



OOP using Java

Trainer: Mr. Rohan Paramane



Java Text Books

- Java, The complete reference, Herbert Schildt
- Core and Advanced Java Black Book
- Head First Java: A Brain-Friendly Guide , by Kathy Sierra



Java History

- **James Gosling, Mike Sheridan** and **Patrick Naughton** initiated the Java language project in June 1991.
- **Java** was originally **developed by James Gosling** at **Sun Microsystems** and **released in 1995**.
- The language was initially called **Oak** after an oak tree that stood outside Gosling's office.
- Later the project went by the name **Green** and was finally renamed **Java**, from Java coffee, a type of coffee from Indonesia.
- Gosling and his team did a brainstorm session and after the session, they came up with several names such as **JAVA, DNA, SILK, RUBY, etc.**
- Sun Microsystems released the first public implementation as Java 1.0 in 1996.
- The Java programming language is a general-purpose, concurrent, class-based, object-oriented language.
- The Java programming language is a high-level language.
- The Java programming language is related to C and C++ but is organized rather differently, with a number of aspects of C and C++ omitted and a few ideas from other languages included.
- **It is intended to be a production language, not a research language.**
- The Java programming language is statically typed.
- It promised **Write Once, Run Anywhere (WORA)** functionality.



Version History

| Version | Date |
|------------|------------------------------------|
| JDK Beta | 1995 |
| JDK1.0 | January 23, 1996 ^[39] |
| JDK 1.1 | February 19, 1997 |
| J2SE 1.2 | December 8, 1998 |
| J2SE 1.3 | May 8, 2000 |
| J2SE 1.4 | February 6, 2002 |
| J2SE 5.0 | September 30, 2004 |
| Java SE 6 | December 11, 2006 |
| Java SE 7 | July 28, 2011 |
| Java SE 8 | March 18, 2014 |
| Java SE 9 | September 21, 2017 |
| Java SE 10 | March 20, 2018 |
| Java SE 11 | September 25, 2018 ^[40] |
| Java SE 12 | March 19, 2019 |
| Java SE 13 | September 17, 2019 |
| Java SE 14 | March 17, 2020 |
| Java SE 15 | September 15, 2020 ^[41] |
| Java SE 16 | March 16, 2021 |

- The first version was released on January 23, 1996.
- The acquisition of Sun Microsystems by Oracle Corporation was completed on January 27, 2010
- As of September 2020, Java 8 and 11 are supported as Long Term Support (LTS) versions
- In September 2017, Mark Reinhold, chief Architect of the Java Platform, proposed to change the release train to "one feature release every six months".
- OpenJDK (Open Java Development Kit) is a free and open source implementation of the (Java SE). It is the result of an effort Sun Microsystems began in 2006.
- Java Language and Virtual Machine Specifications: <https://docs.oracle.com/javase/specs/>



Java Platforms

1. Java SE

1. Java Platform Standard Edition.
2. It is also called as Core Java.
3. For general purpose use on Desktop PC's, servers and similar devices.

2. Java EE

1. Java Platform Enterprise Edition.
2. It is also called as advanced Java / enterprise Java / web Java.
3. Java SE plus various API's which are useful for client-server applications.

3. Java ME

1. Java Platform Micro Edition.
2. Specifies several different sets of libraries for devices with limited storage, display, and power capacities.
3. It is often used to develop applications for mobile devices, PDAs, TV set-top boxes and printers.

4. Java Card

1. A technology that allows small Java-based applications (applets) to be run securely on smart cards and similar small-memory devices.



SDK, JDK, JRE, JVM

- **SDK** = Development Tools + Documentation + Libraries + Runtime Environment.
- **JDK** = Java Development Tools + Java Docs + **rt.jar** + **JVM**.
 - JDK : Java Development Kit.
 - It is a software, that need to be install on developers machine.
 - We can download it from oracle official website.
- **JRE**[**rt.jar** + **JVM**].
 - JRE : Java Runtime Environment.
 - rt.jar and JVM are integrated part of JRE.
 - JRE is a software which comes with JDK. We can also download it separately.
 - To deploy application, we should install it on client's machine.
 - rt.jar file contains core Java API in **compiled form**.
- **JVM** : An engine, which manages execution of Java application. (also called as Execution Engine)



- **Java Development Kit (JDK)**

1. Java Developmet Tools

- a. Javac
- b. Java
- c. Javap
- d. jar
- e. Jarsigner

2. Java RunTime Environment (JRE)

- I. Java API library

- a. Bundled package classes
- b. Jre/lib/rt.jar

- II. Java Virtual Machine (JVM)

- a. Load and Execute java application



Entry point method

- Syntax:
 1. `public static void main(String[] args)`
 2. `public static void main(String... args)`
- Java compiler do not check/invoke main method. JVM invoke main method.
- When we start execution of Java application then JVM starts execution of two threads:
 1. Main thread : responsible for invoking main method.
 2. Garbage Collector : responsible for deallocating memory of unused object.
- We can overload main method in Java.
- We can define main method per class. But only one main method can be considered as entry point method.



