OOP using Java

Trainer: Mr. Rohan Paramane

Agenda

- History
- Platforms
- JDK,JRE,JVM
- OOSD
- Main
- Hello World using cmd line & STS
- Java BuzzWords
- Compiling .class file in bin

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

Documentation

- **Oracle Tutorial:**
  - https://docs.oracle.com/javase/tutorial/

- **Java 8 API and Java 11 API Documentation Documentation:**
  - https://docs.oracle.com/javase/8/docs/api/
  - https://docs.oracle.com/en/java/javase/11/docs/api/allpackages-index.html

MySQL Connector:
  - https://dev.mysql.com/downloads/connector/j/

- **Offline Documentation Download:**
  - https://www.oracle.com/java/technologies/javase-jdk8-doc-downloads.html

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 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)

JDK,JRE,JVM

- **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

Object-oriented software development (OOSD)

- In the past, the problems faced by software development were relatively simple, from task analysis to programming, and then to the debugging of the program, if its not too big it can be done by one person or a group.
- With the rapid increase of software scale, software personnel faces the problem that is very complicated, and there are many factors that need to be considered.
- The errors generated and hidden errors may reach an astonishing degree, this is not something that can be solved in the programming stage.
- Need to standardize the entire software development process and clarify the software
- The tasks of each stage in the development process, while ensuring the correctness of the work of the previous stage, proceed to the next stage work.
- This is the problem that software engineering needs to study and solve. Object-oriented software development and engineering include the following parts:

1.Object oriented analysis (OOA)

- The first step of Object-oriented software development is Object-Oriented Analysis (OOA)
- In the system analysis stage of software engineering, system analysts must integrate with users to make precise Accurate analysis and clear description, summarize what the system should do (not how) from a macro perspective.
- Face right the analysis of the image should be based on object-oriented concepts and methods.
- In the analysis of the task, from the objective existence of things and the relationship between the related objects (including the attributes and behaviors of the objects) and the relationship between the objects are summarized.
- The Objects with the same attributes and behaviors are represented by a class.
- Establish a need to reflect the real work situation model. The model formed at this stage is relatively rough (rather than fine).

2.Object oriented design (OOD)

- The second step of Object-oriented software development is Object-Oriented Design (OOD).
- According to the demand model formed in the object-oriented analysis stage, each part is specifically designed.
- The design of the line class may contain multiple levels (using inheritance).
- Then these classes put forward the ideas and methods of program design, including the design of algorithms.
- In the design stage no specific plan is involved, but a more general description tool (such as pseudo code or flowchart)is used to describe.

3.Object-oriented programming (OOP)

- The third step of Object-oriented software development is Object-oriented Programming (OOP).
- According to the results of object-oriented design, to write it into a program in a computer language, it is obvious that object-oriented Computer language (e.g. C++) needs to be used.
- Otherwise the requirements of object-oriented design cannot be achieved.

OOPS(Object Oriented Programming Language)

- OOPS is not a syntax.
- It is a process / programming methodology which is used to solve real world problems.
- It is a programming methodology to organize complex program in to simple program in terms of classes and object such methodology is called oops.
- It is a programming methodology to organized complex program into simple program by using concept of abstraction , encapsulation , polymorphism and inheritance.
- Languages which support abstraction , encapsulation polymorphism and inheritance are called oop language.

Major pillars of oops

- **Abstraction**
  - getting only essential things and hiding unnecessary details is called as abstraction.
  - Abstraction always describe outer behavior of object.
  - In console application when we give call to function in to the main function , it represents the abstraction
- **Encapsulation**
  - binding of data and code together is called as encapsulation.
  - Implementation of abstraction is called encapsulation.
  - Encapsulation always describe inner behavior of object
  - Function call is abstraction
  - Function definition is encapsulation.
  - Information hiding
    - Data : unprocessed raw material is called as data.
    - Process data is called as information.
    - Hiding information from user is called information hiding.
    - In java we used access Modifiers to provide information hiding.
- **Modularity**
  - Dividing programs into small modules for the purpose of simplicity is called modularity.
- **Hierarchy (Inheritance [is-a] , Composition [has-a] , Aggregation[has-a], Dependancy)**
  - Hierarchy is ranking or ordering of abstractions.
  - Main purpose of hierarchy is to achieve re-usability.

