OOP
Object Oriented Programming

for: SCII- SCCI- & SCCJ- 2021
Jan - April 2024

By: Salesio M. Kiura
Department of Computer Science and Informatics
School of Computing and Informatics,
Technical University of Kenya (TU-K)

4

Session 4:

Java info brief Overview,

Demo of Hello World, Anatomy of Java Applications

Date: __8th February 2024__Online___

Recap

- *What do we know so far?*

TODAY

- Demo Hello World Application, Overview of the structure (Anatomy) of a Java program, Objects, constructors

JAVA

- Is used for creating:
  - intelligent consumer-electronic devices (cell phones)
  - Web pages with dynamic content
  - large-scale enterprise applications
  - etc

Java life cycle

- Java programs normally undergo four phases:
  - **Edit (**Source code (.java)**)**

    Programmer writes program (and stores program on disk)
  - **Compile (**Byte codes (.class) , as (.exe) in c++

    Compiler creates bytecodes from program (.class as .exe in c^{++})
  - **Load**

    Class loader stores bytecodes in memory
  - **Execute**

    Interpreter: translates bytecodes into machine language

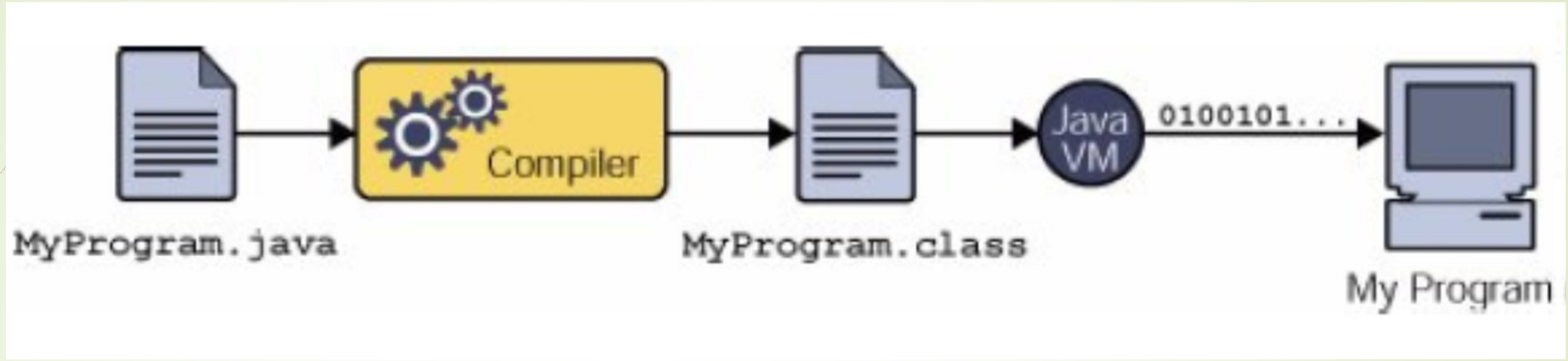
Other Concepts

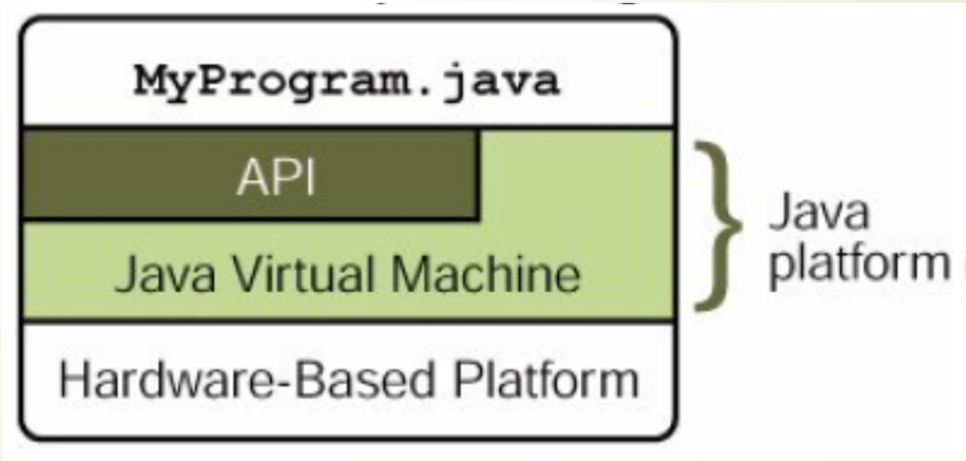
– The Java Application Programming Interface (API)

