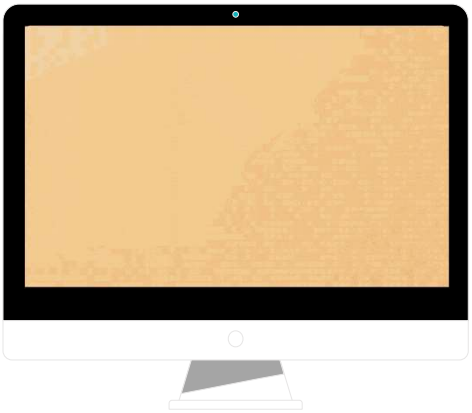
Recall: Front-End Fundamentals

1) What attribute do we need to use to ensure that a radio button is part of the same collection?
2) What is the code for getting a page element by it's name?
3) What attribute can we use to execute some JavaScript when an element is clicked?
4) Name <u>three</u> front end languages.
5) What is a framework? Can you name any?



1

08. Intro to Java

What is Java? What is Object-Oriented Programming?
What are primitive and non-primitive datatypes?
How do I setup my first class in Eclipse?
How do I declare variables?
How do I use mathematical operators?
What is the syntax for an IF statement?

01
Front-End
Development

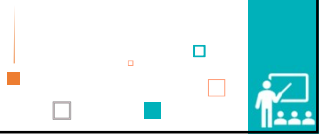
02
Back-End
Development

03
Agile
Methodology

2

Key words

- OOP
- Class-based
- Object-orientated
- Primitive
- Non-primitive
- Byte, Short, Int ,Long ,Float ,Double ,Boolean, Char
- String
- Array
- Classes
- Methods
- Interfaces
- Immutable



3

Task Icons

Trainer-led lecture

Code-along

Links to Certification

Independent task

Reflection

4

Learner Journey
Unit 2: Back-End Development

Intro to Java | Java Fundamentals | Object Oriented Programming | Testing with Junit | MySQL

5

Java

- Java is a high-level, class-based, object-oriented programming language designed to have as few implementation dependencies as possible.
  - High-level : Abstracts (hides) the complexities of a computer
  - Class-based : Uses classes of objects
  - Object-oriented : A concept where “data types” (for lack of a better term) contain data and code
- Designed in 1991 by James Gosling, Mike Sheridan, and Patrick Naughton and acquired by Oracle 2009.
- Originally designed for interactive television and based on the C/C++ syntax.
- It’s name, Java, comes from java coffee.
- Java can be ran everywhere – From computers, mobiles, game systems, data centres, and even super-computers!
- More than 3 **billion** devices run Java

6

Data Types in Java

- There are two over-arching groups of data types within Java:
  - Primitive
  - Non-Primitive
- Primitive are pre-defined data types that specifies size and type of variable values and have no methods attached to it.
- Non-Primitive are objects built from classes and have methods attached to them.

7

Data Types in Java: Primitive

- Note; String is **not** a primitive data type! This is because it is an object and has methods attached to it.

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

8

Data Types in Java: Non-Primitive

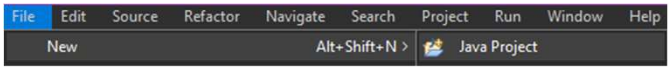
- Non-Primitive data types are created from classes, often by the programmer
- Can have methods/functions called **from** them. For example:
  - myString.toLowerCase()
- Cannot have their value set to null (primitives can be null)
- Start with a capital letter (primitives start with a lower-case letter)!