Minor pillars of oops

- **Polymorphism (Typing)**
  - One interface having multiple forms is called as polymorphism.
  - Polymorphism have two types
    1. **Compile time polymorphism**

       when the call to the function resolved at compile time it is called as compile time polymorphism. And it is achieved by using function overloading and operator overloading
    2. **Runtime polymorphism.**

       when the call to the function resolved at run time it is called as run time polymorphism. And it is achieved by using function overriding.
  - Compile time / Static polymorphism / Static binding / Early binding / Weak typing / False Polymorphism
  - Run time / Dynamic polymorphism / Dynamic binding / Late binding / Strong typing / True polymorphism
- **Concurrency**
  - The concurrency problem arises when multiple threads simultaneously access same object.
  - You need to take care of object synchronization when concurrency is introduced in the system.
- **Persistence**
  - It is property by which object maintains its state across time and space.
  - It talks about concept of serialization and also about transferring object across network.

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
```

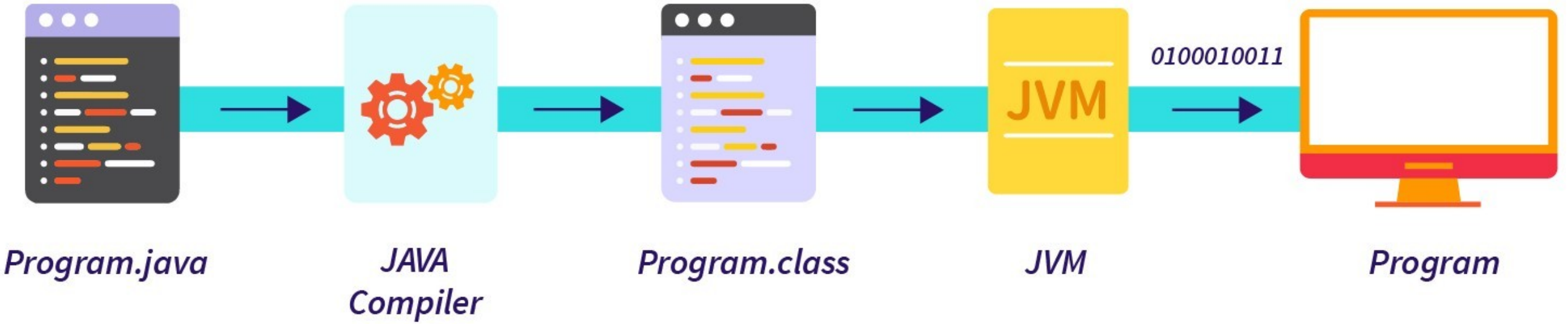
To view class File :

javap -c Program.class