- a large collection of ready-made software components. It is grouped into libraries of related classes and interfaces; these libraries are known as packages.
- E.g. System.out.*; java.util.*

– Java Virtual Machine (JVM)

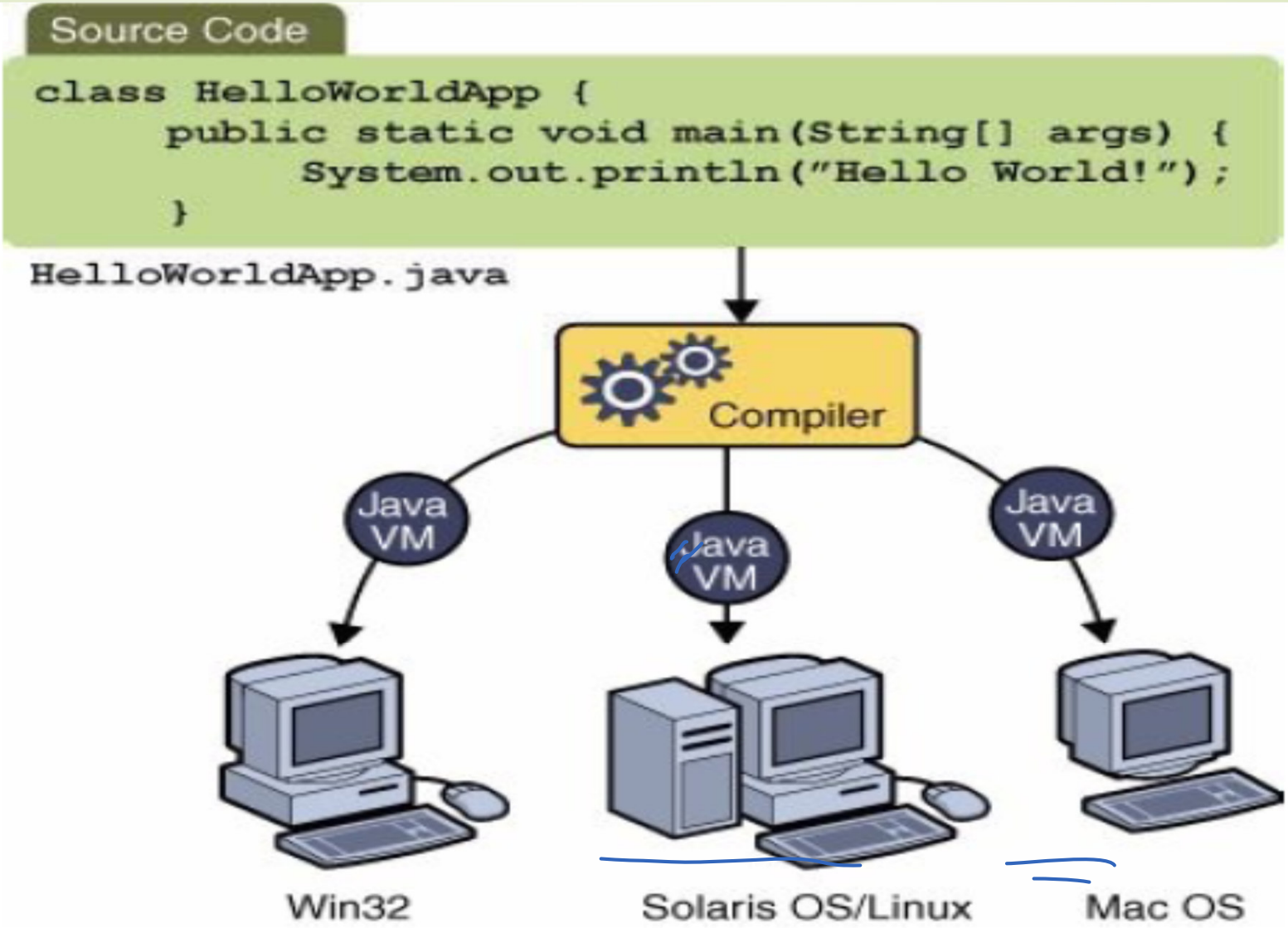
– Machine code (platform dependent)


