

Module 1-2

Variables and Data Types

History of Java

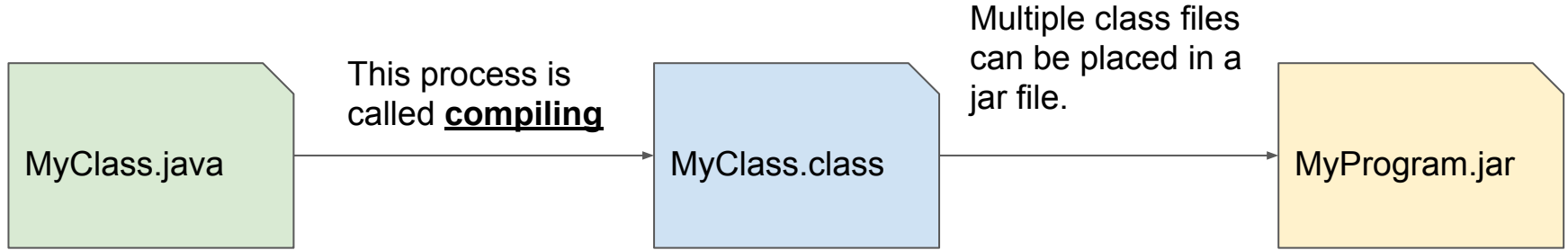
Java is an object oriented language (you will learn what this means later) developed by Sun Microsystems in 1995. Sun Microsystems was acquired by Oracle Corporation in 2010. Oracle currently owns the Java platform.

It is one of the most widely used languages. It consistently ranks in the top 2 in terms of popularity. (Source: <https://stackify.com/popular-programming-languages-2018/>).

Java Features

- It shares similar syntax with C/C++
- It can be used to create desktop, mobile, and web applications.
- Unlike other languages, Java is not run natively on a given device, it is instead executed in a Java Virtual Machine (JVM) / Java Runtime Environment (JRE).... think of this as a virtual computer running inside your computer.
- Advantages of this model:
 - Oracle (not you :)) is responsible for the lifecycle of the JRE / JVM.
 - Developers are therefore freed from the idiosyncrasies of each individual platform! (i.e. pc's, macs, mobile devices, refrigerators, etc)

Java Development Workflow



The files containing our code end with a .java extension. **These files are our source code.**

In the past, the command **javac** was used to compile code, since the advent of modern development tools, manually typing this command is no longer needed.

Java Bytecode Example

```
outer:
for (int i = 2; i < 1000; i++) {
    for (int j = 2; j < i; j++) {
        if (i % j == 0)
            continue outer;
    }
    System.out.println (i);
}
```



This is what
we will write

A Java compiler might translate the Java code above into byte code as follows, assuming the above was put in a method:

```
0:  iconst_2
1:  istore_1
2:  iload_1
3:  sipush 1000
6:  if_icmpge 44
9:  iconst_2
10: istore_2
11: iload_2
12: iload_1
13: if_icmpge 31
16: iload_1
17: iload_2
18: irem
19: ifne 25
22: goto 38
25: iinc 2, 1
28: goto 11
31: getstatic #84; // Field java/lang/System.out:Ljava/io/PrintStream;
34: iload_1
35: invokevirtual #85; // Method java/io/PrintStream.println:(I)V
38: iinc 1, 1
41: goto 2
44: return
```



This is what
the java
translates it to

Source: https://en.wikipedia.org/wiki/Java_bytecode

Java Program Structure

- The main unit of organization in Java is a class. Here is a simple example:

This file must be called MyClass.java. Its contents are as follows:

```
public class MyClass {  
    public static void main(String[] args) {  
        int i = 1;  
    }  
}
```

This must match the file name. It is the name of the class.

This is a variable called i, which currently stores a value of 1.

All java statements end with a semicolon.

This is a method called main... main is a special method that determines what gets run when the program executes

Java Variables

- A variable is a representation for something that might change.

Consider this well known math formula: $c^2 = a^2 + b^2$

We don't know for a fact what a, b, or c are, they can take different values depending on the situation. Therefore, a, b, and c are like containers that could take on different values at different times. These containers are called **variables**.

Java Variables: Declaring and Assigning Values.

- This is what a java variable declaration looks like:

```
int i = 0;
```

Here we have declared a variable called i of type integer, we have also given it an initial value of zero. Assigning values is accomplished using equals (=).

- Consider this:

```
int i = 0;  
i = 1;
```

Here we have declared a variable called i of type integer, we gave it an initial value of zero, but then changed the value of the variable to i.

Java Variables: Common Data Types.

Type of Data	Suitable Java Data Type	Example
Integers, whole numbers	int, long, byte, short	int i = 1; long j =10L;
Decimals	double (preferred) float	double x = 1.23; float k =2.3f;
Text	char (for 1 character) String (many characters)	char myChar = 'a'; String myName = "Drew";
true or false	boolean	boolean isItOn = true; boolean paidBills = false;

Java Variables: Rules & Conventions

Conventions:

- Ideally, variables should be named using “Camel Case.” Examples of variable names: *playerOneScore*, *cityTemperature*, *shirtSize*, etc. (See the pattern?)
- Ideally, variables should never start with an upper case.
- Variable names should be descriptive and of reasonable length.

