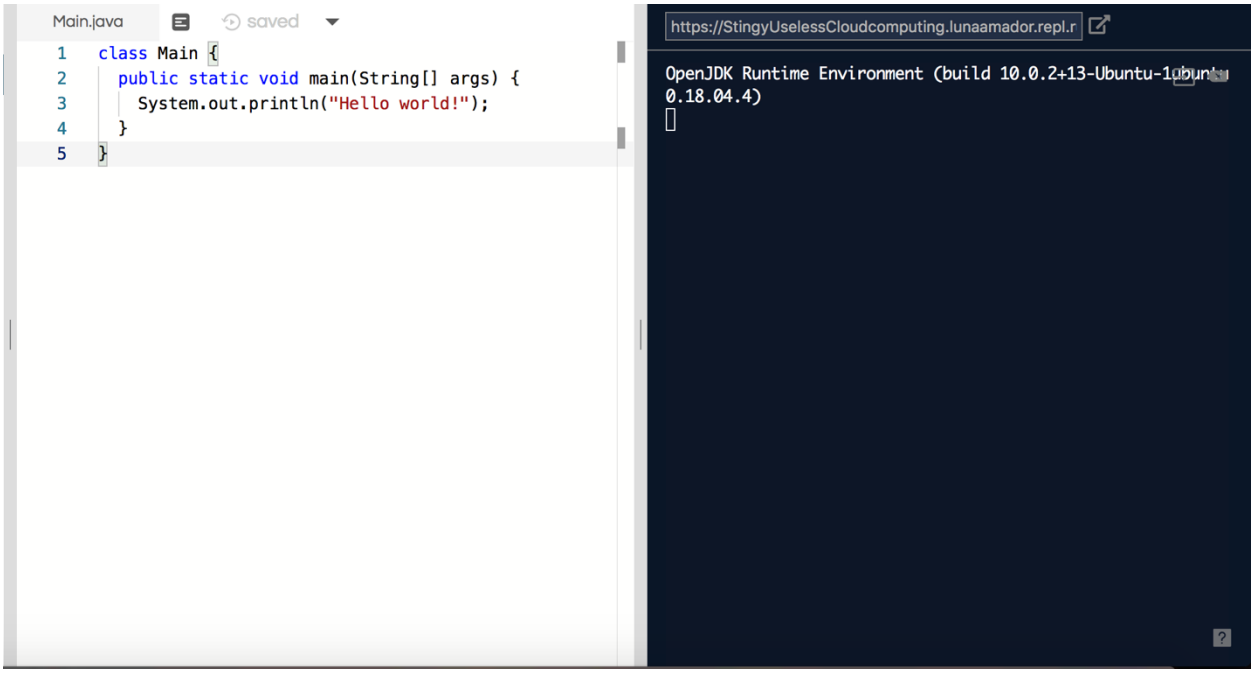

Comment:

- What is it?
  - A comment is a way that programmers make notes to themselves and others about what it is they are trying to do in every line of code they write
- How do I write a comment?
  - First write two forward slashes, and then your comment after it.
    - *For example:* //This line calculates the factorial of the number entered by the user

Print:

- What is it?
  - Printing is a way to show text on the black screen of your computer. When you code, only you, the programmer, can see the white screen of your computer. The person who uses your program can only see the black screen. So if you want to show instructions or the result of your code on the black screen, to show whoever is using the program you made, you have to **print** it.
- How do I print something?
  - The format is: System.out.println();
    - Inside the brackets you can write text in quotation marks (“ ”) or the names of variables with information you want to print
      - *For example:* System.out.println(“Hello world”);
    - If you want to put things together in a print statement, you can do so with a plus (+) sign. This is called **concatenating**
      - *For example:* System.out.println(“Hello” + “ world”);

String:

- What is it?
    - A string is a collection of letters or numbers or symbols written in quotation marks (“ ”)
- How do I make a string?
    - You can make a string by writing anything in quotation marks
        - *For example:* “This is a string”
        - *For example:* “1234”
        - *For example:* “***”
        - *For example:* “*5*6%”

