# Strings & Packages

Camille Duquesne
1ère IB

Any questions/difficulties you would like to share about last session’s content ?

# B2.1 Programming fundamentals

B2.1.1 Construct and trace programs using a range of global and local variables of various data types.

- Data types: Boolean value, char, decimal, integer, string

B2.1.2 Construct programs that can extract and manipulate substrings.

- Writing of programs that accurately identify and extract substrings from given strings, demonstrating
the ability to perform various manipulations, such as altering, concatenating or replacing

# Generative AI - Traffic Light Guide

🟢Green - For this assignment using Generative AI is authorised and encouraged to help you learn and discover new ideas. Prompts must always be cited in the following way: “Text of prompt” prompt. *ChatGPT*, Day Month version, OpenAI, Day Month Year, chat.openai.com. You should also share the discussion you had with the generative AI if you used many prompts.

🟠Orange - For this assignment using Generative AI is not recommended as it will not make you practice valuable research and thinking skills. If you are finding the assignment difficult try to use online searches instead, come to office hours, send emails to your teacher or ask your peers.

🔴Red - For this assignment using Generative AI is not allowed and goes against academic integrity rules. If Generative AI is detected you are exposing yourself to academic sanctions.

# What is a Java package ?

A package is a set of files with reusable code.

Packages can either be built-in (prewritten) or user-defined. We will focus on the built-in packages.

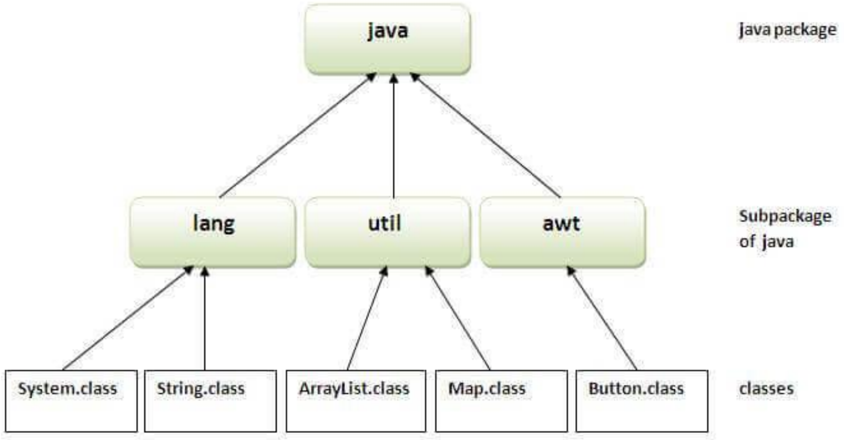


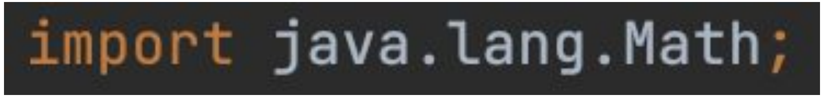
*Image from: https://www.javatpoint.com/package*

Documentation of all java packages can be found here:
here:https://docs.oracle.com/en/java/javase/17/docs/api/java.base/module-summary.html#packages-summary

Let’s focus on the Math package for now namely:
https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Math.html