Rules:

- Variables can begin with an underscore (`_`), a dollar sign (`$`), or a letter.
- Subsequent characters can be letters, numbers, or underscores.
- Variable names cannot be the same as java keywords.

Working with Numbers: Basic Operators

- Math operators can be used to perform basic arithmetic between two numeric variables (From the previous slides, variables of type int, float, and double are examples).
- These are the basic operators: **+** (addition), **-** (subtraction), **/** (division), **%** (modulo, aka remainder).
- The basic operators can be combined with the assignment operator to store result calculations, for example: **int i = 4 + 6;**

Working with Numbers: Basic Operators

- Math operators can be used to perform basic arithmetic between two numeric variables (From the previous slides, variables of type int, float, and double are examples).
- These are the basic operators: **+** (addition), **-** (subtraction), **/** (division), **%** (modulo, aka remainder).
- The basic operators can be combined with the assignment operator to store result calculations, for example: **int i = 4 + 6;**

Working with Numbers: Order of Operations

The order of operations is the same as in normal arithmetic: **Please Excuse My Dear Aunt Sally**

- Anything inside **P**arentheses first.
- **E**xponents
- **M**ultiplication *
- **D**ivision
- **A**ddition *
- **S**ubtraction

* Note that even though the acronym suggests that multiplication outranks division, and addition outranks subtraction, multiplication and division have the same priority, likewise addition and subtraction have the same priority. In other words, between a multiplication and division, Java will evaluate the first one it encounters going left to right (just like in regular math).

Working with Numbers: Example 1

- Express the following English statement in Java: I paid for an item that cost \$8.50 with a \$10.00 bill, how much change would I get in return?

```
public class MyClass {  
  
    public static void main(String[] args) {  
  
        double price = 8.50;  
        double payment = 10.00;  
        double change = payment - price;  
        System.out.println(change);  
    }  
}
```

Working with Numbers: Example 2

- Let's try something a little bit more complex: We can convert degrees in Fahrenheit to Celsius using the following formula: $(T_F - 32) \times (5/9) = T_C$. How much is 98.6 degrees Fahrenheit in Celsius?

```
public class MyClass {  
  
    public static void main(String[] args) {  
  
        double tempInF = 98.6;  
        double tempInC = (tempInF - 32.0) * (5.0/9.0);  
        System.out.println(tempInC);  
    }  
}
```

Java Type Promotions

When multiple data types are used on the right side of an equals sign, Java observes the following type promotion rules.

- byte and short are both promoted to an int.
- If one operand is a long, the result is a long.
- If one operand is a float the result is a float.
- If one operand is a double the result is a double.

In some situations, these rules can be circumvented by using a casting operator. For example, placing (byte) in front of the operation casts the entire result to a byte.

Java Type Promotions Examples

Incorrect:

```
byte myByte = 4;  
byte myOtherByte = 1;  
  
byte firstAttempt = myByte + myOtherByte;
```

Correct:

```
byte myByte = 4;  
byte myOtherByte = 1;  
  
int firstAttempt = myByte + myOtherByte;
```

Correct (with casting):

```
byte myByte = 4;  
byte myOtherByte = 1;  
  
byte firstAttempt = (byte) (myByte + myOtherByte);
```

Java Type Promotions Examples 2

Incorrect:

```
long thisVariable = 4;  
int anotherVariable = 7;  
  
int someVariable1 = thisVariable + anotherVariable;
```

Correct:

```
long thisVariable = 4;  
int anotherVariable = 7;  
  
long someVariable1 = thisVariable + anotherVariable;
```

Java Type Promotions Examples 3

Here are some trickier examples, what do you think the result of these two operations are?

A

```
int k = 8 / 5
```

B

```
int k = 7 / 3 * 4
```

Java Type Promotions Examples 3 (continued)

- For problem A, 8 and 5 are int's, the result will be an int. Now $8/5$ does not result in a whole number (it is 1.6), but because the result is an int, we have to round the answer down to a 1.

A

```
int k = 8 / 5
```

- For problem B, we start out by dividing 7 and 3, which are both int's. The result is an int, but again we round down to 2. We then multiple this by 4 to get 8.

B

```
int k = 7 / 3 * 4
```

Combining Strings

The plus sign can also be used with Strings:

```
public class MyClass {  
  
    public static void main(String[] args) {  
  
        String firstName = "Homer";  
        String lastName = "Simpson";  
        String comma = ", "  
        String combinedName = lastName + comma + firstName;  
        System.out.println(combinedName);  
    }  
}
```

The following code will print ***Simpson, Homer***. This process is known as **concatenation**.