Module 1 - Lecture 2

Variables and Data Types

Yesterday

- File systems
- What is a Shell?
- Git overview

Java Overview

- Java is an object-oriented programming language.
- It’s portable. Write once, run anywhere.
- First created at Sun Microsystems in 1995, now controlled by Oracle.

Java vs Related Languages

- C/C++
  - Similar syntax
  - Java manages memory and is portable
- JavaScript
  - Originally developed by Netscape in 1995 as “LiveScript”
  - No technical relationship
- C#
  - Released by Microsoft with .NET platform in 2000
  - Similar features and syntax

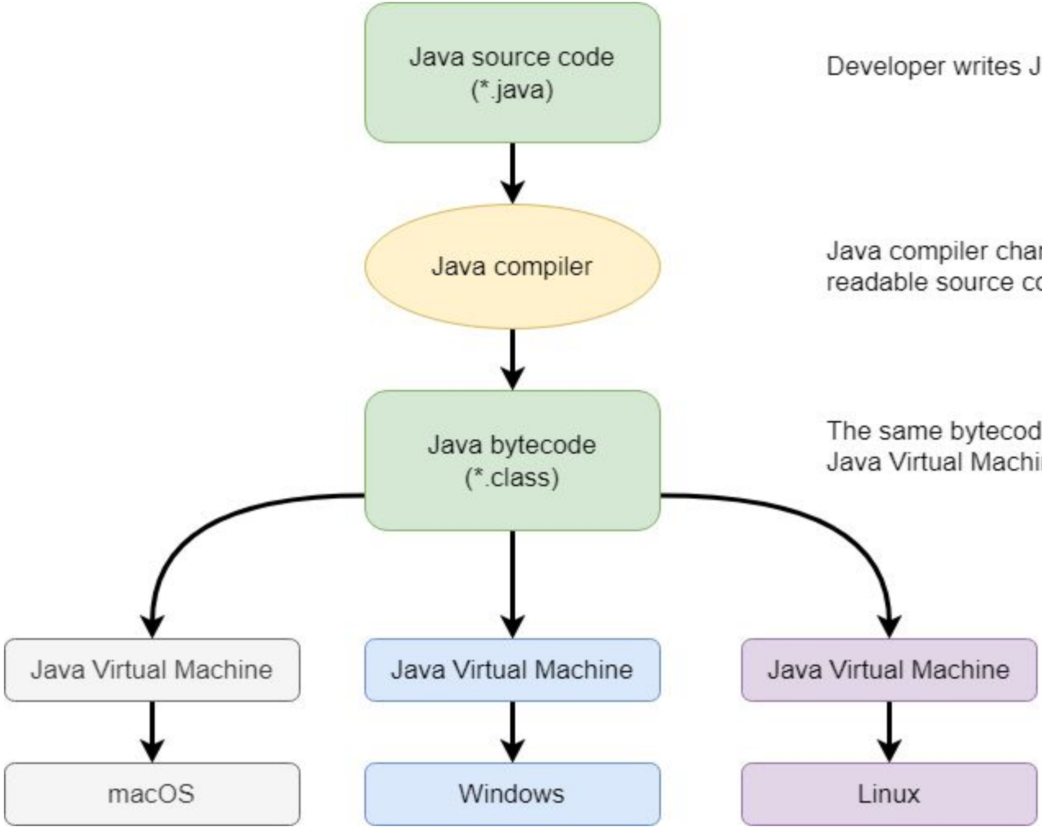
Parts of Java

- Java Virtual Machine (JVM)
  - The JVM interprets the **bytecode** into machine code and executes it.

- Java Runtime Environment (JRE)
  - Contains the tools required to **run** Java programs.
  - Contains the JVM along with Java’s built-in libraries/packages

- Java Development Kit (JDK)
  - Contains the tools required to **write** Java programs.
  - Contains a JRE, javac compiler, etc.

How is Java portable?



Developer writes Java source code

Java compiler changes human-readable source code into bytecode

The same bytecode runs on any Java Virtual Machine (JVM)

JVMs exist for many different platforms

IntelliJ

- IntelliJ is an Integrated Development Environment (IDE).
- It provides features to increase efficiency for software developers.
    - An easy to use user interface
    - Code organization
    - Immediate feedback when errors occur
    - Debugging
    - Syntax highlighting
    - Intellisense

Let’s Code!