Integer:

- What is it?
    - An integer is any whole number. An integer can be positive or negative and includes 0.
- How do I make an integer?
    - To make an integer, just write the number as you would normally
        - *For example:* 34
        - *For example:* -56
        - *For example:* 1234567890

Boolean:

- What is it?
    - “Boolean” is just a fancy word for “true or false”.
- How do I make a boolean?
    - To make a Boolean just write “true” or “false”
        - *For example:* true
        - *For example:* false

Character:

- What is it?
    - A character is any **single** letter or number.
- How do I make a character?
    - To make a character, write a single letter or number in **single** quotation marks.
        - *For example:* ‘q’
        - *For example:* ‘5’

Variable:

- What is it?
    - A variable is a way to store information for later use. Think of it like a box that you put something in. When you use that variable in your code later, it is like pulling the information out of the box.
- How do I make a variable?
    - To make a variable you need to do four things:

1. First, you must write the **type** of your variable.
    a. **Type:** This is what *kind* of information you are storing. That is: Is it a string, integer, boolean, or character?
        i. *A string is written as:* String
        ii. *An integer is written as:* int
        iii. *A boolean is written as:* boolean
        iv. *A character is written as:* char
2. Second, you must write a space and then the **name** of your variable (or box). Ideally, this is a name that identifies what you are storing in your variable (or box).
3. Third, you must write an **equal sign (=)**
4. Fourth, you write whatever information it is you are storing in your variable.

- *For example:* String name = "Chelsea"
- *For example:* int age = 14
- *For example:* boolean redHair = false
- *For example:* char bloodType = 'A'

Input:

- What is input?
    - Input is when you ask the person who is using your program to enter information and then store it as a **variable** (or box) so that you can use it later. To create an input you first use **print** to tell the user what to do. This could be, "Please enter your name" or "Please enter your first number", etc. Then, you must store this in a **variable** (or box).
    - An important thing to remember about input is that it is automatically stored as a String, so when you create a variable (or box) to store the input, you must make sure that you always first write **String**, then a space and the name of your variable (or box).
- How do I write input?
    - Input is two lines long. The first line is a normal **print** statement, and the second is where you store the information as a variable. However, to store an **input** in a variable we use a special method, slightly different from how we normally create a variable. To create a variable for an input statement, instead of setting the name of the variable equal to something specific, we set it equal to reader.nextLine().
    - *For example:* System.out.println("Please enter your name: ");
      String name = reader.nextLine();
    - *For example:* System.out.println("Please enter the year you were born: ");
      String year = reader.nextLine();

Type Conversion (String to integer):

- What is type conversion?
    - Sometimes we store numbers as strings (for example, "54" instead of 54), but we need to use these numbers to do math. To do math with a number stored as a

string we first need to turn it into an integer. Doing this conversion is called **type conversion**
    - In other words, type conversion turns "54" into 54
- How do I do type conversion?
    - To do type conversion we must follow a specific format:
      int _____ = Integer.parseInt(_____)
      In the first blank space, put a new variable (box) name for your string which has been turned into an integer
      In the second blank space, put the name of the variable (or box) where the original string version of your number is
    - *For example:* int intNumber = Integer.parseInt(stringNumber);

Math:

- Some basic math symbols in Java are:
    - Addition : +
    - Subtraction: -
    - Multiplication: *
    - Division: /

Checking if two strings are equal:

- To check if two **strings** are equal we use the following format:
  _____.equals("____")
  In the first blank space we put the name of the variable (or box) where the first string is
  In the second blank space we put the second string, that we are checking is equal to the first
- This line of code will return a boolean, that is: true or false

Checking if two integers are equal:

- To check if two **integers** are equal we use the following format:
  ____ = = ____
  In the first blank space we put either an integer or the variable (box) name of where the first integer is being stored
  In the second blank space we put another integer or variable (box) name of where the second integer is being stored
- This line of code will return a boolean, that is: true of false

Relationships between integers:

- We can check if two integers are greater than, less than, equal, or not equal to each other.
    - Greater than: >
    - Less than: <
    - Equal to: = =
    - Not equal to: !=

