CT60A2411
Object Oriented Programming: Java
Week 1A

Ashok Kumar Veerasamy, PhD

Learning objectives: continued

- History of Java
- What is different about Java?
- Java's design philosophy
- Java IDE tools
- First Java Program
- Anatomy of a Java Program

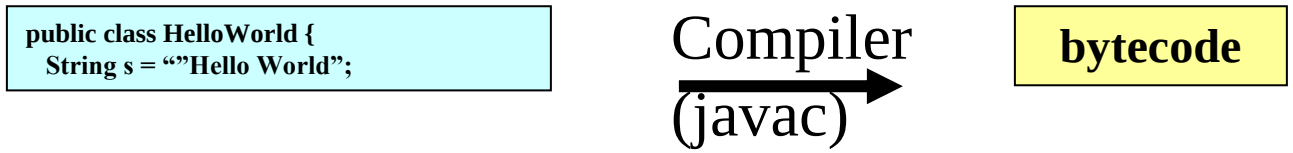
At the conclusion of this lecutre, students will be able to understand the how to start writing and running Java programs using Apache NetBeans code editor.

History of Java

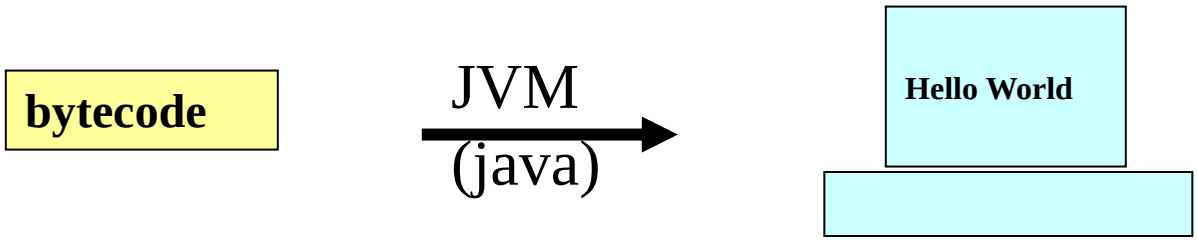
- Developed by Sun Micro Systems and become a part of Oracle now.
- It is an object-oriented programming language
- Used for developing platform independent applications

- What is different about Java?

Java compilers translate source code into an intermediate language called (platform independent) *bytecode*.



Java Virtual Machine (JVM) reads the bytecode (loads the necessary library bytecode) and executes it on a specific machine.



Unlike machine code, Java bytecode is not tied to any particular machine making it architecture neutral.

Java's design philosophy

There were five primary goals in the creation of the Java language:

- It should be "simple, object oriented, and familiar".
- It should be "robust and secure".
- It should be "architecture neutral and portable".
- It should execute with "high performance".
- It should be "interpreted, threaded, and dynamic".

