Introduction to Java Programming

Part 1

Java Basics

Overview

- A Brief History of Java

- How To Talk To A Computer

- Basic Java Architecture

- Introductory Java Constructs

- References

A Brief History of Java

- Java is a “new” programming language
  - Developed in 1991 by Sun Microsystems engineers
  - First public release in 1995
  - Open-sourced in 2006
  - Sun Microsystems bought by Oracle in 2010

A Brief History of Java

- Java is unique as a programming language
  - Designed as “Write Once Run Anywhere” (WORA)
  - Syntax designed to mimic C/C++
  - Object-oriented
  - Second most popular programming language in the world behind C/C++

How To Talk To A Computer

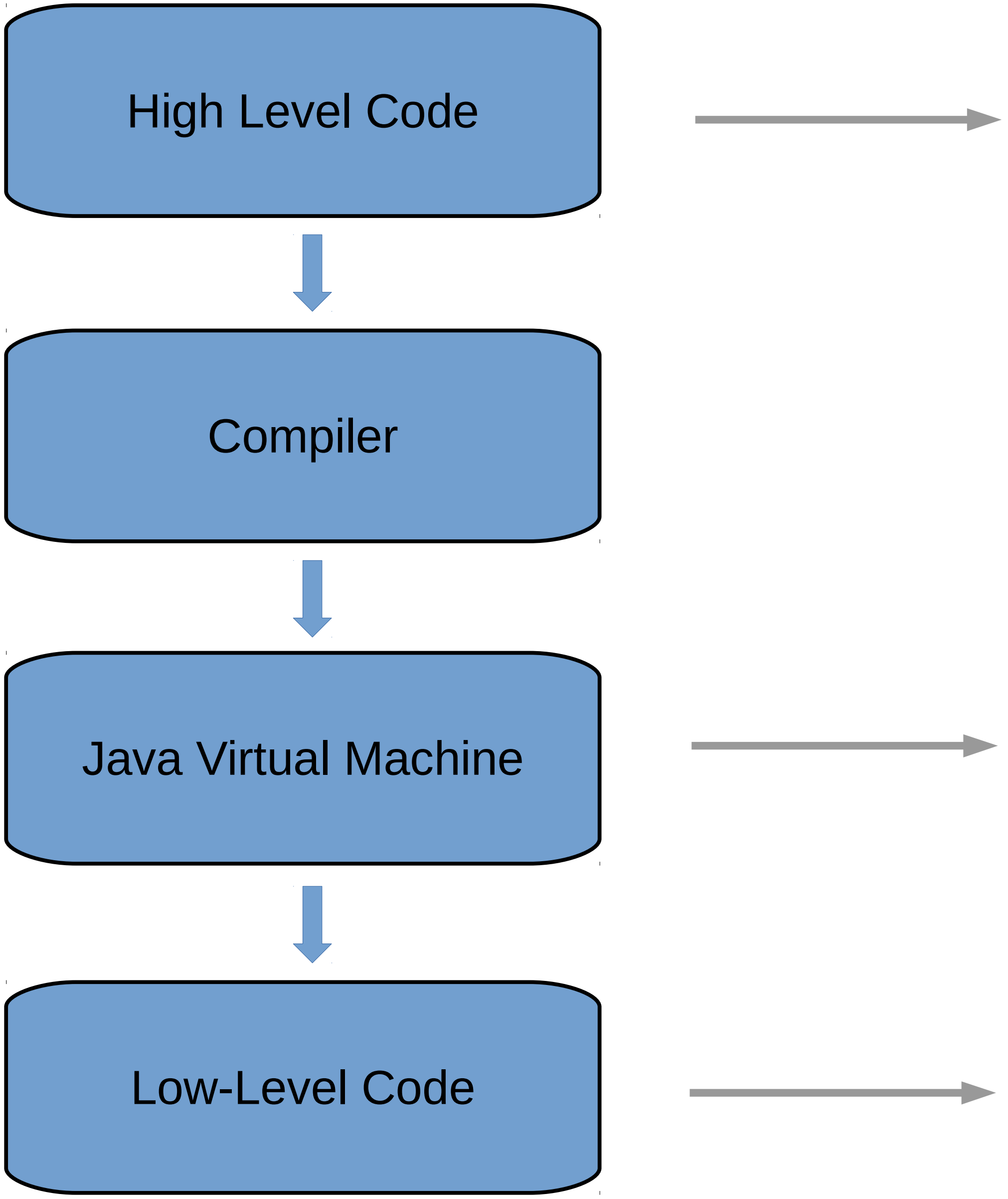
- 1010011 1110100 1100001 1110010 1110010 1111001 1001011 1101110 1101001 1100111 1101000 1110100 1110011

- “Starry Knights” in 7-bit ASCII code

- Obviously programming like this would be miserable

- Modern computer programs use “high level” language which is then converted to “low level” language using programs called compilers

How To Talk To A Computer



High Level Code

Compiler

Java Virtual Machine

Low-Level Code

```
public class Person {
        private String name;
        private int age;
        private int height;
}
```

```
Method Person(java.lang.String,int)
0 aload_0
1 invokespecial #3 <Method java.lang.Object()>
4 aload_0
5 aload_1
6 putfield #5 <Field java.lang.String name>
9 aload_0
10 iload_2
```

```
1001000 1000101 1001100
1001100 1001000 1001001
1010100 1000011 1001000
```