If and else:

- What is if and else?
    - If and else is a way to tell your computer to make a decision if some condition is met. So *if* some condition is met, then your computer will do whatever you tell it to, within the if statement. If the condition is not met, then your computer will skip over what you wrote inside the if statement and instead do whatever you told the computer to do within the *else* statement.
- How do I write if and else?
    - The format for if and else is:

      if (_____) {
      _____;
      }

      else {
      _____;
      }

      In the first blank space, we write a condition, so that if it is true, then the computer will go into the if statement.
      In the second blank space, we write code that the computer will go through if the first blank space is true
      In the third blank space, we write code that the computer will go through if the first blank space is not true

Multiple Conditions (and, or, not):

- Sometimes it is useful to make our if statements more complex. This is when we use *and, or, not.*
    - And: When two things are true
    - Or: When at least one thing is true
    - Not: when the first thing is true and the second is not
- How do I write and, or, not?
    - And: &&
    - Or: ||
    - Not: !

While loop:

- What is a while loop?
    - A while loop is a block of code that keeps repeating over and over again until the condition we write for the while loop is not true anymore
- How do I write a while loop?

- The format for a while loop is:

  while (_____) {
  _____;
  }

  In the first blank space, we write a condition that determines when the loop will stop running
  In the second blank space, we write what we want our code to do if the condition in the while loop is true

For loop:

- What is a for loop?
  - A for loop is similar to a while loop, except we more clearly define when we want the loop to end. The block of code within a for loop will keep running until the condition at the beginning of the for loop is no longer true.
- How do I write a for loop?
  - First we must decide a starting value for our for loop, this is usually int i=0. Then we write a semicolon and write an end point for our for loop, using i. Then we write a semicolon. Finally, we write i++, which tells the computer to add 1 to i every time it runs through the code in the loop once.
  - The format for a for loop is:

    for (i=0 ; i<___; i++) {
    _____;
    }
    In the first blank space, we write the condition at which we wish for our for loop to stop
    In the second blank space, we write the code our for loop will run through if the condition in the first blank space of the loop is met.

Array:

- What is an array?
  - An array is a list of values that can be accessed by index, starting from 0. For example, in an array containing {'h','e','l','l','o'}, we access the letter 'h' using the number 0 and letter 'o' using the number 4
  - However, you cannot change the length of an array after you have created it. You cannot add or take away elements from it.
- How do I make an array?
  - To make an array follow this format:

    ___[] ___= {_,_,_}

In the first blank space, write the type that your array will contain. That is: int, String, char, or boolean
In the second blank space, write the name you want to give your array
In the blank spaces between the curly brackets, write what you are putting in your array

For example: int[] = {2,4,6,8};
For example: char[]={'a', 'b', 'c', 'd'};

- To make a **blank array** (an array that doesn't have values, but rather has a certain number of blank spaces) follow this format:

  ___[] ____= new ___[___];

  In the first and third blank space, write the type of your array (String, int, char, boolean)
  In the second blank space, write the name of your array
  In the fourth blank space, write the number of slots you want your array to have and that you will later fill

ArrayList:

- What is an arrayList?
  - An arrayList is the same as an array except you can add and remove items from them. That is, you can change their length.
- How do I make and arrayList?
  - To make an arrayList, follow this format:

    ArrayList<Character> _____ = new ArrayList<>();

    In the first blank space, write the name you want to give to your arrayList

Function:

- What is a function?
  - A function is a chunk of code that you write in advance and store under a specific name. Then, whenever you want your code to do something that you created a function to do for you, you just have to "call" that function for your program to do it, rather than write out all the code again.
- How do I make a function?
  - To make a function follow this format:

```
public static int _____(_____) {
    _________________;
    return _____;
}
```

In the first blank space, write the name of your function
In the second blank space, enter the type, then a space, and then the name of the variables with the information you will need in your function
In the third blank space, write the code that will allow your function to perform the specific function you want it to
In the fourth blank space, write the name of the variable you want your function to provide as an end result.