Java IDE tools

- **Apache NetBeans**
  https://netbeans.apache.org/

- **Eclipse**
  https://www.eclipse.org/downloads/

- **IntelliJ Idea**
  https://www.jetbrains.com/idea/

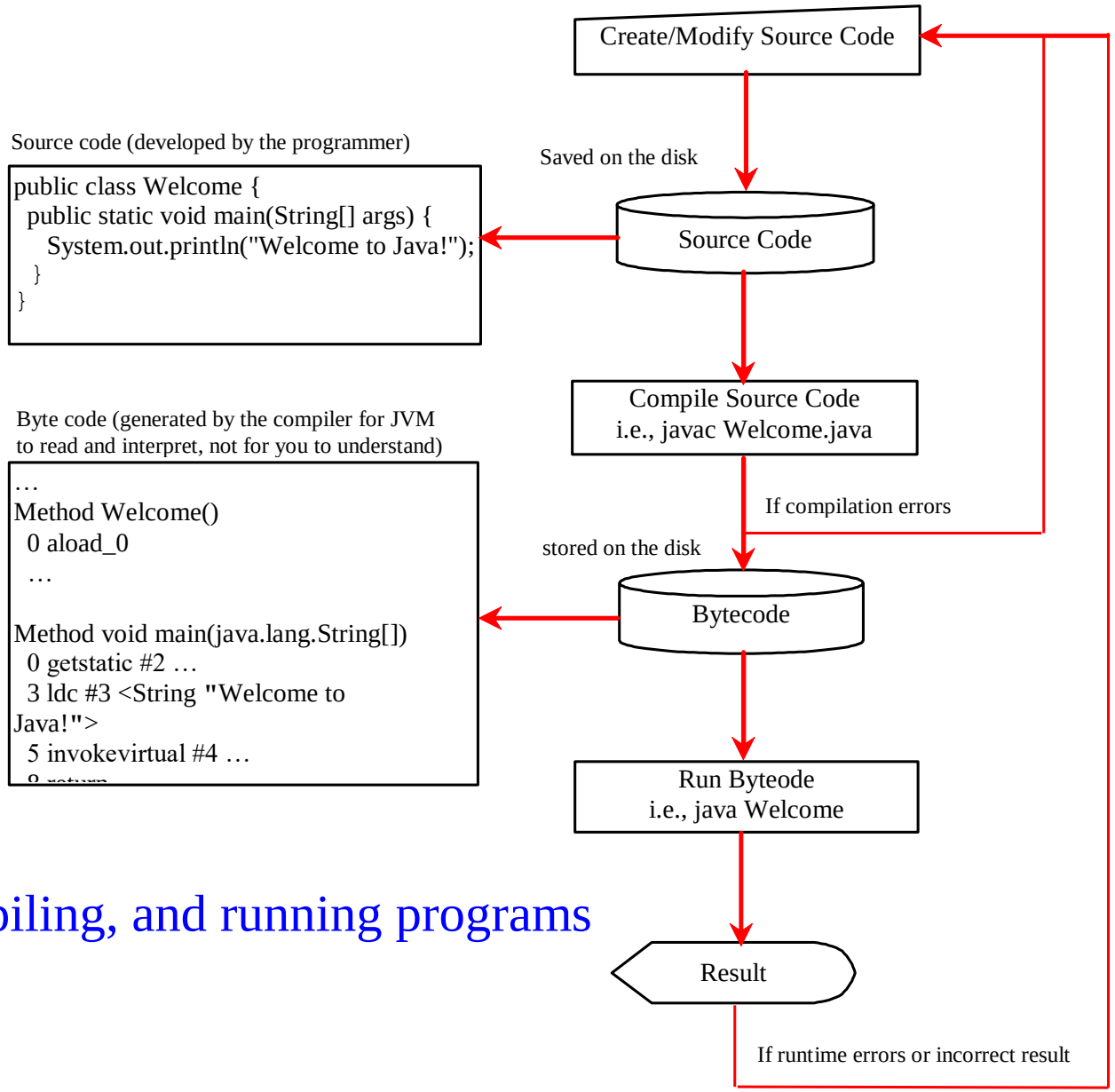
How to install JDK and IDE for Java

- Refer the tab ”JDK and IDE for Java” in Moodle
- JDK version 8.0 or above is recommended
- Apache NetBeans 11.3 or above is recommended. But you are free to use other IDEs’ as well

