Programming II
COMP212

Object Oriented Programming With Java

INTRODUCTION

About The Course

- Fundamental programming concepts
- Object-oriented programming (**OOP**)
- **Problem solving**

About The Course

- Fundamental programming concepts
  - Programming languages
  - Develop environment
  - Programming structures
  - **Java** ES

About The Course

- Object-oriented programming basic concepts
  - Objects and Classes
  - Encapsulation
  - Inheritance
  - Polymorphism
  - Abstract
  - Interfaces
  - Generics and containers
- Exception handling
- Functional Programming

About The Course

- Problem solving
  - *Encapsulate* related data and operations in classes
  - Apply *abstraction* and implementation to separate programming tasks
  - Design and organize programs and data using *classes*
  - Apply the *exception handling* mechanism to handle errors and anomalous
  - Apply *functional programming* techniques in problem solving

BASIC CONCEPTS

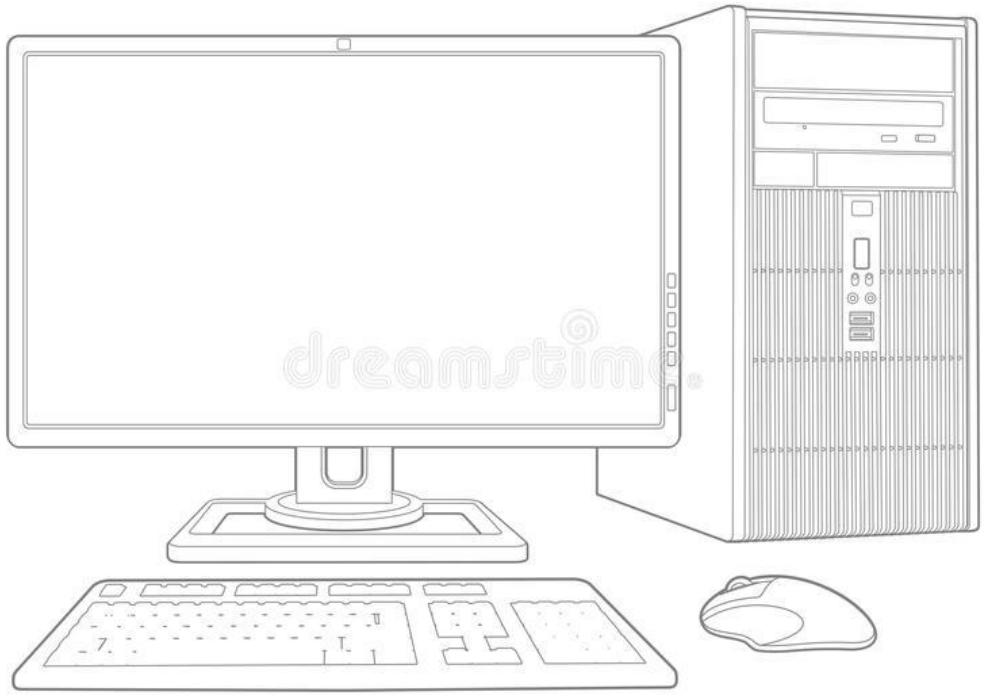
Basic Concepts

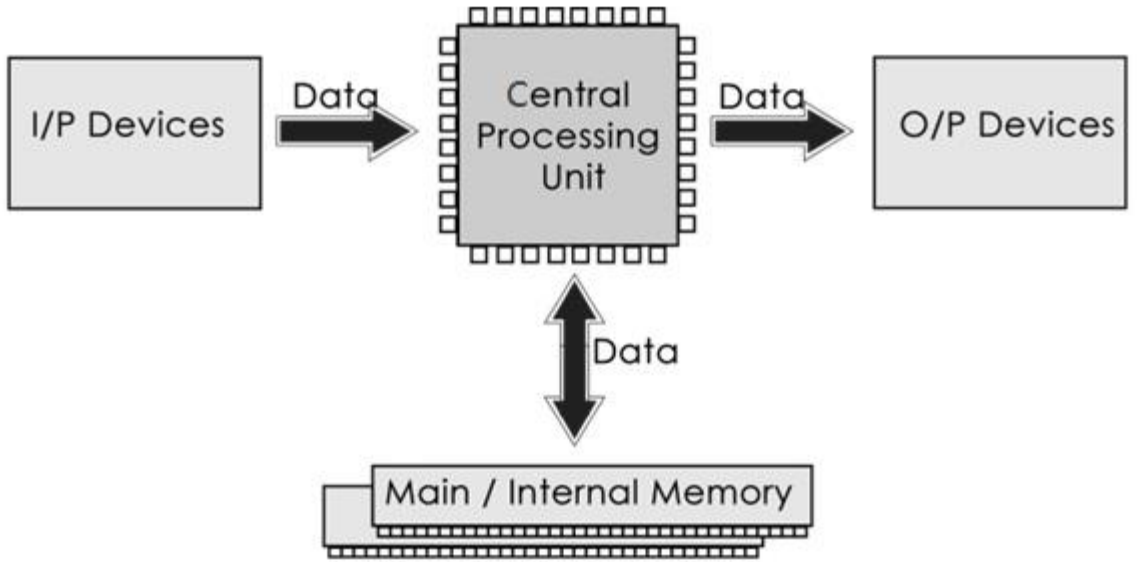
- Some general terms you need to know
  - Computer: Super Computer; Mainframe; Minicomputer; Personal Computer (PC); Industrial Computer; mobile devices
  - **C**entral **P**rocessing **U**nit (CPU): Intel; AMD; ARM; IBM ...
  - Storage: RAM; ROM; Register; Cache; Hard Drive, USB Flash Disc, CD-ROM ...
  - **O**perating **S**ystem (OS): DOS; Windows; Mac Os; Linux; Unix; FreeBSD; ios; Android
  - Platform: Intel+Windows; AMD+Linux; IBM+UNIX...
  - Command-line interface: terminal; console; shell

What is computer?

What is computer?

A computer is a programmable electronic device that accepts raw data as input and processes it with a set of instructions (a program) to produce the result as output.



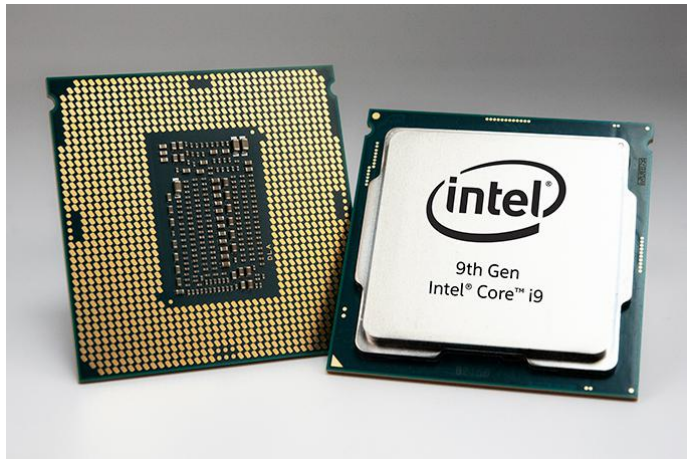

I/P Devices
Data
Central Processing Unit
Data
O/P Devices
Data
Main / Internal Memory

Basic Concepts

- categories of computer

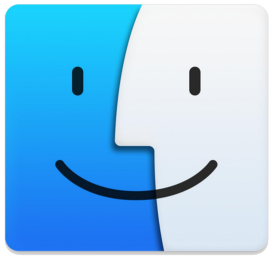
Basic Concepts

- **C**entral **P**rocessing **U**nit (CPU):



Basic Concepts

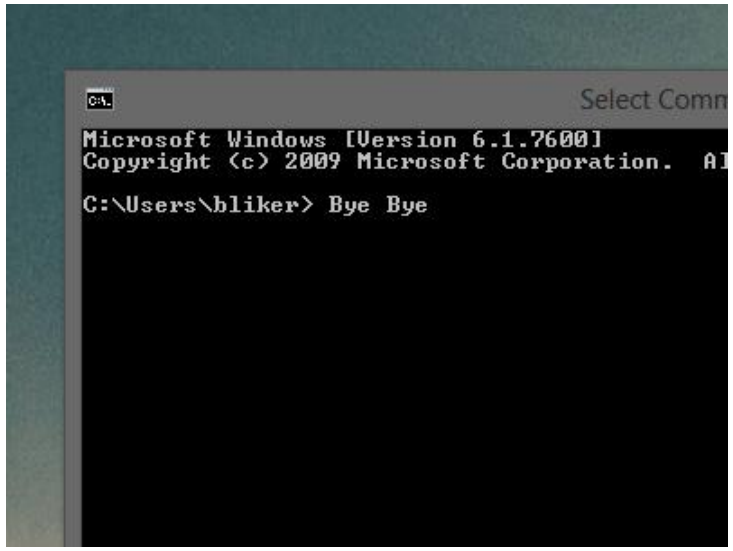
- **O**perating **S**ystem (OS):

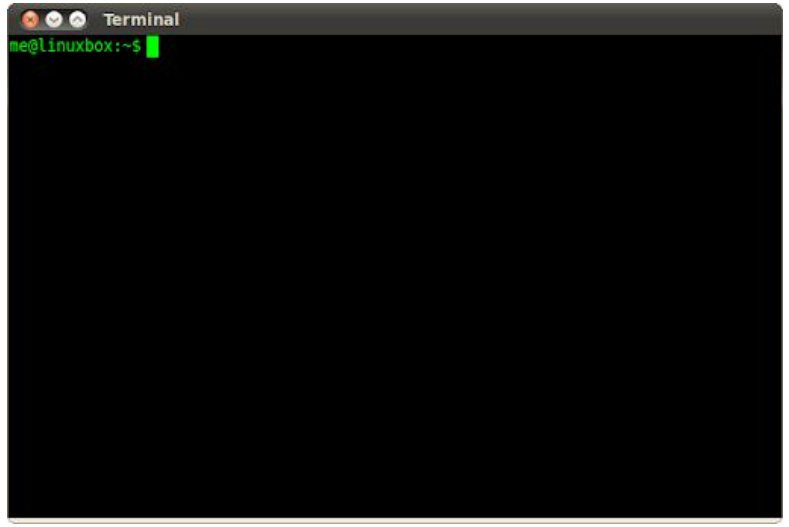


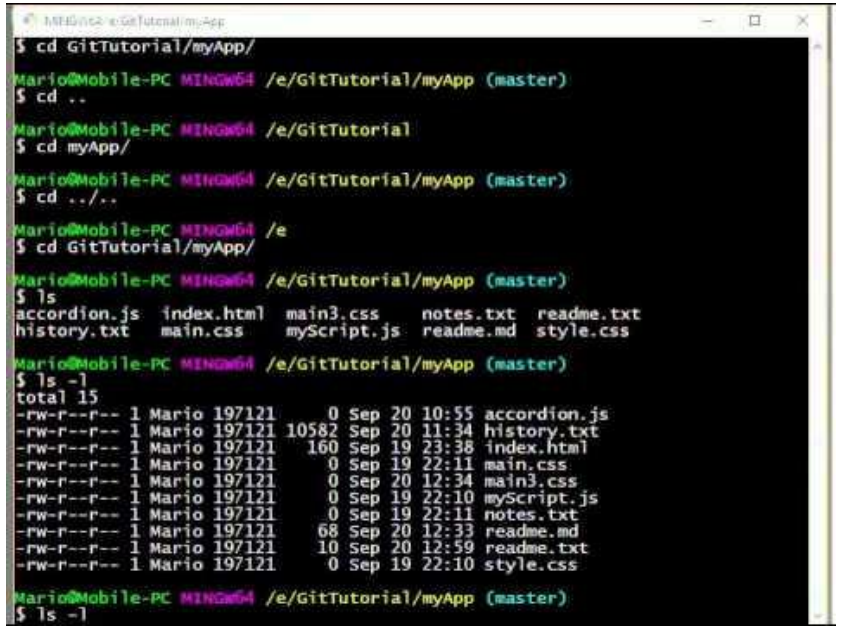
Mac OS

Basic Concepts

- Command-line interface: terminal; console; shell



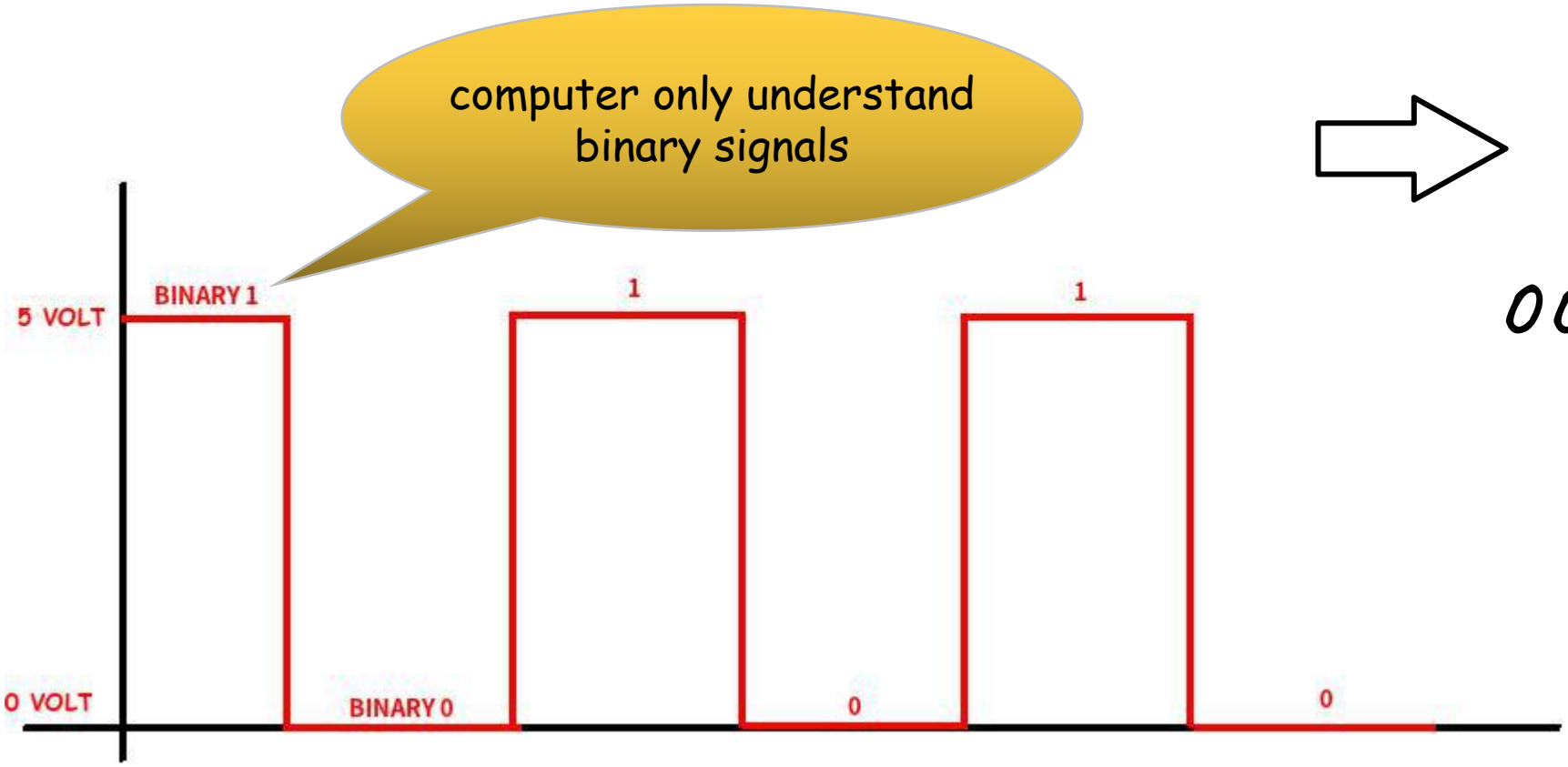




Basic Concepts

- Basic concepts related to programming
  - Programming languages categorizes:
    - machine language
    - assembly language
    - high-level programming language
  - Development tools
    - assembler
    - compiler
    - Editor, Integrated Development Environment (IDE)
    - Application Programming Interface (API) /Library

Machine Language



00100011

Assembly Language

- Program to add 1 with 2

Assembly Language

- Program to add 1 with 2

    1. Store 1 at memory location say A

Assembly Language

- Program to add 1 with 2

1. Store 1 at memory location say A
2. Store 2 at memory location say B

Assembly Language

- Program to add 1 with 2

1. Store 1 at memory location say A
2. Store 2 at memory location say B
3. Add contents of loation A & B

Assembly Language

- Program to add 1 with 2

1. Store 1 at memory location say A
2. Store 2 at memory location say B
3. Add contents of loation A & B
4. Store result

Assembly Language

- Program to add 1 with 2

1. Store 1 at memory location say A → `MOV RA, 1;`
2. Store 2 at memory location say B → `MOV RB, 2;`
3. Add contents of loation A & B → `ADD RA, RB;`
4. Show result → `MOV PortA, RA;`

High-level Programming Language

C

```
printf("1+2 = %d\n", 1+2);
```

JavaScript

```
console.log("1 + 2 = "+ (1+2));
```

Python

```
print("1+2 = " + str(1+2));
```

Java

```
System.out.println("1 + 2 = "+ (1+2));
```

JAVA BASICS

Basic Concepts

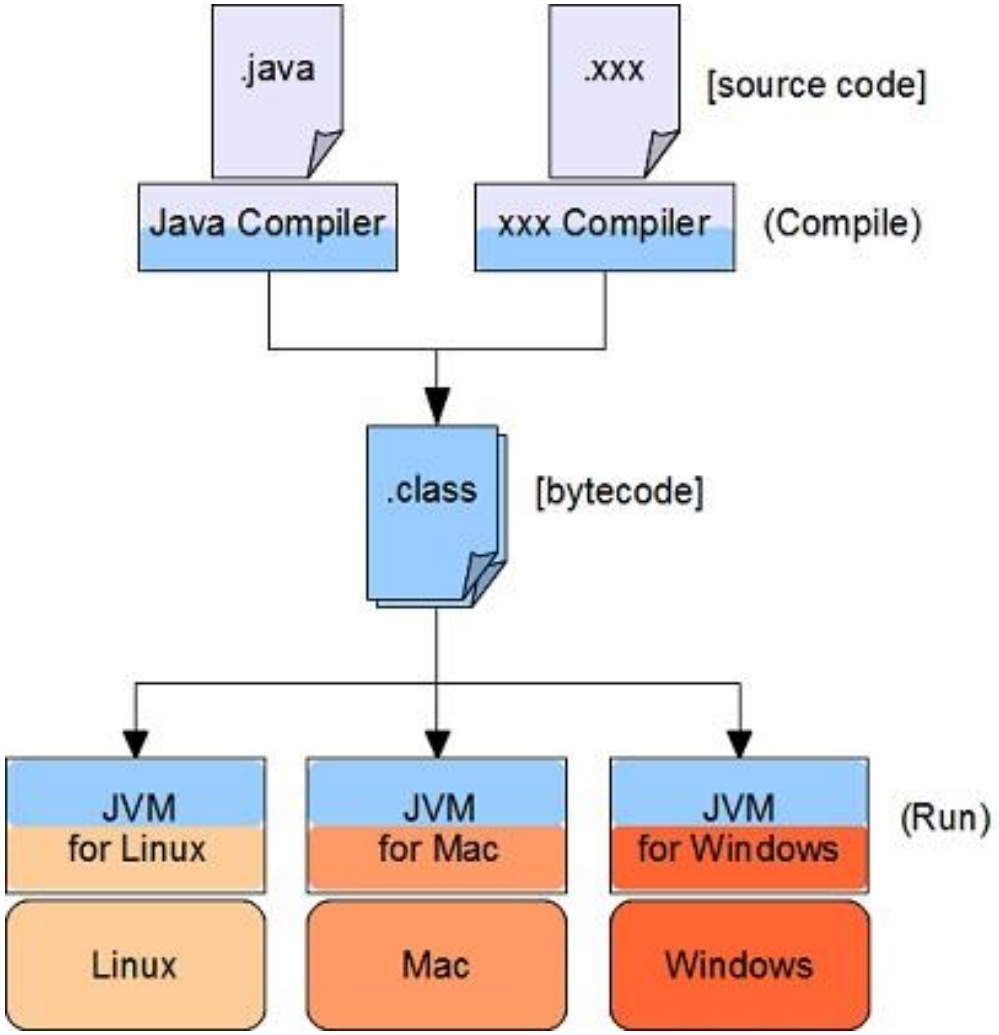
- Basic concepts about Java
  - Java editions: Java SE, Java EE, Java ME;
  - Java Virtual Machine (JVM);
  - Java Runtime Environment (JRE);
  - Java Development Toolkit (JDK);
  - Byte code;
  - source code;
  - IDEs: IntlliJ IDEA, Eclipse, NetBeans;

Java Editions

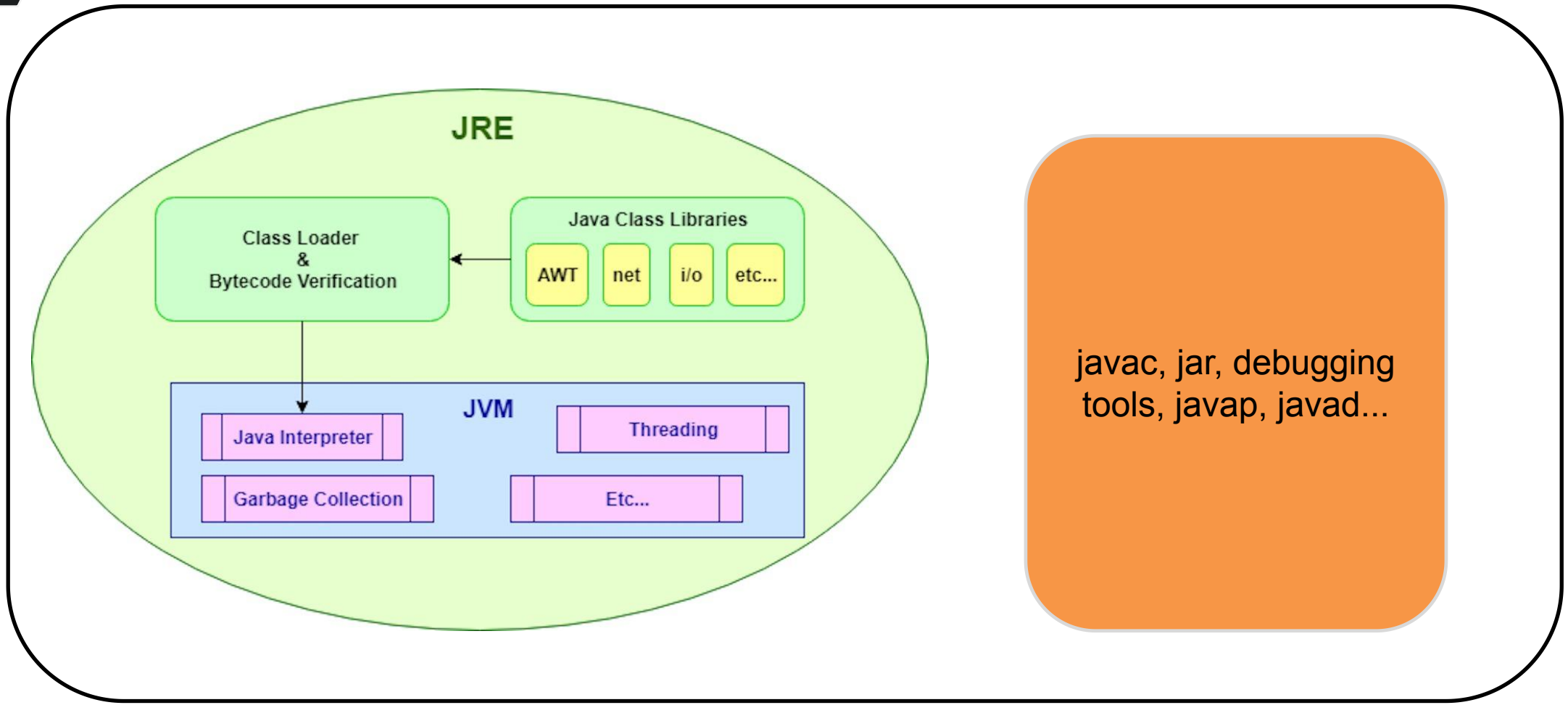
Java Standard Edition	Java Micro Edition	Java Enterprise Edition
Java standard edition is used to develop client-side standalone applications or applets	Java micro edition is used to develop applications for mobile devices such as cell phones, cars	Java enterprise edition is used to develop server-side applications such as java servlets and java server pages

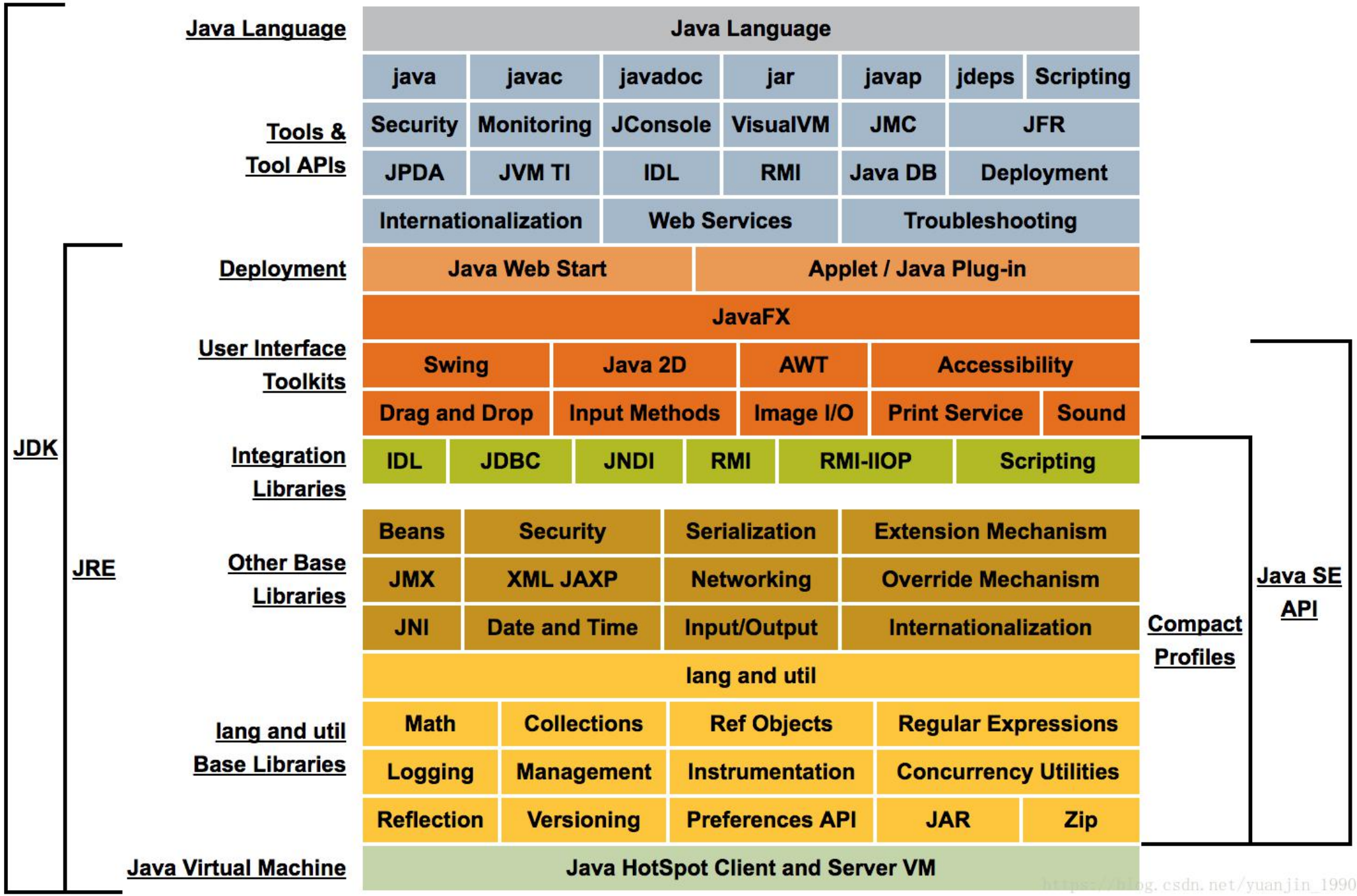
JVM

- A Java virtual machine (JVM) is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java bytecode.
- In computing, a virtual machine (VM) is an emulation of a computer system.



JRE, JDK


JRE
Class Loader
&
Bytecode Verification
Java Class Libraries
AWT
net
i/o
etc...
JVM
Java Interpreter
Threading
Garbage Collection
Etc...
javac, jar, debugging
tools, javap, javad...
JDK



Features of Java

- Simple
- Object-Oriented
- Platform independent
- Secured
- Multi Threading

Simple

- Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun, Java language is a simple programming language because:
  - Java syntax is based on C++ (so easier for programmers to learn it after C++).
  - Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
  - There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

Object-oriented

- Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior.
- Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.

Platform Independent

- Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language.
- A platform is the hardware or software environment in which a program runs.
- The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:
  - Runtime Environment
  - API(Application Programming Interface)

Secured

- Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:
  - No explicit pointer
  - Java Programs run inside a virtual machine sandbox
  - Classloader: Classloader in Java is a part of the Java Runtime Environment(JRE) which is used to load Java classes into the Java Virtual Machine dynamically. It adds security by separating the package for the classes of the local file system from those that are imported from network sources.
  - Bytecode Verifier: It checks the code fragments for illegal code that can violate access right to objects.
  - Security Manager: It determines what resources a class can access such as reading and writing to the local disk.

Multi Threading

- Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

First Java Program

Demo

Hello Java!

```java
public class Welcome {
    /**
     *print the Hello message to the console.
     *@param args the command line.
     */
    public static void main(String[] args) {

        System.out.println("Welcome to Java!");
    }
}
```

Compiling & Running

- javac : is the primary Java compiler included in the JDK. it is used to compile java programs, it takes .java file as input and produces bytecode. The bytecode file ended with .class file extension.
- java : The java commend is used to execute the bytecode of java. It takes byte code as input and runs it and produces the output.

Programming Errors

- syntax Errors
- Runtime Errors
- Logic Errors

Programming Errors

- Syntax Errors

  Errors that are detected by the compiler are called *syntax errors* or *compile errors*. Syntax errors result from errors in code construction, such as mistyping a keyword, omitting some necessary punctuation, or using an opening brace without a corresponding closing brace.These errors are usually easy to detect because the compiler tells you where they are and what caused them.

Programming Errors

- Runtime Errors

  Runtime errors are errors that cause a program to terminate abnormally. They occur while a program is running if the environment detects an operation that is impossible to carry out. Input mistakes typically cause runtime errors.

Programming Errors

- Logic Errors

Logic errors occur when a program does not perform the way it was intended to. Errors of this kind occur for many different reasons.