What is a Program?

A program is made up of two parts, **data** and **behavior**.

Today’s focus: **data!**

Variables

- A variable is a storage container paired with a symbolic name or identifier.
- Variable names must be unique.
- Variables contain a **value** of a certain **type.**
- Variables come about through **declaration**, and **initialization/assignment**.

Java Primitive Data Types

Type	Contains
boolean	true or false
byte	-128 to 127
short	-32,768 to 32,767
char	‘a’, ‘b’, ‘c’, or any Unicode character
int	-2^31 to 2^31 - 1
long	-2^63 to 2^63 - 1
float	-3.4 * 10^38 to 3.4 * 10^38 - 1
double	±5.0 × 10^−324 to ±1.7 × 10^308

String

A string represents a sequence of zero or more Unicode characters.

- **Declaring a String**
  With `"..."`
- **Escape Characters**
  Like `\n` and `\t`. Needed because you can't really type a tab or return in a string directly.

Variable Declaration in Java

```
<data_type> <variable_name>;
```

```
int myAge;
String myName;
boolean isInstructor;
```

Each of the statements above **declare** a variable with a data type and a name.

Variable Initialization in Java

```
<data_type> <variable_name> = <value>;
```

```
int myAge = 50;
String myName = “Billie”;
boolean isInstructor = true;
```

Each of the statements above **initialize** a variable with a data type, a name, and a value.

Variable Assignment in Java

```
<variable_name> = <value>;
```

```
myAge = 50;
myName = “Billie”;
isInstructor = true;
```

Each of the statements above **assign** a variable with a a name to a value.
An assignment statement is only valid if the variable has already been declared or initialized!

Variable Naming

- Follow camel-casing conventions e.g. helloWorld. The first word is lowercase, and subsequent words have the first letter capitalized.
- For constants use UPPERCASE_WORDS_SEPARATED_BY_UNDERSCORES
- Use pronounceable names for variables
- Use names over single characters
- Avoid creating multiple variables that are variations of the same name, as this creates confusion
- Use names that describe what the variable contains
- With booleans, use names that start with is, has, was, and so on; avoid using a double negative

CANNOT

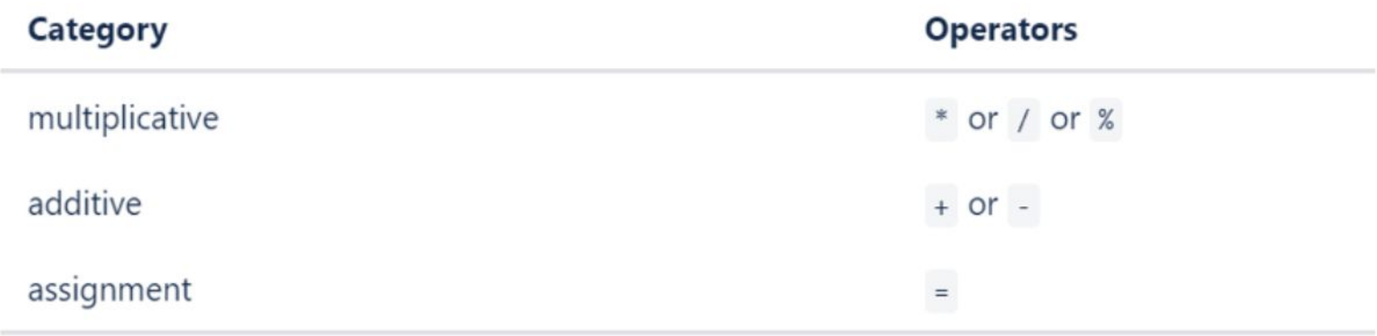
- Start with a number
- Use keywords

Let’s Code!

Expressions

An **expression** is statement of code which can be evaluated to produce a result. We use the result to often assign the value to another variable or as the input to another expression.

Today we will focus on **arithmetic expressions**.



Category	Operators
multiplicative	`*` or `/` or `%`
additive	`+` or `-`
assignment	`=`

Let’s Code!

Type Conversion

There are two types of casting that occur.

Widening / Implicit Casting occurs when we convert from one type with less size to a type with more size.

Narrowing occurs when we convert from a type with more size to a type with less size. **Truncation** occurs when you go from larger to smaller and have to lose some of the data in the process.

Let’s Code!

Binary

QUESTIONS?