First Java Program

```
//This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

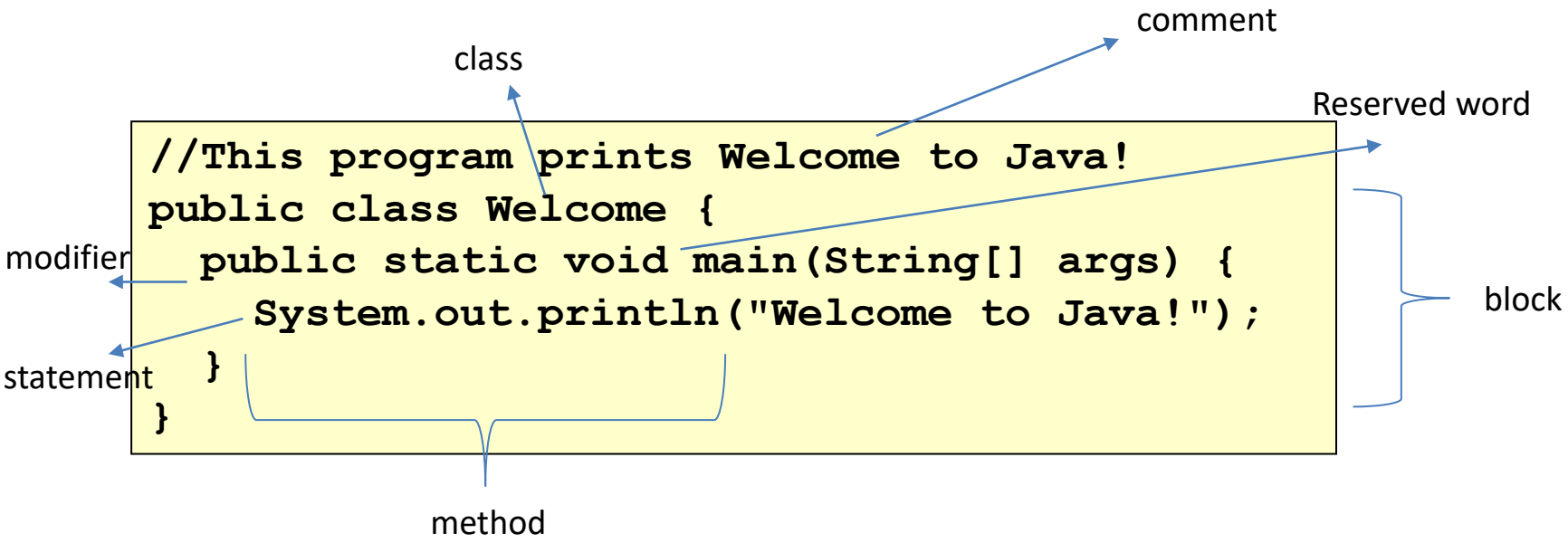
Creating, compiling, and running programs



Anatomy of Java program

- Comments → starts with // or /*
- Package → project name at present
- Reserved words → keywords such as public, static, void …
- Modifiers → Example access modifier such as public, private..
- Statements → lines of code / source code
- Blocks → the block of code enclosed by { }
- Classes → a template that define/contain the object and methods
- Methods → subprogram which are part of the class
- The main method → Where the defined class and its properties called for execution

- **Anatomy of Java program**



- **How to create, compile and run Java code?**

  Check the video posted in the Moodle.

Quiz: What is the output ?

```
// One of the first Java programs
// It prints out Hello World and the year

public class HelloWorld {
  public static void main(String[] args) {
    int year = 0;
    // year = 2009;
    System.out.println("Hello World " + year);
  }
}
```

(a) Hello World
(b) Hello World 0
(c) Hello World 2009

What will be the output when we compile and interpret the code below ?

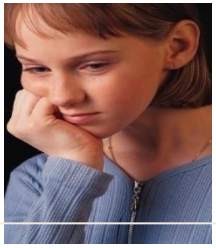
```
public class HelloThere  {
   public static void anotherMethod( )   {
      System.out.print("There ");
   }

   public static void main(String[] args)   {
      System.out.print("Hello ");
   }
}
```

(A) Hello
(B) There
(C) Hello There
(D) There Hello

What will be the output now ?
(note the additional line in main())

```
public class HelloThere  {
   public static void anotherMethod( )   {
      System.out.print("There ");
   }
   public static void main(String[] args)   {
      System.out.print("Hello ");
      anotherMethod();
   }
}
```



(A) Hello
(B) There
(C) Hello There
(D) There Hello

Why do we use methods?