MyProgram.java
Compiler
MyProgram.class
Java VM
0100101...
My Program


MyProgram.java
API
Java Virtual Machine
Hardware-Based Platform
Java platform

Portability of Java

- Through the Java VM, the same application is capable of **running on multiple platforms**


Source Code
class HelloWorldApp {
public static void main(String[] args) {
System.out.println("Hello World!");
}
HelloWorldApp.java
Compiler
Java VM
Java VM
Java VM
Win32
Solaris OS/Linux
Mac OS

Java technologies (1of 2)

Java technology series

- Java EE vs. Java SE
  - EE: enterprise edition (web services, distribution, RMI, …)
  - SE: standard edition (stand alone applications)

Development Tools

- The main tools you'll be using are the javac compiler, the java launcher (java), and the javadoc documentation tool.

Application Programming Interface (API)

- Java SE Development Kit 8 (JDK 1.8)
- Offers a wide array of useful classes ready for use in your own applications.

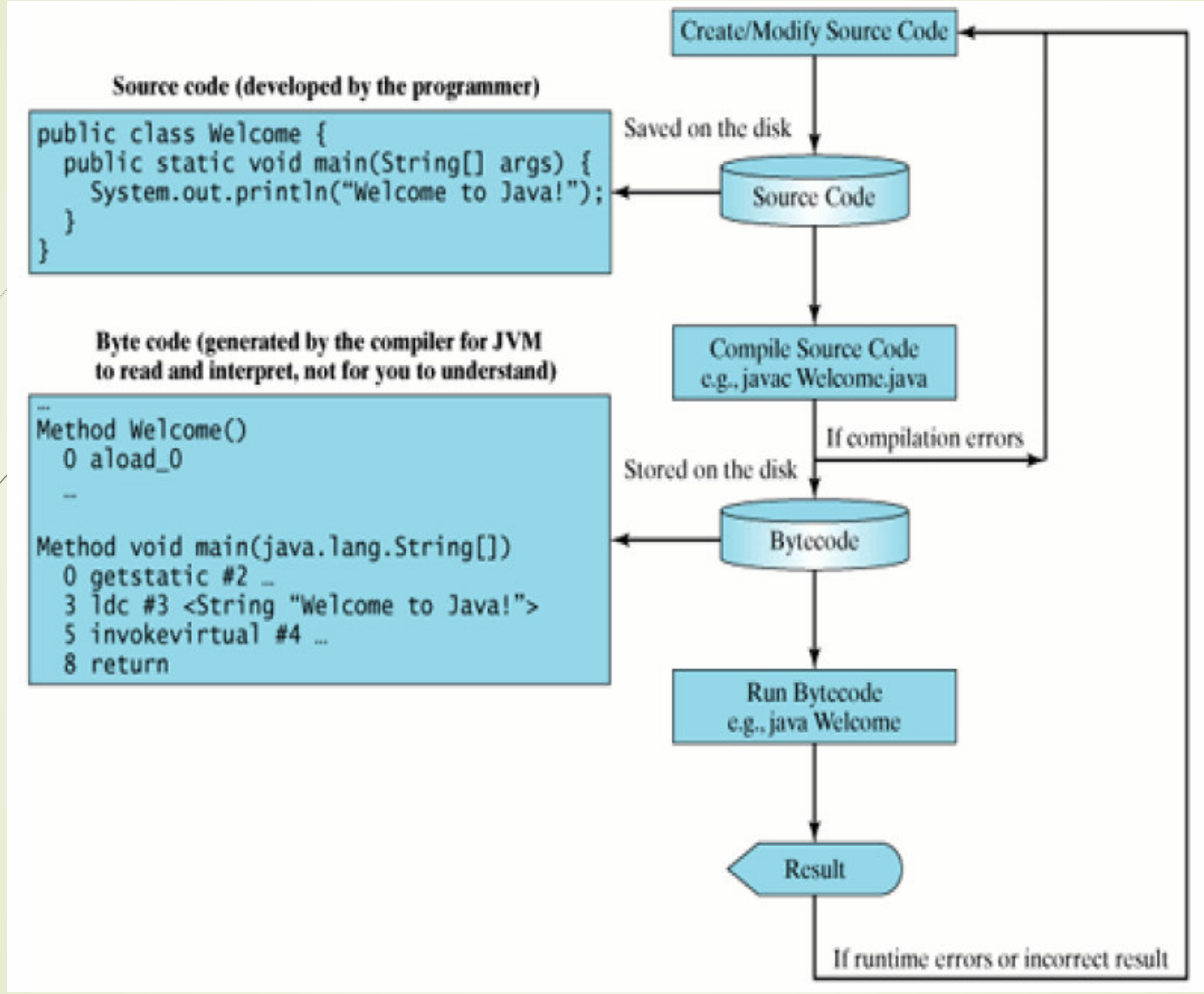
Java technologies (2 of 2)

- User Interface Toolkits
  - Swing and Java 2D toolkits to create Graphical User Interfaces (GUIs).