# How to import a package ?

## Importing a specific class:

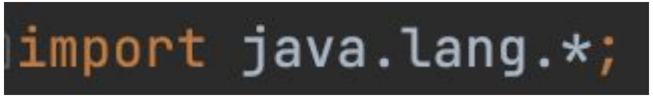


```java
import java.lang.Math;
```

Import keyword

Package path

## Importing the whole package:

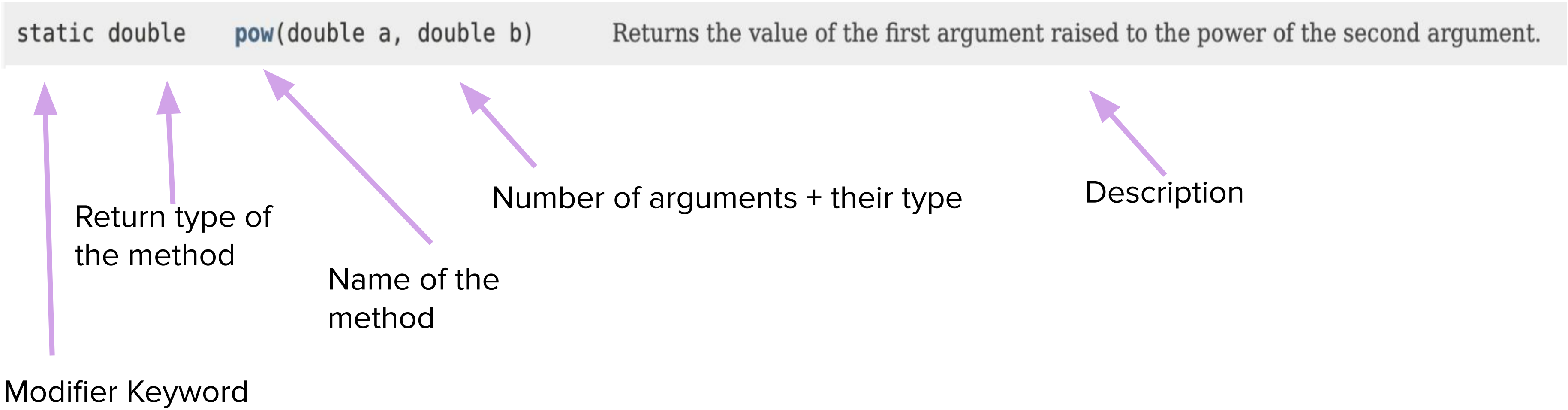


```java
import java.lang.*;
```

In most programming languages the `*` means “everything”.

# What methods are available in our package ?

Let’s read the documentation to see if there is a method for computing powers.



```
static double    pow(double a, double b)
```

Returns the value of the first argument raised to the power of the second argument.

Modifier Keyword

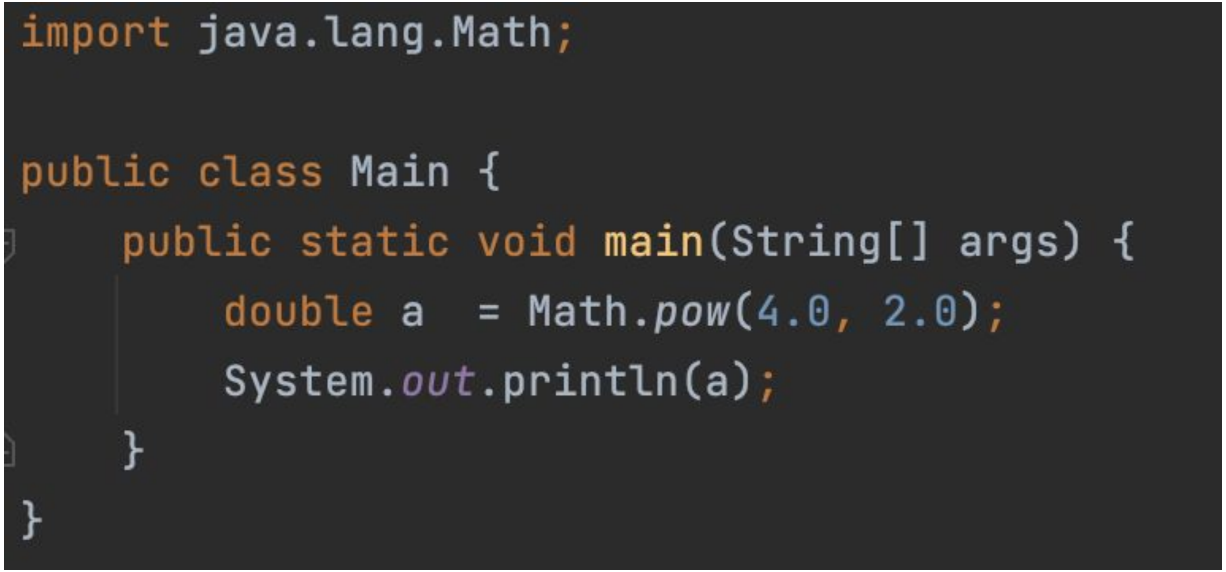
Return type of the method

Name of the method

Number of arguments + their type

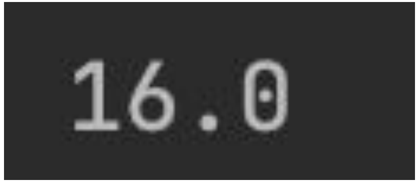
Description

# How do we use our pow method ?



```java
import java.lang.Math;

public class Main {
    public static void main(String[] args) {
        double a  = Math.pow(4.0, 2.0);
        System.out.println(a);
    }
}
```



```
16.0
```