How To Talk To A Computer

- Java uses a device called the Java Virtual Machine (JVM) to accomplish it’s goal of “WORA”

- JVM acts as an intermediate step to convert Java bytecode to run on a specific processor/architecture

  - Not every computer uses the same processor/chipset

  - ARM Cortex / Intel x86 / SPARC / RISC32 / PowerPC are just a few examples of the many options

Bottom Line: JVM allows you to write one program and run on many different types of hardware

Introductory Java Constructs

- Java is “object oriented”
  - This means you can create almost anything you want in the programming language – you just have to define it in a way that the compiler can understand
  - In this lesson, we will look at the following Java constructs:

    1) Packaging

    2) Classes

    3) Class Attributes

Introductory Java Constructs

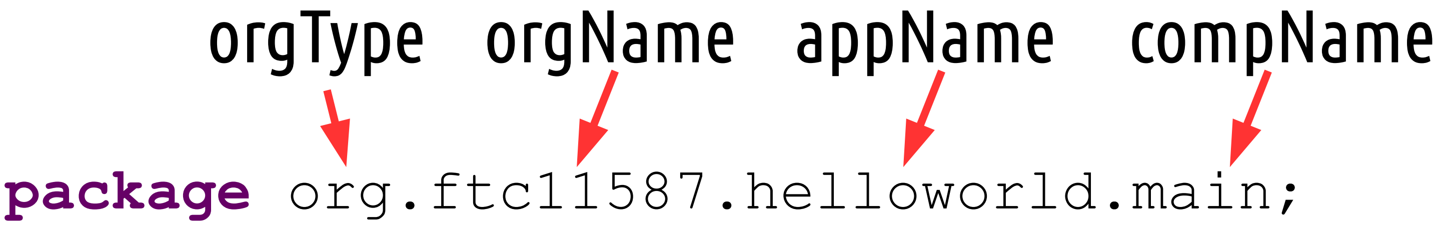
- Packaging
  - Reference your sample code file Line 16

    ```
    package org.ftc11587.helloworld.main;
    ```

  - Packages are used in Java to protect *namespace*
    - Two different things with the same name = BADNESS
    - If you put two things with the same name in two different “rooms”, it’s not a problem anymore

Introductory Java Constructs

- Packaging
  - The "package" construct in Java is simply a way of building a "room" for something to go in without confusing it with something else
  - Java uses a naming convention for package names to help keep them straight across multiple projects
  - The convention is depicted below:



orgType orgName appName compName

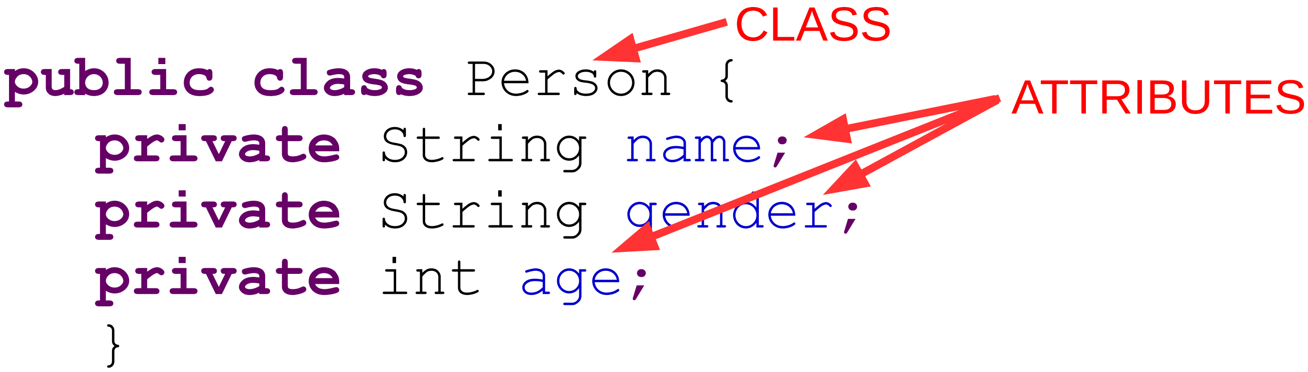
```
package org.ftc11587.helloworld.main;
```

Introductory Java Constructs

- Classes & Attributes

  - Classes are simply things (objects) that you create in Java

  - Classes have attributes (that you define)



CLASS

ATTRIBUTES

```
public class Person {
  private String name;
  private String gender;
  private int age;
  }
```

  - Attributes are simply things which define/characterize that particular class

  - Attributes are normally declares first in the Class definition

Introductory Java Constructs

- Classes & Attributes
  - When you define an attribute, you also have to define that attribute’s dataType

```
public class Person {
   private String name;
   private String gender;
   private int age;
   }
```

dataTypes

  - Java needs you to tell it the difference between 13 and “13”

```
13 + 13 = 26      “13” + “13” = “1313”
```

  - Attributes are normally contained in *variables*

References

- IBM DeveloperWorks
  https://www.ibm.com/developerworks/java/tutorials/j-introtojava1
- Code Academy
  https://www.codecademy.com/learn/learn-java/