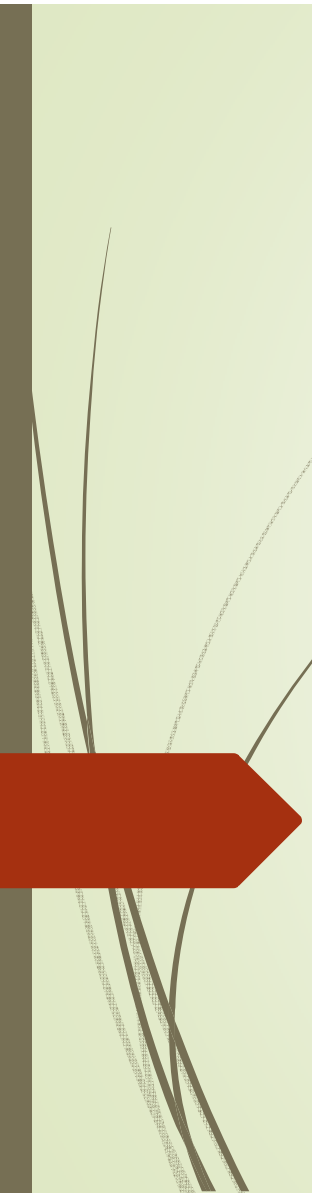
- Integration Libraries
  - Application Programming Interface (API)
  - Java RMI (Remote Method Invocation) over Internet and Inter-ORB (Object Request Broker) protocol technology enable database access

Demo of Sample CODE

- Basic "Hello World" application
- Computation in Hello World
- User input via the keyboard (Using the Scanner class)
- User input via an (input) dialog message


Source code (developed by the programmer)
public class Welcome {
public static void main(String[] args) {
System.out.println("Welcome to Java!");
}
}
Saved on the disk
Source Code
Byte code (generated by the compiler for JVM
to read and interpret, not for you to understand)
...
Method Welcome()
0 aload_0
...
Method void main(java.lang.String[])
0 getstatic #2 ...
3 ldc #3 <String "Welcome to Java!">
5 invokevirtual #4 ...
8 return
Create/Modify Source Code
Compile Source Code
e.g., javac Welcome.java
If compilation errors
Stored on the disk
Bytecode
Run Bytecode
e.g., java Welcome
Result
If runtime errors or incorrect result

- Student to share screen and be guided to write Code, compile and show output



Anatomy of a Java Application

Anatomy?

- Definition
- Importance

A closer look at the hello World App

```
public class HelloWorld
{
   //main method
      public static void main (String args [])
      {
         System.out.println("Hello World!");
      }
}
```

- In the Java programming language, every application must contain a main method whose signature is:

- public static void main(String[] args)
  - The modifiers public and static can be written in either order (public static or static public).

  - You can name the argument anything you want, but most programmers choose "args" or "argv."
  - This is the string array that will contains the command line arguments.

```
public class HelloWorld
{
   //main method
      public static void main (String args [])
      {
         System.out.println("Hello World!");
      }
}
```

- The main method is the entry point for your application and will subsequently invoke all the other methods required by your program.

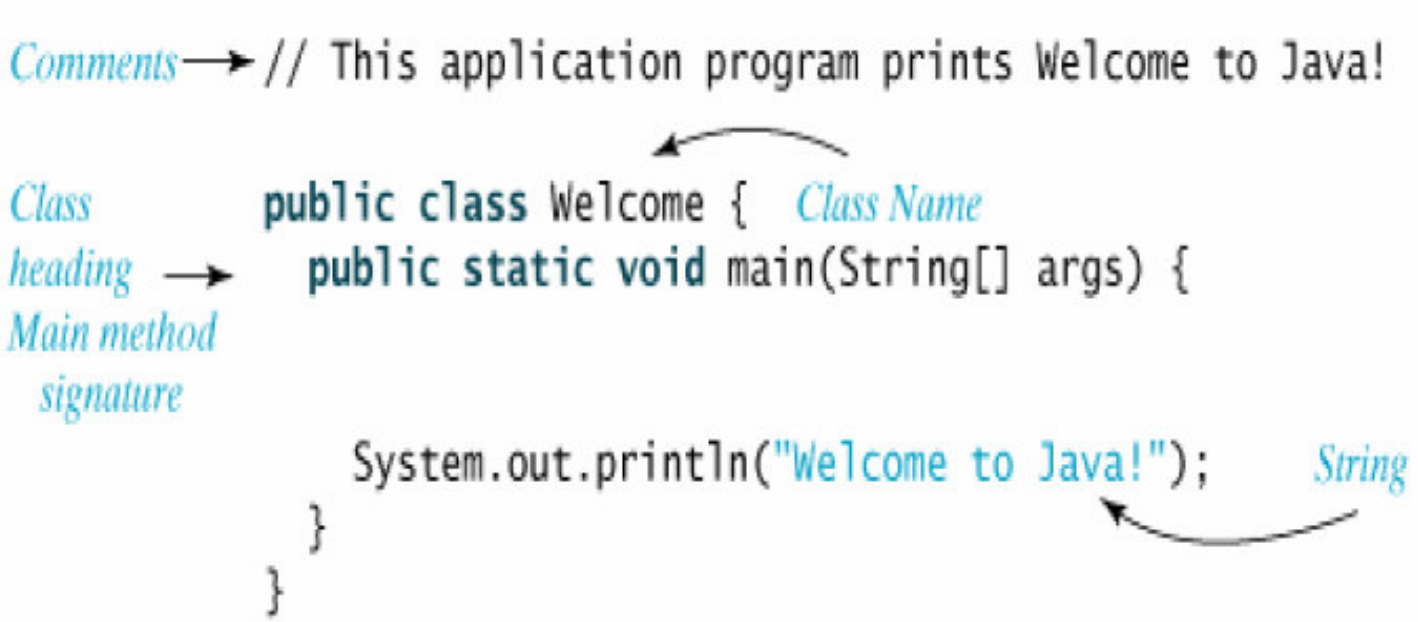
- System.out.println("Hello World!");
  - uses the System class from the API to print the "Hello World!" message to standard output.

Anatomy of a Java Program – the main aspects

1. Comments
2. Modifiers
3. Statements
4. Blocks
5. Classes
6. Methods
7. The main method
8. Package
9. Reserved words

Anatomy of a Java Program

Welcome.java



Comments

- In Java, comments are preceded by two slashes (//) in a line, or enclosed between /* and */ in one or multiple lines.
- When the compiler sees //, it ignores all text after // in the same line.
- When it sees /*, it scans for the next */ and ignores any text between /* and */.

Modifiers

- Java uses certain reserved words called modifiers that specify the properties of the data, methods, and classes and how they can be used.
- Examples of modifiers are public and static.
- Other modifiers are private, final, abstract, and protected.
  - A public datum, method, or class can be accessed by other programs.
  - A private datum or method cannot be accessed by other programs. Modifiers are discussed later in the course

Statements

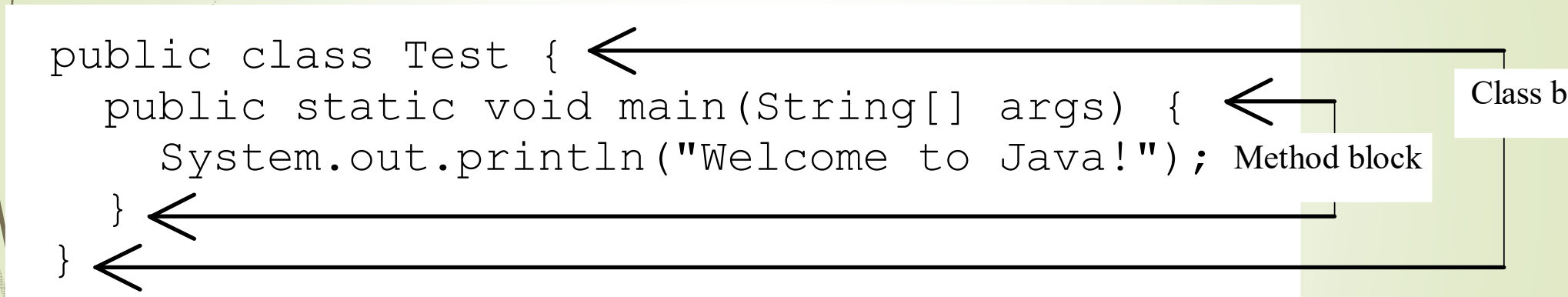
- A statement represents an action or a sequence of actions.

The statement System.out.println("Welcome to Java!"); in the exercise is a statement to display the greeting "Welcome to Java!“

- Every statement in Java ends with a semicolon (;)

Blocks

- A pair of braces in a program forms a block that groups components of a program.



- All statements within a block are executed sequentially

Classes

- The class is the essential Java construct.
- A class is a template or blueprint for objects.
- To program in Java, you must understand classes and be able to write and use them.
- The mystery of the class will continue to be unveiled throughout this course. For now, though, understand that a program is defined by using one or more classes.

Methods

What is System.out.println?

- It is a *method*: a collection of statements that performs a sequence of operations to display a message on the console.
- It can be used even without fully understanding the details of how it works.
- It is used by invoking a statement with a string argument. The string argument is enclosed within parentheses. In this case, the argument is "Welcome to Java!" You can call the same println method with a different argument to print a different message.

main Method

- The main method provides the control of program flow. The Java interpreter executes the application by invoking the main method.
-
- The main method looks like this:

```
public static void main(String[] args) {
  // Statements;
}
```

Packages

- A package is a namespace for organizing classes and interfaces in a logical manner.
- A namespace organizes a set of related classes and interfaces
- Placing your code into packages makes large software projects easier to manage
- Conceptually you can think of packages as being similar to different folders on your computer. For a website project, you might keep HTML pages in one folder, images in another, and scripts or applications in yet another.

Packages (contd...)

- Because software written in the Java programming language can be composed of hundreds or thousands of individual classes, it makes sense to keep things organized by placing related classes and interfaces into packages.
- The Java platform provides an enormous class library (a set of packages) suitable for use in your own applications. This library is known as the "Application Programming Interface", or "API" for short. Its packages represent the tasks most commonly associated with general-purpose programming. For example:
  - Print to screen, String operations, etc

API definition

- An application program interface (API) is a set of routines, protocols, and tools for building software applications
- Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components
- A good API makes it easier to develop a program by providing all the building blocks. A programmer then puts the blocks together.

- Already used API reference:
  - System.out…
  - java.util.Scanner
  - javax.swing.JOptionPane

Ref: https://docs.oracle.com/javase/7/docs/api/

Reserved words

- aka ??
- Same meaning as in C++ ☺
  - Words that have a defined meaning in a given programming language
  - You can't use those words as your defined variable names
    - Examples: int, class, boolean, false, public, private, Object!

Reflection: Anatomy of a Java Program

- Ref: the "TU-K Gate Management System"
  - The application to manage visits in a university

- Model Classes
- Model the flow (flow chart)
- Code (only to an extent)

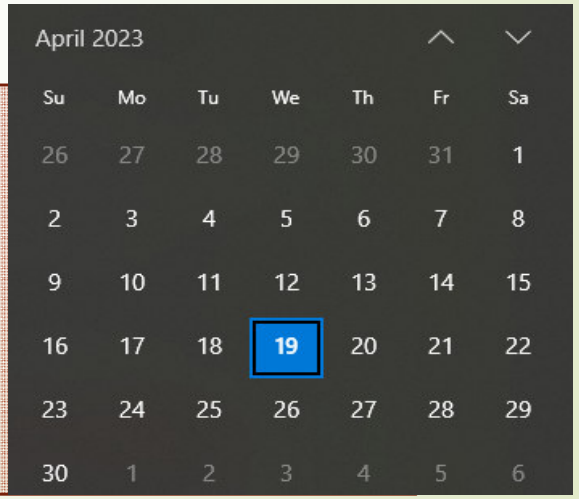
Thanks!

Your project!

Reflection Example: Anatomy of a Java Program

- Ref: the "Date" Assignment
  - The application requires the user to enter a date in April 2023, your program displays the day of the week.

- E.g.
  - Input 1, output: Saturday
  - Input : 10, output : Monday ➔➔
  - Input: 27, output : Thursday



Thank you