Introduction to Java

CSE 2101: Object Oriented Design and Design Patterns

Creating the environment

- JDK (Download Link) (Install this first)
- NetBeans (Download Link) (You can choose other IDEs such as Eclipse)

What is Java

- Java is:
  - platform independent programming language
  - similar to C++ in syntax
- Compiled and Interpreted
- A general purpose, high-level programming language with support for object-oriented programming.
- A collection of wide-ranging application programming interfaces (APIs).
- A complete set of development tools.

JDK – Java Development Kit

- A software package
- To create Java-based applications
- Provides the environment to develop and execute(run) the Java program
- Includes two things
  - Development Tools such as the java compiler (Provides the environment)
  - JRE (Executes java program)

JRE – Java Runtime Environment

- An installation package
- It physically exists
- Provides environment to **only run (not develop)**
- Part of the JDK
- It contains a set of libraries + other files that JVM uses at runtime

JVM – Java Virtual Machine

- An abstract machine
- It is called a virtual machine because it doesn't physically exist
- Part of both JDK and JRE
- JVM is responsible for **executing the java program line by line**
- The interpreter

Comparison

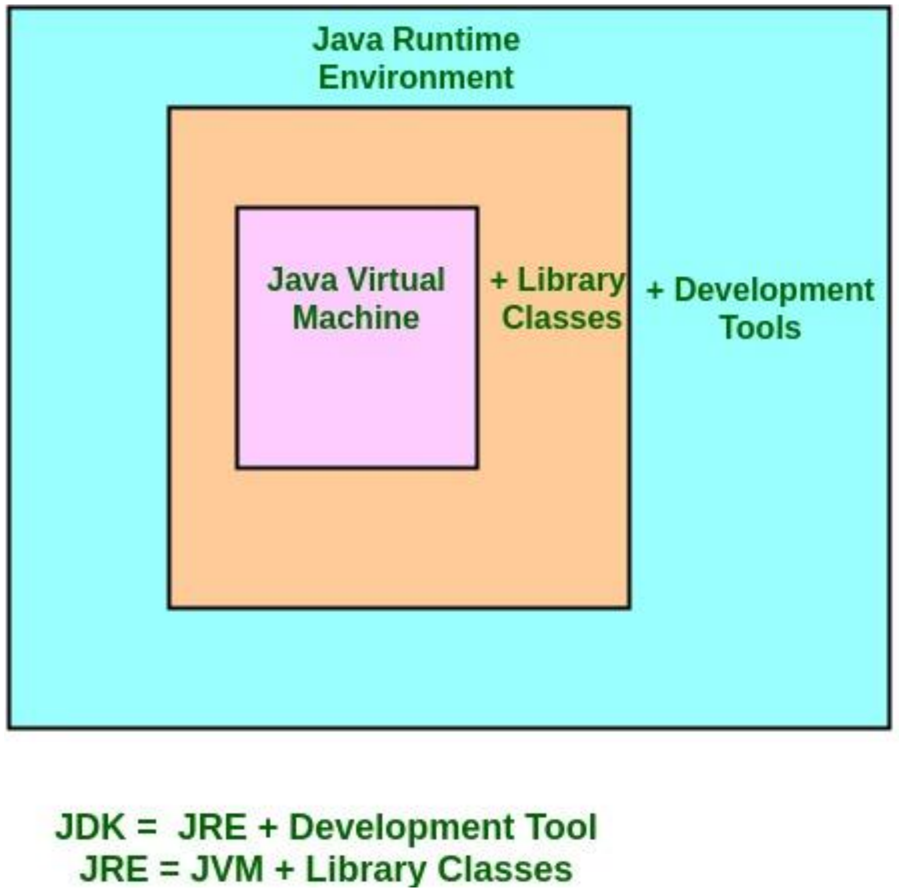
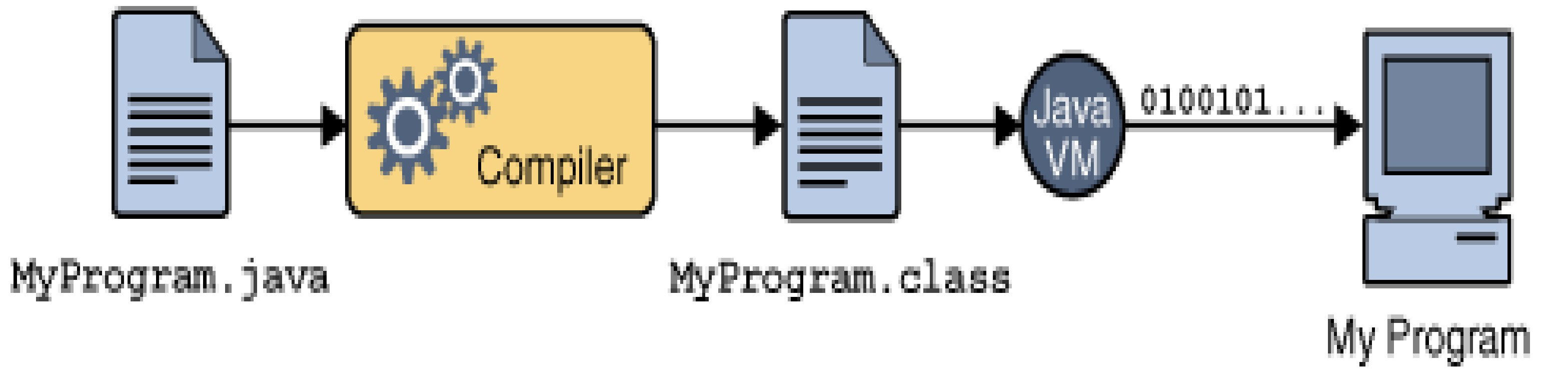


