Week 1: Java Basics and Control Statements

Day 1: Introduction to Java

1. Java History and Features

History of Java:

- Java was developed by James Gosling and his team at Sun Microsystems in the mid-1990s. It was initially called "Oak" but was renamed to "Java" in 1995.
- Java was designed with the goal of being platform-independent, secure, and simple to use.
- Sun Microsystems was later acquired by Oracle, and Oracle now maintains Java.

Key Features of Java:

- **Simple:** Java has a clean syntax, similar to C++, but with fewer low-level features like pointers and operator overloading.
- **Object-Oriented:** Everything in Java revolves around classes and objects.
- **Platform Independent:** Java's "Write Once, Run Anywhere" (WORA) philosophy allows Java programs to run on different operating systems without modification.
- **Secure:** Java provides a secure environment by eliminating pointers and providing built-in security mechanisms.
- **Multithreaded:** Java has built-in support for multithreading, allowing programs to perform multiple tasks at the same time.
- **High Performance:** Java achieves high performance through the use of Just-In-Time (JIT) compilers.
- **Distributed:** Java provides networking libraries that make writing distributed applications easy.
- **Robust:** Java's strong memory management, exception handling, and type checking help in building reliable software.

2. Understanding the JDK, JRE, and JVM

Java Development Kit (JDK):

- A software development environment used for developing Java applications. It includes the JRE, a compiler, and various development tools like the debugger.

Java Runtime Environment (JRE):

- The part of the JDK that contains libraries and other files that the JVM needs to execute Java applications. It allows running Java programs on your machine.

Java Virtual Machine (JVM):

- The JVM is the core component of Java's platform-independent capabilities. It translates compiled Java bytecode into machine language and runs it on the local machine.

3. Installing and Setting Up Java

Step-by-Step Process:

1. **Download the JDK:**
   - Go to Oracle's official website and download the latest JDK for your operating system.
   - Download Link
2. **Install the JDK:**
   - Follow the installation instructions for your operating system.
3. **Set Environment Variables (For Windows):**
   - Go to the System Properties > Environment Variables.
   - Add the path to the JDK bin directory in the system's `Path` variable.
4. **Verify Installation:**
   - Open Command Prompt (Windows) or Terminal (Mac/Linux).
   - Type `java -version` and `javac -version` to check if Java is correctly installed.

4. First Java Program: "Hello World"

Writing Your First Java Program:

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

Explanation:

- **public class HelloWorld**: Defines a public class named `HelloWorld`. In Java, every application must have at least one class.
- **public static void main(String[] args)**: The main method is the entry point of any Java program.
- **System.out.println("Hello, World!")**: Prints "Hello, World!" to the console.

Compiling and Running the Program:

1. Open a terminal or command prompt.
2. Navigate to the directory where your file is saved.
3. Compile the program:`bash`
   Copy code

```
javac HelloWorld.java
```

4

5 Run the program:bash
Copy code

```
java HelloWorld
```

6

Output:
Copy code

```
Hello, World!
```

7

5. Structure of a Java Program

Java programs follow a specific structure. Here are the core components:

1 **Classes:**

- Java is an object-oriented language, and classes are blueprints for objects.
- A class may contain fields (variables) and methods (functions).

2 **Methods:**

- Methods define the behavior of objects. Every Java program must have a `main` method, which acts as the starting point.
- Syntax:java
  Copy code

```
returnType methodName(parameters) {
    // method body
}
```

3 **Main Function:**

- Every Java application must contain a `main` method.
- Signature of the `main` method:java
  Copy code

```
    public static void main(String[] args) {
        // code
    }
```

Exercises:

Exercise 1: Basic Setup

1. Install the JDK on your machine.
2. Verify the installation by running `java -version` and `javac -version` commands.

Exercise 2: "Hello, World!" Program

- Write a simple program that prints "Hello, World!" to the console.
- Modify the program to print your name after "Hello, World!"

Exercise 3: Understanding Program Structure

1. Create a new class `Greeting` and write a program that prints "Good Morning!".
2. In the same class, add another method called `eveningGreeting` that prints "Good Evening!".
3. Call this method from the `main` method.

Exercise 4: Using Comments

- Add single-line and multi-line comments to explain each section of your "Hello, World!" code.

Exercise 5: Research Task

- Research the role of JIT (Just-In-Time) compilation in Java. Write a brief explanation.