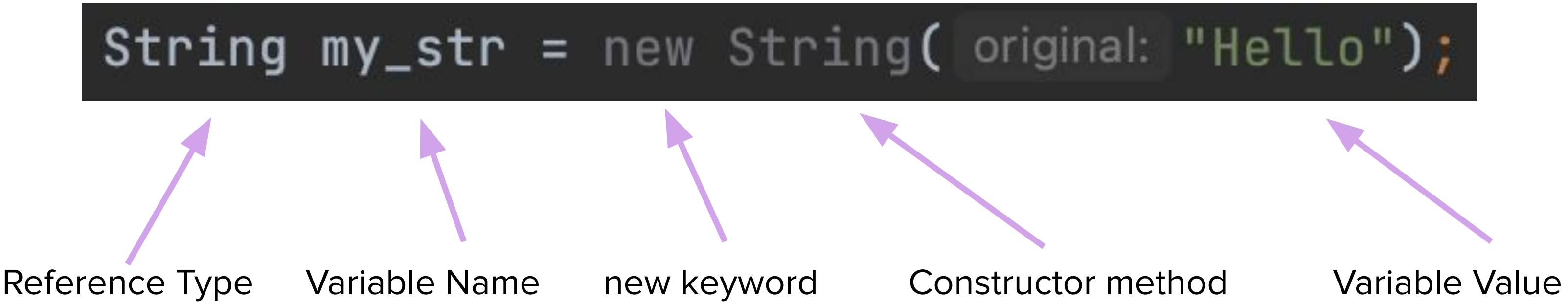
# Reference types

**Primitive types** are internally predefined by java and don’t have related methods.

On the contrary **Reference types** are not predefined internally and are Objects that need to be constructed from a Class. Reference types can have related methods (method means function in java).

Some reference types are built-in by java, such as `String` or `Array`, but they can also be user defined.

# Creating an instance of a reference type/a class



```
String my_str = new String( original: "Hello");
```

Reference Type

Variable Name

new keyword

Constructor method

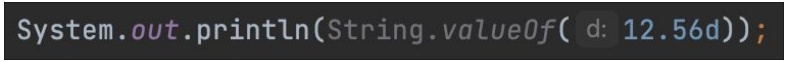
Variable Value

# String methods

If we look at the documentation of the string types we see two categories of methods: Static methods and instance methods.

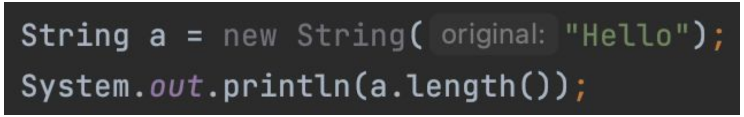
https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html

**Static methods** are accessible *without* an instance of a class



```
System.out.println(String.valueOf( d: 12.56d));
```

**Instance methods** are *only* accessible through an instance of a class

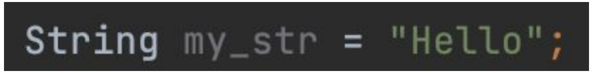


```
String a = new String( original: "Hello");
System.out.println(a.length());
```

# String literal vs String Constructor

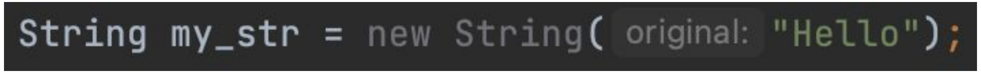
There are two ways to create a String in Java:

**With a string literal directly:**



```
String my_str = "Hello";
```

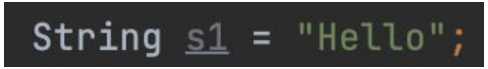
**With the constructor of the String class:**



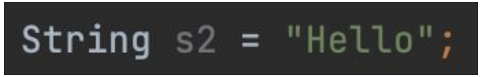
```
String my_str = new String( original: "Hello");
```

Preferred way
(but only for
Strings)