Fig 1: Comparison between JDK, JRE and JVM

Program execution process

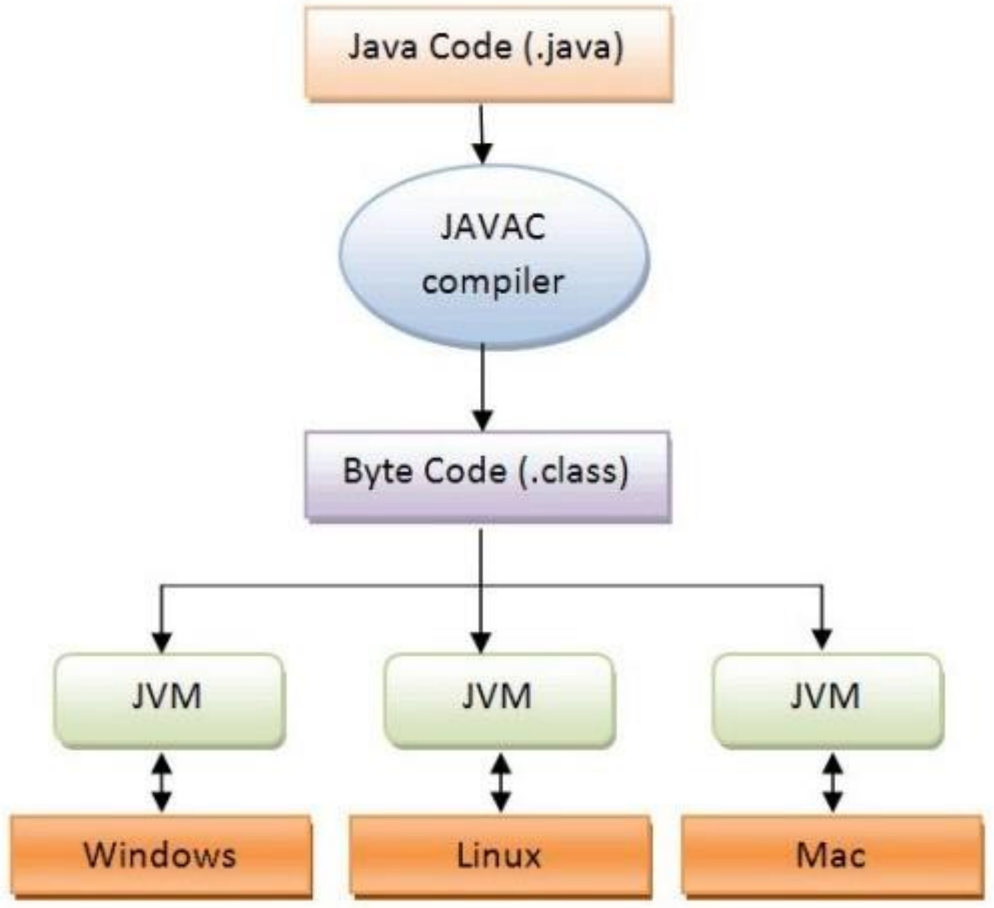
- All source code is first written in plain text files ending with the .java extension.
- Source files are then compiled into .class files by the javac compiler.
- A .class file contains bytecodes - the machine language of the Java Virtual Machine (JVM).
- The java launcher tool then runs your application with an instance of the Java Virtual Machine.



Java’s Platform Independency

- Because of **JVM**, Java is supported on many different operating systems.
- So the same **.class** files are capable of running on different operating systems.

  such as:
  - Microsoft Windows,
  - Linux or
  - Mac OS.



Features of Java

Platform Independence

- Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is interpreted by virtual Machine (JVM) on whichever platform it is being run.

Object Oriented

- Object oriented throughout - no coding outside of class definitions, including main(). An extensive class library available in the core language packages.

Compiler/Interpreter Combo

- Code is compiled to bytecodes that are interpreted by a Java virtual machines (JVM).
- This provides portability to any machine for which a virtual machine has been written.

Robust

- Java’s Exception handling mechanism allows developers to write code that can effectively deal with errors and exceptional conditions.
- Java's automatic memory management system handles memory allocation and deallocation, reducing the risk of memory leaks and pointer-related errors.

C++ vs Java

C++	Java
It is Platform dependent.	It is Platform Independent.
It supports goto statement, Operator Overloading, Pointer, destructor and multiple inheritance.	It does not support goto statement, Operator overloading, pointer, destructor and multiple inheritance (but it is achieved by interfaces).
It uses a compiler.	It uses compiler and interpreter.
It supports call by value and call by reference.	It supports only call by value.
It allows procedural programming as well as object oriented programming.	It allows only the object-oriented programming.
It has a scope resolution operator (::), structure, union.	It does not have a scope resolution operator (::), structure, union.
It has new and delete keyword for object management.	It has automatic garbage collection for object management.

Java Program Structure

- Every line of code that runs in Java must be inside the class.
- { … } left brace and right brace define the body of the class.

```
public class FirstProgram {

  public static void main(String[] args) {

      //This will print Hello world:
     System.out.println("Hello World");

  }

}
```

Main Method

- In Java programs, the point from where the program starts its execution or simply the entry point of Java programs is the main() method.

Public: It is an Access modifier, which makes the main() method globally available, so that JVM can invoke it from outside the class.

```
public class FirstProgram {

  public static void main(String[] args) {

     //This will print Hello world:
    System.out.println("Hello World");

  }

}
```

Static: It is a keyword makes The main() method static so that JVM can invoke it without instantiating the class.

Void: It is a keyword and used to specify that a method doesn’t return anything. it doesn’t make any sense to return from main() method as JVM can’t do anything with the return value of it.

String[] args: It stores Java command line arguments in a array of type java.lang.String class.

{ … } left brace and right brace define the body of the function.

Benefits of Programming in Java

- **Get started quickly**
  Although the Java programming language is a powerful object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.
- **Write less code**
  Comparisons of program metrics suggest that a program written in the Java programming language can be four times smaller than the same program in C++.
- **Develop programs faster**
  Development time may be as much as twice as fast compared to writing the same program in C++ because you write fewer lines of code and it is a simpler programming language than C++.
- **Avoid platform dependencies.**
  You can keep your program portable by avoiding the use of libraries written in other languages and native methods.