/23/latest/jdk-23_macos-aarch64_bin.dmg (sha256)
x64 Compressed Archive	228.97 MB	https://download.oracle.com/java/23/latest/jdk-23_macos-x64_bin.tar.gz (sha256)
x64 DMG Installer	228.47 MB	https://download.oracle.com/java/23/latest/jdk-23_macos-x64_bin.dmg (sha256)

Documentation Download


JDK 21.0.1
JDK 21.0.1.pkg


Install JDK 21.0.1
Welcome to the JDK 21.0.1 Installer
Introduction
Destination Select
Installation Type
Installation
Summary
This program will guide you though the installation process for the Java SE Development Kit 21.0.1.
Java™
ORACLE
Go Back
Continue





Q3. Create a Sample Hello World Program using simple text editor (e.g. Notepad) and show the steps to compile and execute the program using command prompt.

A3.

Step 1: Write the below code using a text editor or an online compiler:

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

Step 2: Open Command Prompt
Navigate to the directory where the file is saved:

```
cd C:\JavaPrograms
```

Step 3: Compile the Program

```
javac HelloWorld.java
```

Step 4: Execute the Program

Output:

```
Hello, World!
```

Q4. Display your name and complete address in different lines.

A4. The code for the above is as follows:

DisplayAddress.java

```java
public class DisplayAddress {
    public static void main(String[] args) {
        System.out.println("Daksh Mehrotra");
        System.out.println("UPES Bidholi");
        System.out.println("Dehradun");
    }
}

```

Output: