Object Oriented Programming with Java 8
PG-DAC

Rohan Paramane

Agenda

- Installation
- History
- BuzzWords
- Java Platforms
- SDK,JDK
- OOSD
- OOP

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

- **JDK download:**
  - https://adoptium.net/temurin/releases/?version=11
- **Spring Tool Suite 3 download:**
  - https://github.com/spring-projects/toolsuite-distribution/wiki/Spring-Tool-Suite-3
- **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://downloads.mysql.com/archives/c-j/
- **Core Java Tutorials:**
  1. https://www.baeldung.com/java-tutorial
  2. https://www.journaldev.com/7153/core-java-tutorial

Java Installation

1. JDK 11 download

   - https://adoptium.net/temurin/releases/?version=11

   Choose your platform , download the installer and install the JDK

2. Open command prompt or terminal
Type
java -version
It should show you : java version "11.0.X"
Note : Basic version should be 11, update version may differ.

3.Java IDE download
https://github.com/spring-projects/toolsuite-distribution/wiki/Spring-Tool-Suite-3
Choose 3.9.18 version and download. Extract it. No installation is required.

4. Java API Dcoumentation link

   - https://docs.oracle.com/en/java/javase/11/docs/api/allpackages-index.html

5. Offline version
https://www.oracle.com/java/technologies/javase-jdk11-doc-downloads.html

JDK

JDK 1.8 Java SE 8

Java dev tools
eg --
javac,java,javap,jar,jar signer,appletviewer....

JRE --Java run time env

Java API libs
JAR --java archival format.
<java_home>/jre/lib rt.jar

--bundled packged classes
eg : java.lang(def available),java.util,java.io,java.net...
eg: import java.io.*;
import java.io.FileReader;

Java Virtual Machine(JVM)
Load & execute
1. class loader --loads reuiqred classes in JVM's memory(method area)
2.Interpreter(translator --)
3.JIT --Just in time compiler --native code cache
4. Hot spot profiler

JDK 11 (Java SE 11)

Java development tools

Java Run time Environment

Java API library
Divided in modules & packages

Java Virtual Machine : To load & execute java app.

JAR : <java_home>/lib/jrt-fs.jar

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.

Java History

- 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/

WHY Java ? – (Buzz Words)

1.Simple

- simple and easy to learn, understand, and code
- complicated features like pointers, operator overloading, structures, unions, etc. have been removed.
- most useful features is the garbage collector it makes java more simple.

2. Secure

- because it does not have pointers concept
- provides a feature "applet" which can be embedded into a web application.
- applet in java does not allow access to other parts of the computer, which keeps away from harmful programs like viruses and unauthorized access.

3. Architecture-neutral (or) Platform Independent

- invented to archive "write once; run anywhere, any time
- JVM to provide this feature
- allows the java program created using one operating system can be executed on any other operating system.

4. Portable

- one of the core features of java which enables the java programs to run on any computer or operating system

Buzz Words

5. Object-oriented

- Java is said to be a pure object-oriented programming language.
- In java, everything is an object
- primitive data types java also implemented as objects using wrapper classes
- But Still it allows primitive data types to archive high-performance.

6. Robust

- Java is robust because of following reasons

1. Architecture Neutral
2. Object oiented
3. Automatic Memory management
    - strong memory management mechanism (garbage collector)
4. Exception handling
    - strong set of exception handling mechanism, etc

7. Multi-threaded

- Java supports multi-threading programming, which allows us to write programs that do multiple operations simultaneously.
- Main Thread, Garbage Collector Thread

Buzz Words

8. Interpreted
   - creation of cross-platform programs by compiling into an intermediate representation called Java bytecode
   - The byte code is interpreted to any machine code so that it runs on the native machine.

9. High Performance
   - Java provides high performance with the help of features like JVM, interpretation, and its simplicity.

10. Distributed
   - java to support the distributed environment of the Internet.
   - supports Remote Method Invocation (RMI), this feature enables a program to invoke methods across a network.

11. Dyamic
   - Java is a dynamic language as It supports the dynamic loading of classes.
   - It means classes are loaded on demand.

Java Platforms

1. **Java SE**
   - Java Platform Standard Edition.
   - It is also called as Core Java.
   - For general purpose use on Desktop PC's, servers and similar devices.
2. **Java EE**
   - Java Platform Enterprise Edition.
   - It is also called as advanced Java / enterprise java / web java.
   - Java SE plus various API's which are useful client-server applications.
3. **Java ME**
   - Java Platform Micro Edition.
   - Specifies several different sets of libraries for devices with limited storage, display, and power capacities.
   - It is often used to develop applications for mobile devices, PDAs, TV set-top boxes and printers.
4. **Java Card**
   - 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.

- **JDK** = Java Development Tools + Java Docs + **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)

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 things The relationship between the related objects (including the attributes and behaviors of the objects) and the relationship between the objects are summarized, and the Objects with the same attributes and behaviors are represented by a class.
- Establish a need to reflect the real work situation Seek a 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, the design of the class may contain multiple levels (using inheritance).
- Then these classes are based Put forward the ideas and methods of program design, including the design of algorithms.
- In the design stage, no specific plan is involved. Computer language, but a more general description tool (such as pseudo code or flowchart) 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++), Otherwise, the requirements of object-oriented design cannot be achieved.
- 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.

Object Oriented

- Java is **object oriented** programming language.
  - Java Supports all the major and minor pillars of oops hence it is considered as object oriented programming language.

  - **Major pillars of oops.**
    1. Abstraction
    2. Encapsulation
    3. Modularity
    4. Hierarchy

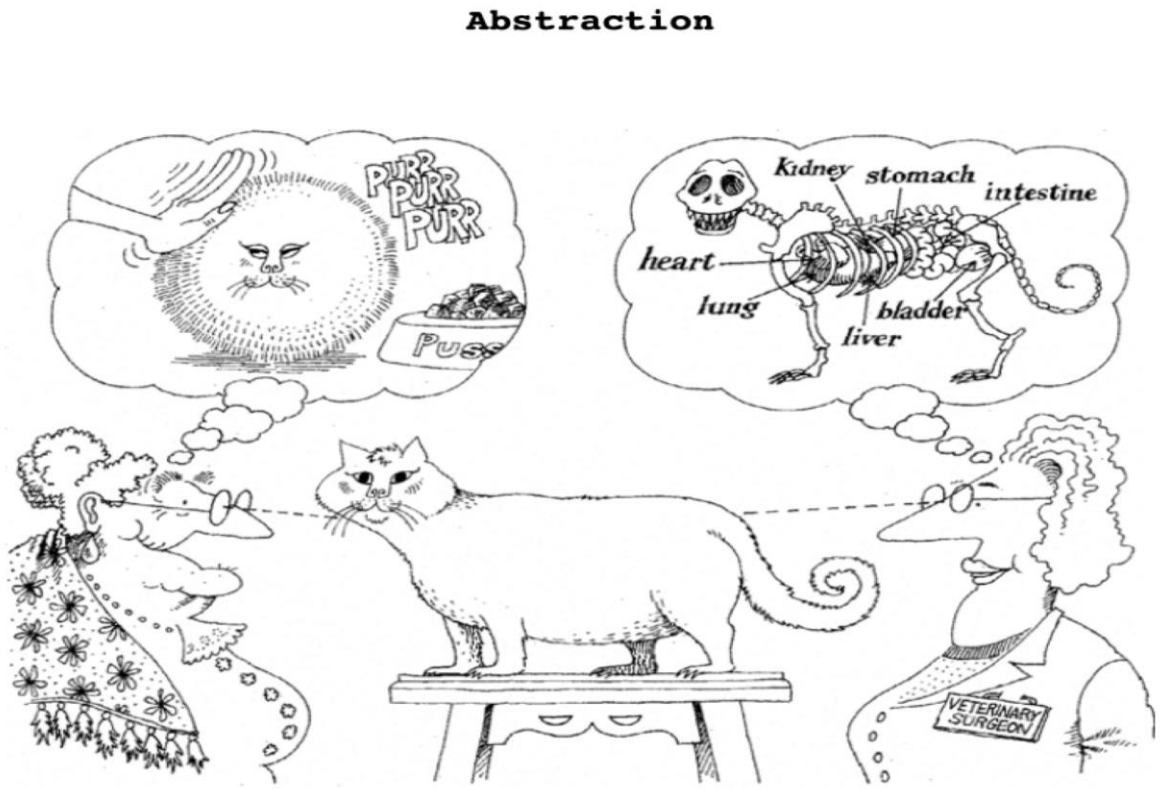
  - **Minor pillars of oops.**
    1. Typing / Polymorphism
    2. Concurrency
    3. Persistence.

Abstraction

- It is a major pillar of oops.
- **It is a process of getting essential things from object.**
- It describes outer behaviour of the object.
- Abstraction focuses on some essential characteristics of object relative to the perspective of viewer. In other words, abstraction changes from user to user.
- **Using abstraction, we can achieve simplicity.**
- Abstraction in Java

```
Complex c1 = new Complex( );
c1.acceptRecord( );
c1.printRecord( );
```

Abstraction



Abstraction focuses on the essential characteristics of some object, relative to the perspective of the viewer.

Encapsulation

- It is a major pillar of oops.

- Definition:
  1. **Binding of data and code together is called encapsulation.**
  2. To achieve abstraction, we should provide some implementation. It is called encapsulation.

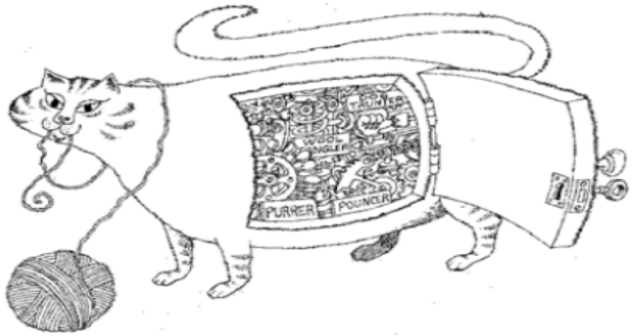
- Encapsulation represents, internal behaviour of the object.

- **Using encapsulation we can achieve data hiding.**

- Abstraction and encapsulation are complementary concepts: **Abstraction focuses on the observable behaviour** of an object, whereas **encapsulation focuses on the implementation** that gives rise to this behaviour.

Encapsulation

Encapsulation



Encapsulation hides the details of the implementation of an object.

Abstraction and encapsulation are complementary concepts: Abstraction focuses on the observable behavior of an object, whereas encapsulation focuses on the implementation that gives rise to this behavior.

Modularity

- It is a major pillar of oops.
- It is the process of developing complex system using small parts.
- **Using modularity, we can reduce module dependency.**
- We can implement modularity by creating library files.
  - .lib/.a, .dll / .so files
  - .jar/.war/.ear in java

Hierarchy

- It is a major pillar of oops.
- **Level / order / ranking of abstraction is called hierarchy.**
- Main purpose of hierarchy is **to achieve reusability.**
- Advantages of code reusability
    1. We can reduce development time.
    2. We can reduce development cost.
    3. We can reduce developers effort.

- Types of hierarchy:
    1. **Has-a** / Part-of => Association
    2. **Is-a** / Kind-of => Inheritance / Generalization
    3. **Use-a** => Dependency
    4. **Creates-a** => Instantiation

Typing

- It is a minor pillar of oops.
- Typing is also called as polymorphism.
- Polymorphism is a Greek word. Polymorphism = Poly(many) + morphism(forms).
- **An ability of object to take multiple forms is called polymorphism.**
- **Using polymorphism, we can reduce maintenance of the system.**
- Types of polymorphism:
  - **Compile time polymorphism**
    - It is also calling static polymorphism / **Early binding** / Weak Typing / False polymorphism.
    - We can achieve it using:
      1. **Method Overloading**
  - **Run time polymorphism**
    - It is also calling dynamic polymorphism / **Late binding** / Strong Typing / True polymorphism.
    - We can achieve it using:
      1. **Method Overriding.**

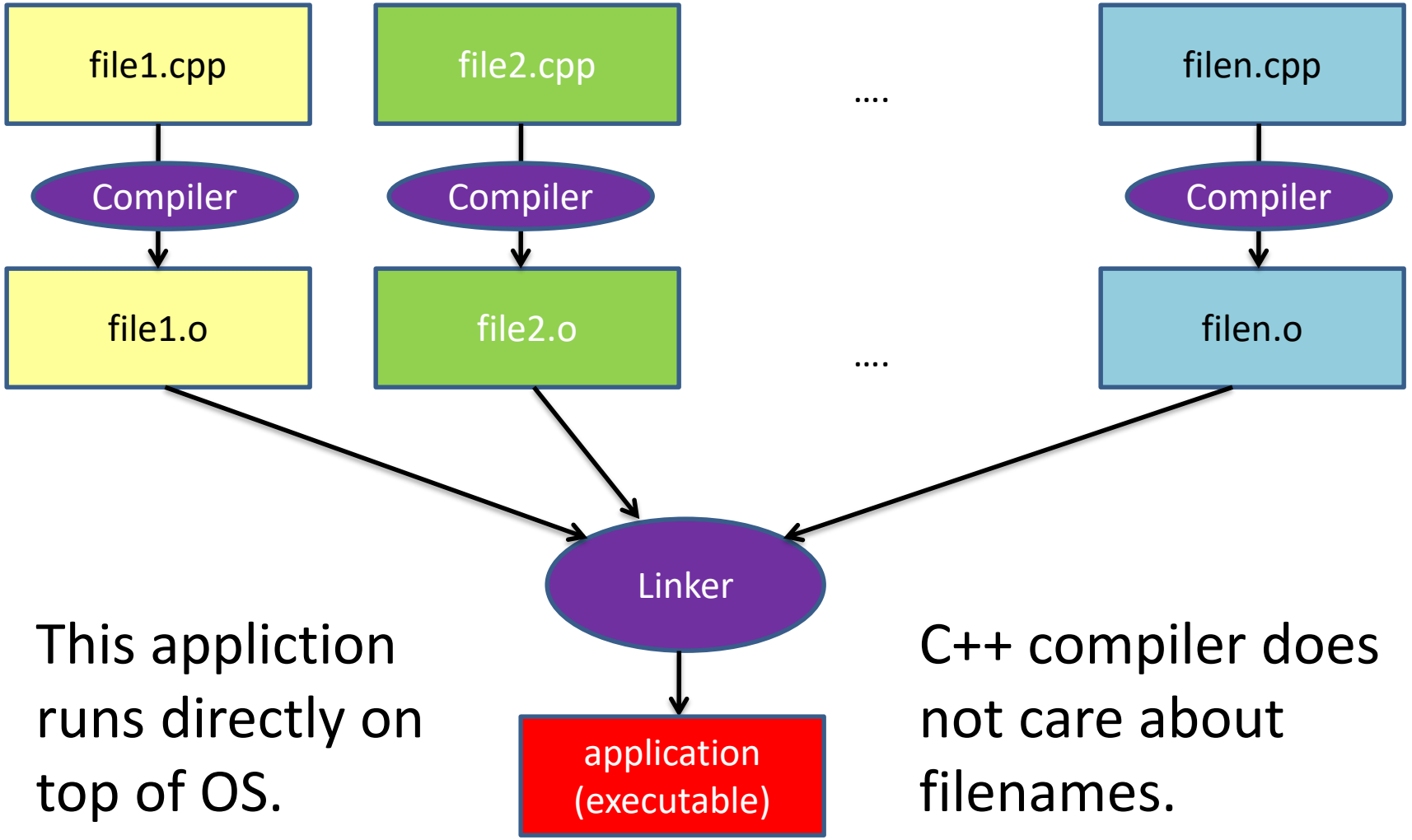
Concurrency

- It is a minor pillar of oops.
- In context of operating system, it is called as multitasking.
- It is the process of executing multiple task simultaneously.
- Main purpose of concurrency is to utilise CPU efficiently.
- In Java, we can achieve concurrency using thread.

Persistence

- It is a minor pillar of oops.
- It is process of maintaining state of object on secondary storage.
- In Java, we can achieve Persistence using file and database.