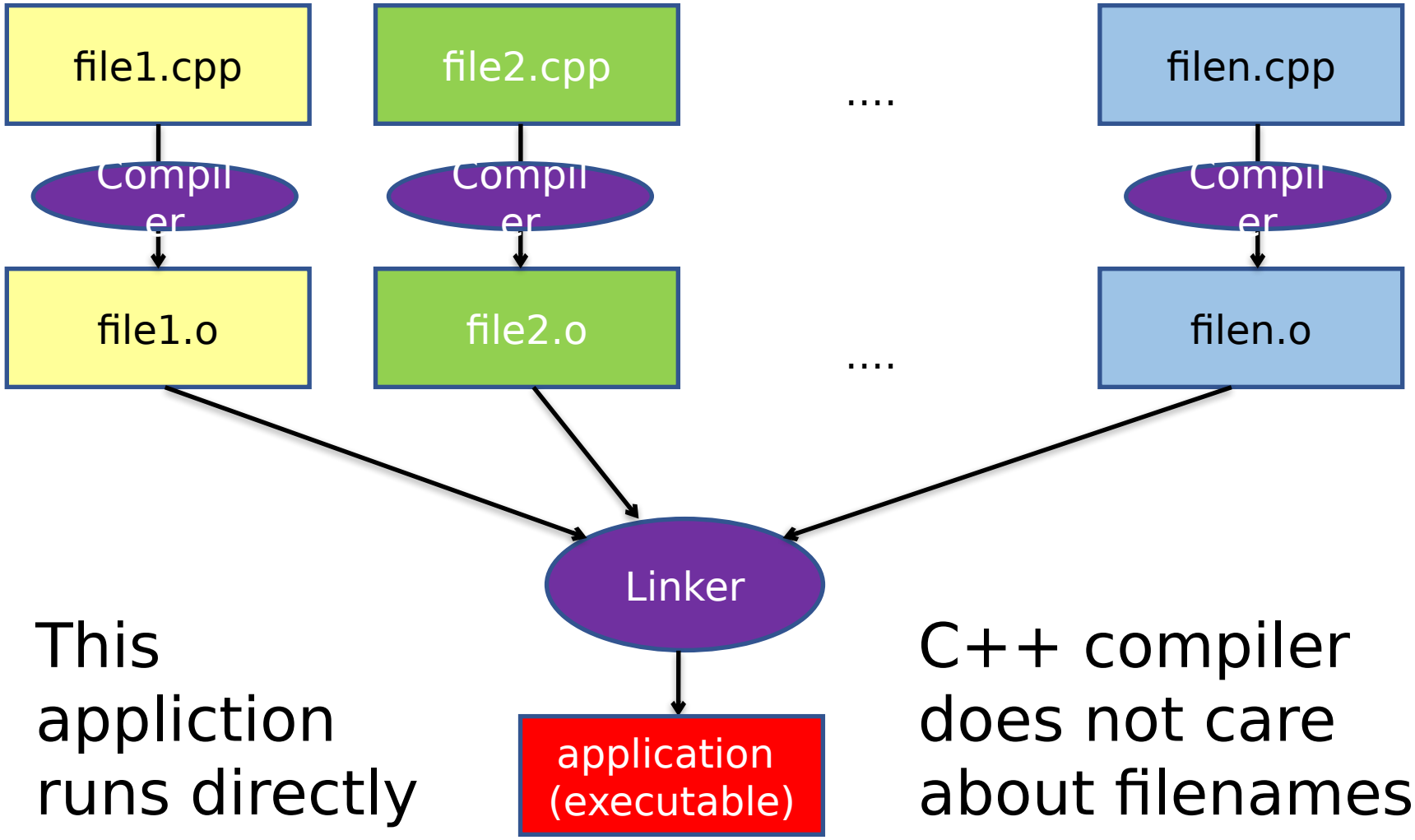
Using STS

- 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



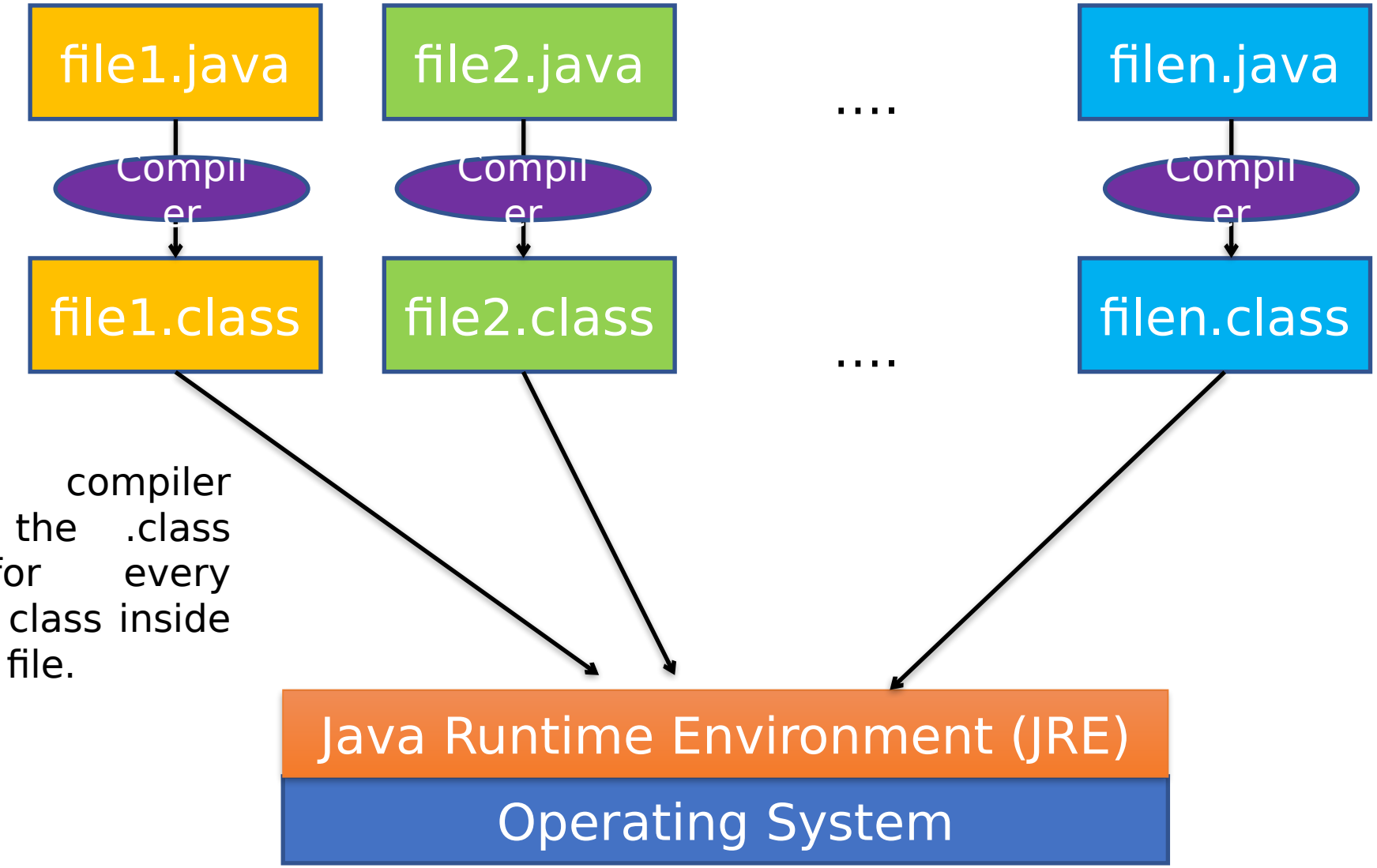
C++ compiler & Linker usage



This appliction runs directly on top of OS.

C++ compiler does not care about filenames.

Java compiler usage



Java compiler creates the .class files for every declared class inside the .java file.

Java Buzzwords

1. Simple
2. Object Oriented
3. Architecture Neutral
4. Portable
5. Robust
6. Multithreaded
7. Dynamic
8. Secure
9. High Performance
10. Distributed

Buzzwords

1. Simple
   - Java is simple programming language.
   - Syntax of Java is simpler than syntax of C/C++ hence it is considered as simple
   - No need of header files and macros.
   - We cannot define anything global
   - Do not support structure, union, operator overloading
   - No diamond problem
   - Do not support pointer

2. Object Oriented
   - It follows all major and minor pillars of OOPS

3. Architecture Neutral (Platform Independent)
   - the Java Compiler generates *bytecodes* (an *architecture neutral)* intermediate format designed to transport code efficiently to multiple hardware and software platforms.

Buzzwords

4. Portable
   - As java is architecture neutral it makes the code to be carried to any hardware and able to be executed on same. This is what makes java portable. JVM plays important role for its portability.

5. Robust
   - Java programming language is designed for creating highly *reliable* software. It provides extensive compile-time checking, followed by a second level of run- time checking.
   - Java is robust because of following features:
     1. Architecture Neutral.
        - Java developer is free from developing H/W or OS specific coding.
     2. Object Oriented.
        - Reusability reduces developer's effort.
     3. Automatic memory management.
        - Developer need not to worry about memory leakage / program crashes.
     4. Exception handling.
        - Java compiler helps developer to provide try-catch block.

Buzzwords

6. Multithreaded
   - When we start execution of Java application then JVM starts 2 threads and hence it is called as multithreaded.
   - The two threads that are started are :
     1. Main Thread
        - It is user thread responsible for invoking main method
     2. Garbage Collector
        - It is daemon/background thread responsible for releasing/deallocating memory of unused objects.

7. Dynamic
   - Java is capable of linking in new class libraries, methods, and objects.
   - Java programs carry with them run-time type information that is used to verify and resolve accesses to objects at runtime.
   - This makes it possible to dynamically link code in a safe manner.

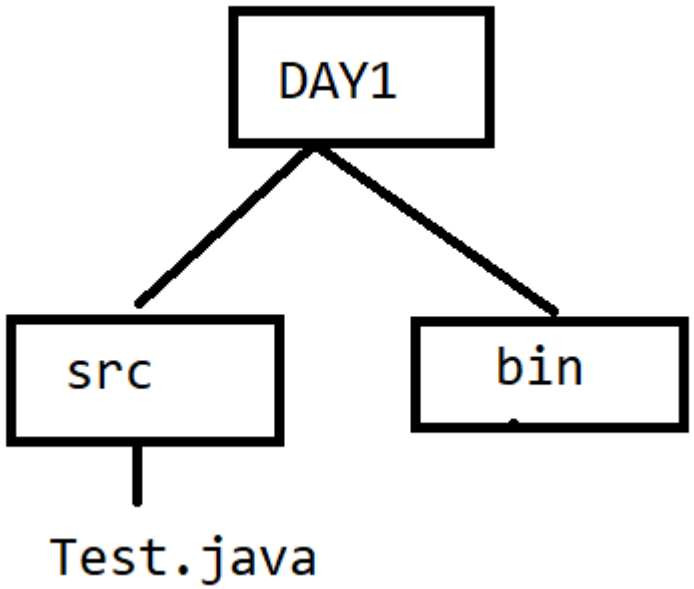
Buzzwords

8. Secure
   - Java is intended to be used in networked/distributed environments.
   - Java was designed to make certain kinds of attacks impossible, among them:
     1. Overrunning the runtime stack—a common attack of worms and viruses
     2. Corrupting memory outside its own process space
     3. Reading or writing files without permission

9. High Performance
   - Java performance is high because of the use of bytecode.
   - The bytecode was used so that it can be easily translated into native machine code.
   - The JIT Compiler helps even in speeding up the process of conversion to native code

10. Distributed
    - Java has an extensive library of routines for coping with protocols like HTTP , TCP/IP and FTP.
    - Java applications can open and access objects across the Net via URL with the same ease as when accessing a local file system.

How to compile the code from src and store .class inside bin



Windows Steps :

D:\DAY1\src>javac –d ..\bin Test.java

D:\DAY1\src>cd ..\bin

D:\DAY1\bin>java Test

Linux Steps :

/DAY1/src>javac –d ../bin Test.java

/DAY1/src>cd ../bin

/DAY1/bin>java Test

Thank you!

Rohan Paramane

rohan.paramane@sunbeaminfo.com