Simple Hello Application using command line

```
//File Name : Program.java
class Program{
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

```
Compilation=> javac Program.java //Output : Program.class
Execution=>    java Program
```

```
System      : Final class declared in java.lang package
out         : public static final field of System class. Type of out is PrintStream
println     : Non static method of java.io.PrintStream class
```

To view class File :

javap -c Program.class

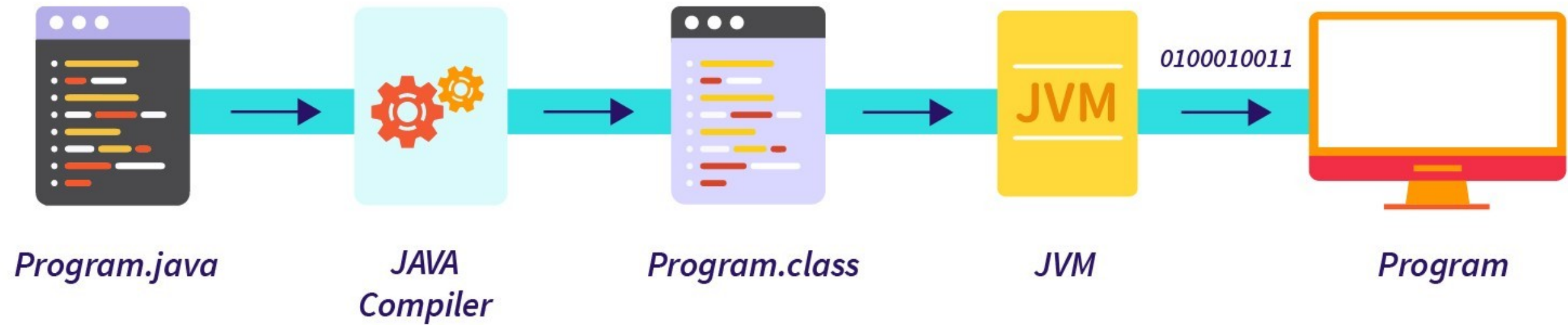


Using STS

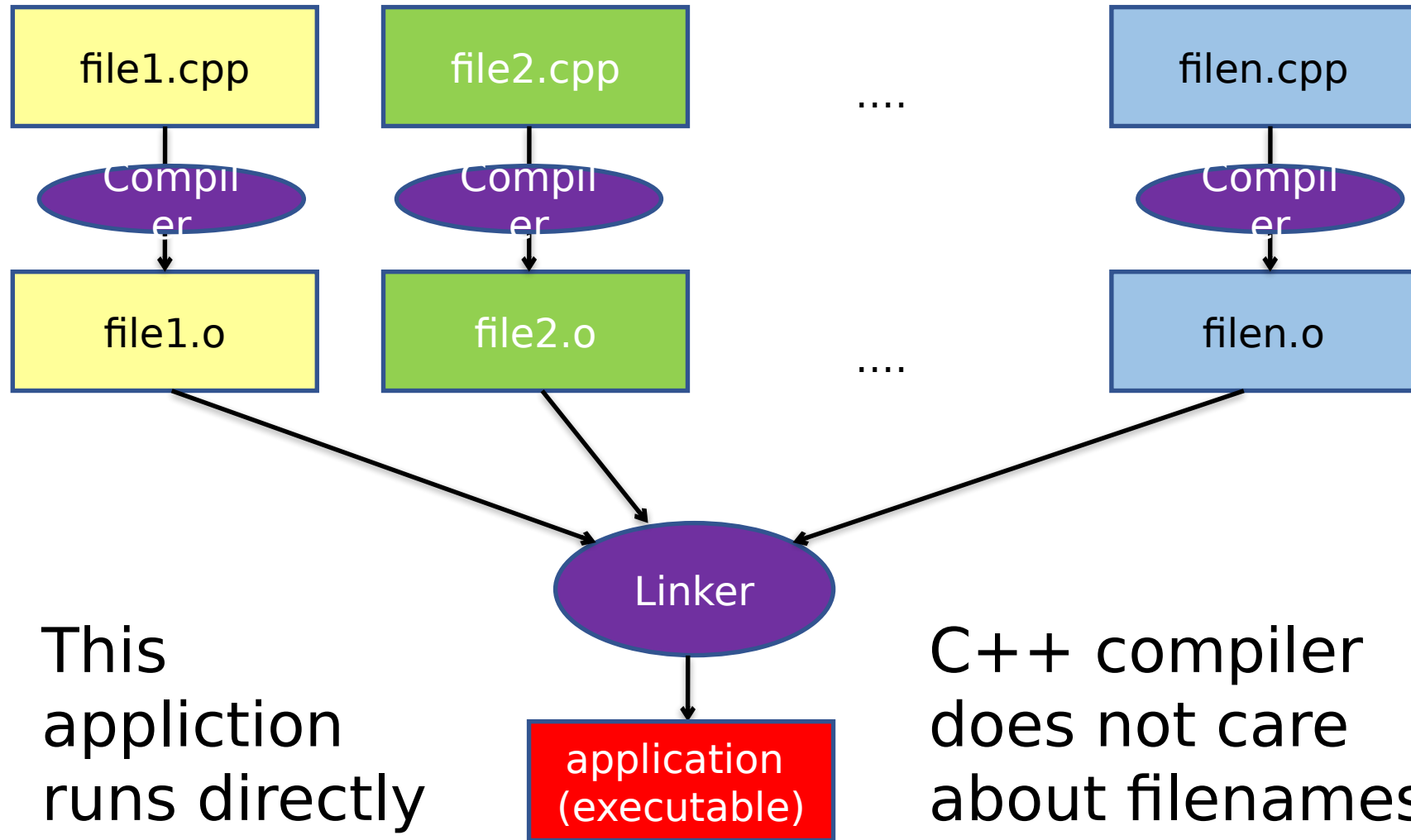
- Open the STS and chose the workspace where you want to store the projects
- Change the perspective to java
- Create a new java project by giving the proper name
- Create a new file named 'Program.java' inside src folder.
- The class will be automatically created.
- Write the entry point function main inside your class and write the hello world program.
- Execute the code.



Flow of Execution



C++ compiler & Linker usage





Thank you!

Rohan Paramane

rohan.paramane@sunbeaminfo.com