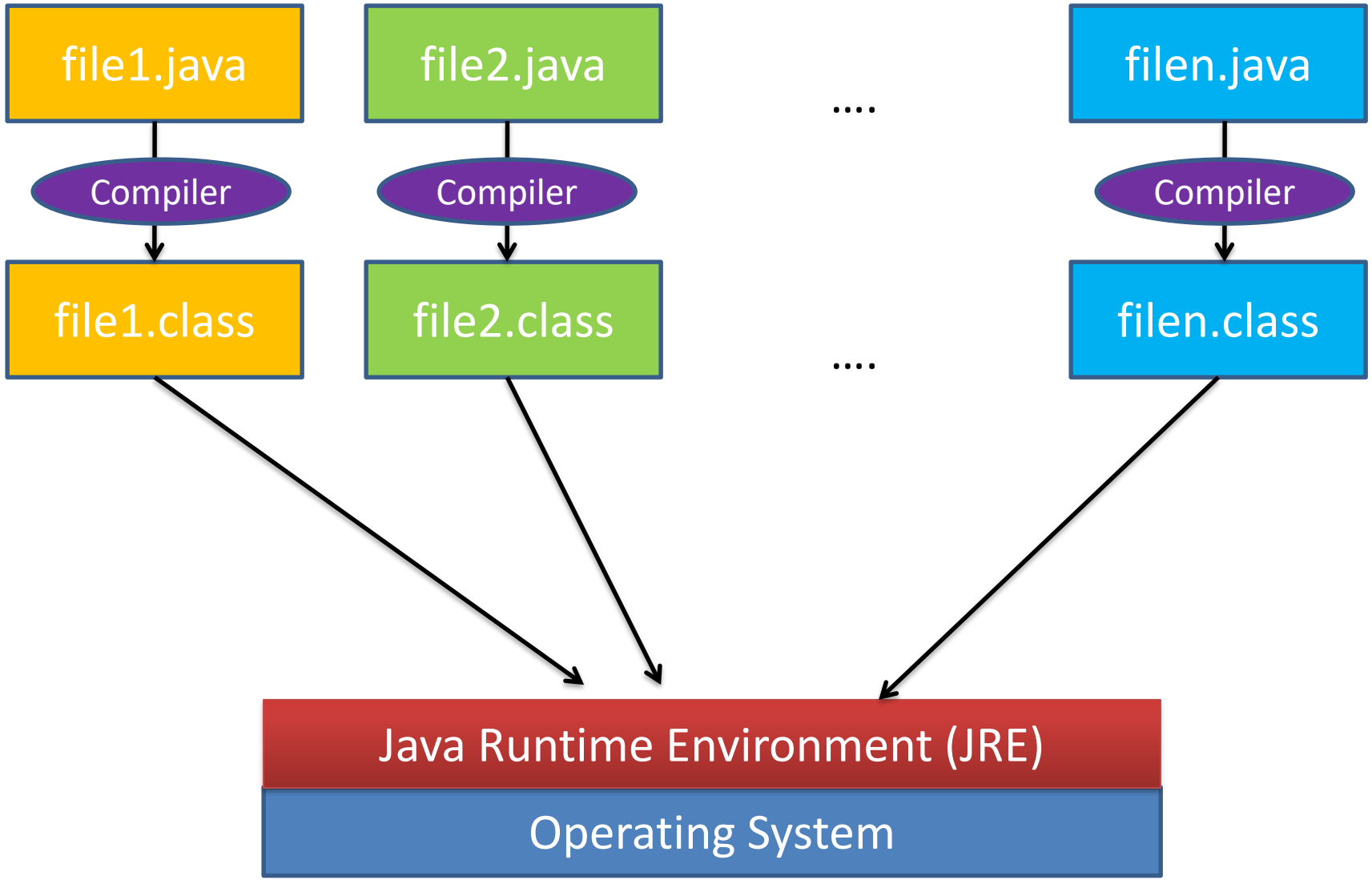
C++ compiler & Linker usage



This appliction runs directly on top of OS.

C++ compiler does not care about filenames.

Java compiler usage



Thank you.

Rohan.paramane@sunbeaminfo.com