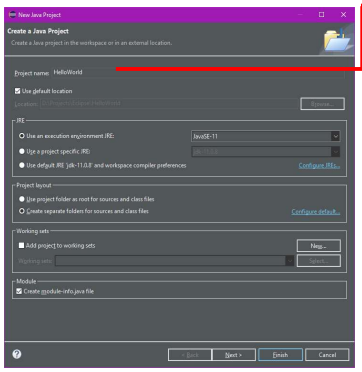
- Strings, Arrays, Classes, Interfaces are all non-primitive

9

Hello World: Setting Up

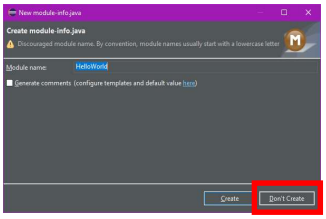
- Let us start investigating Java data types by looking at a default, "Hello World" project.
- Within Eclipse, go to File -> New -> Java Project





A relevant title – HelloWorld
(no spaces)

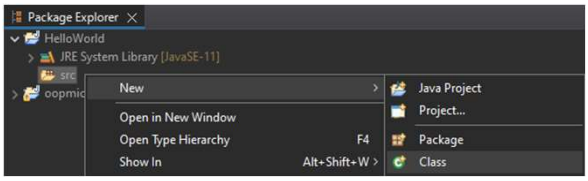
"Don't Create"

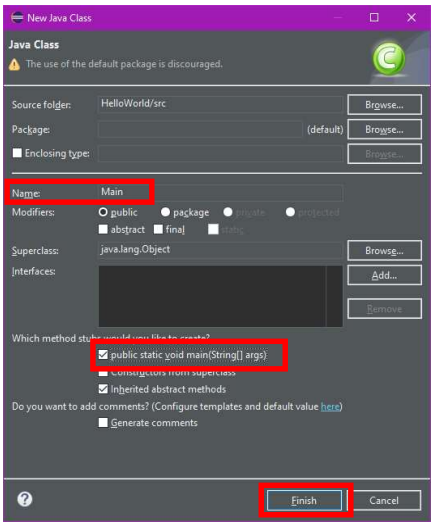


10

Hello World: Setting Up

- Right click the ‘src’ folder – this stands for ‘source’, as in ‘source code’
- Go to New -> Class



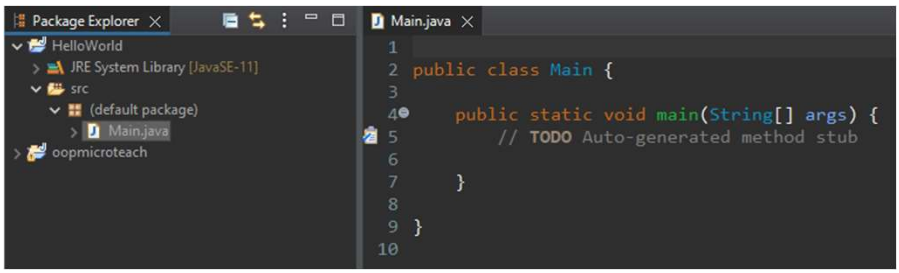


- In the field ‘Name’, call it ‘Main’
- **Ensure you tick the check box** that says:
  - Public static void main(String[] args)
- Click Finish

11

Hello World: Setting Up

- You should now be presented with some code:



- For now, let us ignore the default code. We will explore what this is and why it is important later.

12

Hello World: Setting Up

- The code we need to print “Hello World” to the screen is:
  - System.out.println(“Hello World”);
    - Note, that the ‘S’ in ‘System’ must be a capital!

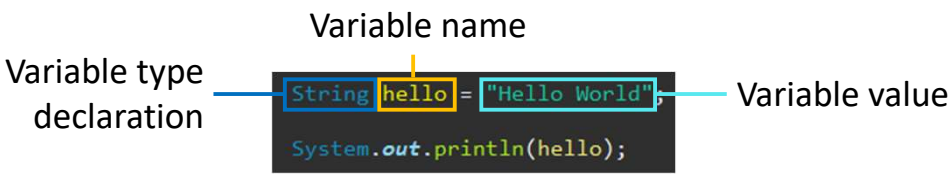
```
public static void main(String[] args) {
    System.out.println("Hello World");
}
```

- What this line of code is doing is accessing the ‘System’ class which is code that exists within the Java language. It then accesses the output stream, and then finally, a method called println (print line).

13

Declaring Variables

- Unlike JavaScript or Python, Java’s variables are **immutable**. This means that the variable **cannot change it’s type!**
- We must declare a variable as the data type that it is going to store, and once chosen, will remain as that type.



14

Executing a Program

- To execute the program, navigate to Run -> Run
  - The hotkey to execute the program is : Control + F11

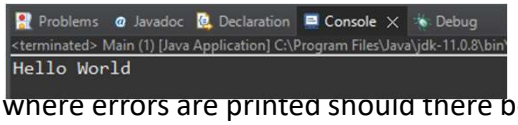
- You can also click the “Run” icon from the main

NB: Make sure you save the file before running!

15

Console

- When we execute our program, the Console will be where we can see our output.



- This will also be the place where errors are printed should there be an issue with the code execution.
- It will also be the location where user input is entered.

16

Task 1: Hello World

1) Create a new project within Eclipse.
2) Create a new class within the 'src' folder called 'Main', ensuring the checkbox for 'public static void main(String[] args)' is checked.
3) Create a String variable with the value "Hello World!".
4) Output to the console the value of the variable.

Stretch Challenge: Create a String variable called 'name' with a name as the value, then print to the console "Hello" + name

17

Declaring Variables

- As with String, we must declare the data type before the name of the variable.

```
int x = 5;
float y = 5.3f;
boolean z = true;
```

- If we want to declare multiple variables of the same data type, we can reduce it to a single line of code:

```
int x = 5, y = 6, z = 50;
```

- If we want to declare multiple variables of the same data type, and with the same value, we can do a multiple instantiation (without a value), and then multiple initialisation (assigning a value):

```
int x, y, z;
x = y = z = 50;
```

18

Mathematical Operators

- Note; that for increment and decrement, the ++ or -- can be before or after the variable (x++ is the same as ++x)

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x

19

If Statements

- In Java, the syntax for an if-statement is **identical** to JavaScript, so this should be a re-familiarisation:

```
int x = 10;

if (x < 5) {
    System.out.println("Values is less than 5");
} else if (x > 5) {
    System.out.println("Values is greater than 5");
} else {
    System.out.println("Value is exactly 5");
}
```

20

Concatenation

- Java uses the + symbol for concatenation as well as addition.
- Should one of the values be a string, string concatenation will occur. Should both values be numeric, addition will occur.
- We can force mathematics to occur before concatenation by using brackets, too!

```
System.out.println("hello " + "world");
System.out.println(10 + 5);
System.out.println(10 + "5");
System.out.println("I am " + (30+1) + " years old");
```

```
hello world
15
105
I am 31 years old
```

21

Task 2: Create a calculator

1) Using if-statements, create a calculator like you did for your home learning task.
2) Use the 4 operators + - / and *

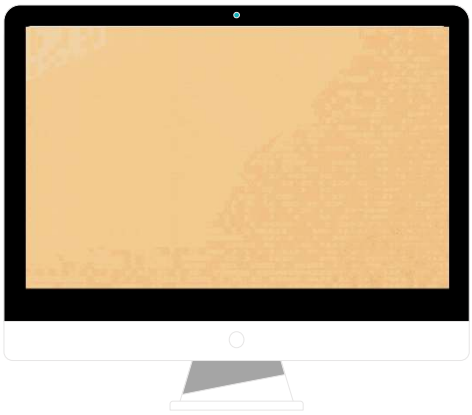
(This task should be relatively simple as you've been able to do it before!)

Stretch Challenge: Include operators for modulo and to-the-power-of

22

Review

- What is the difference between primitive and non-primitive data types and how can we tell the difference when they are declared?



23

Learning Objectives / Key Questions

- What is Java? What is Object-Oriented Programming?
- What are primitive and non-primitive datatypes?
- How do I setup my first class in Eclipse?
- How do I declare variables?
- How do I use mathematical operators?
- What is the syntax for an IF statement?

24