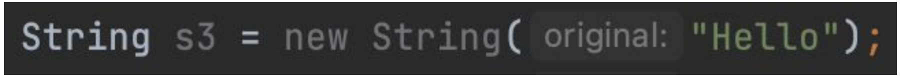
# Strings and memory reference



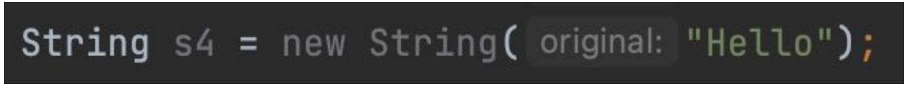
```
String s1 = "Hello";
```



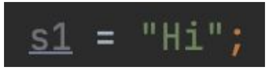
```
String s2 = "Hello";
```



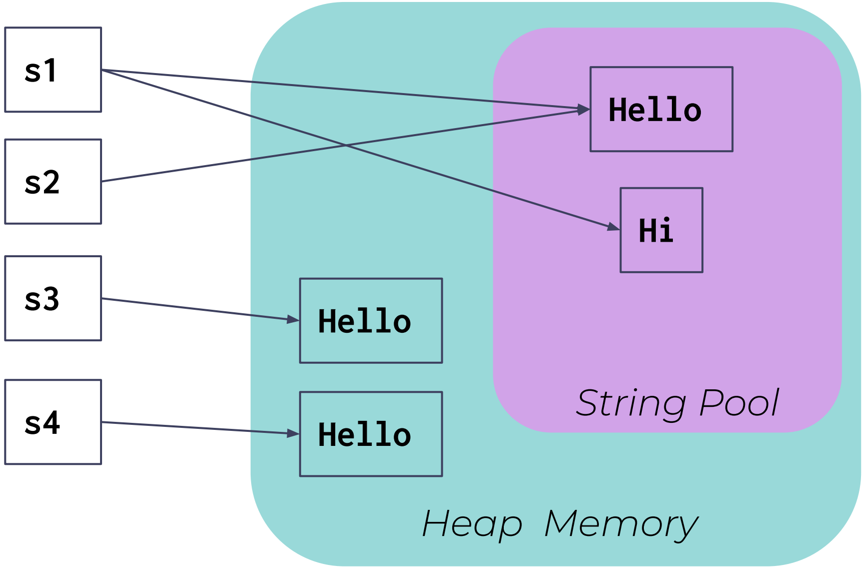
```
String s3 = new String( original: "Hello");
```



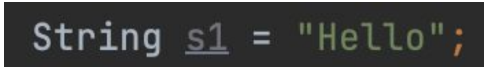
```
String s4 = new String( original: "Hello");
```



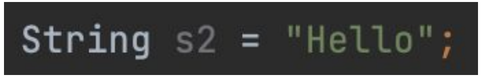
```
s1 = "Hi";
```



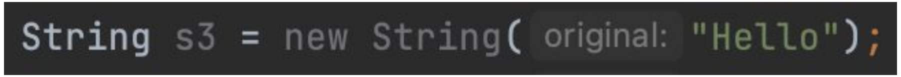
# Strings and memory reference



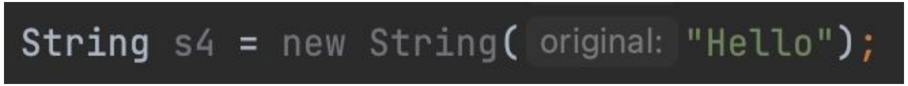
```java
String s1 = "Hello";
```



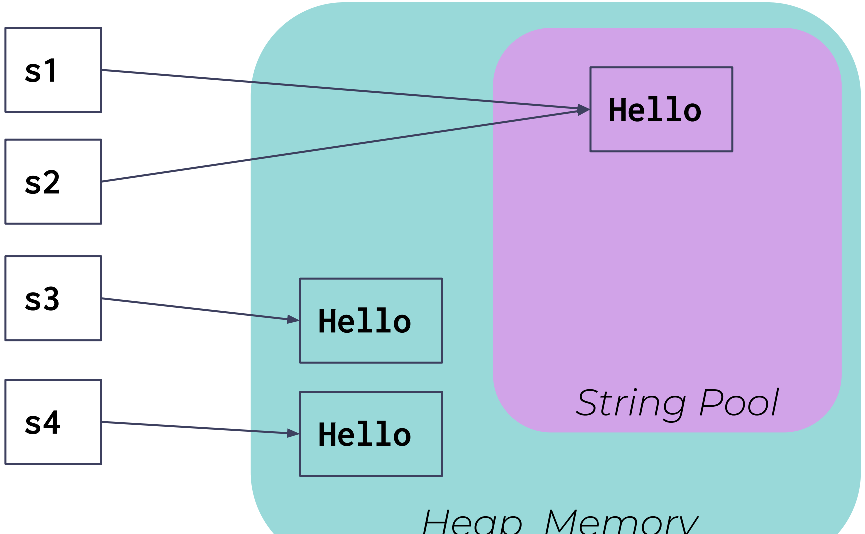
```java
String s2 = "Hello";
```

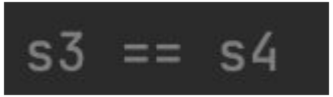


```java
String s3 = new String( original: "Hello");
```



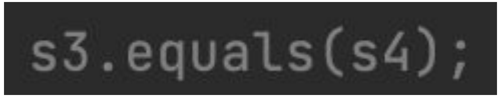
```java
String s4 = new String( original: "Hello");
```





```java
s3 == s4
```

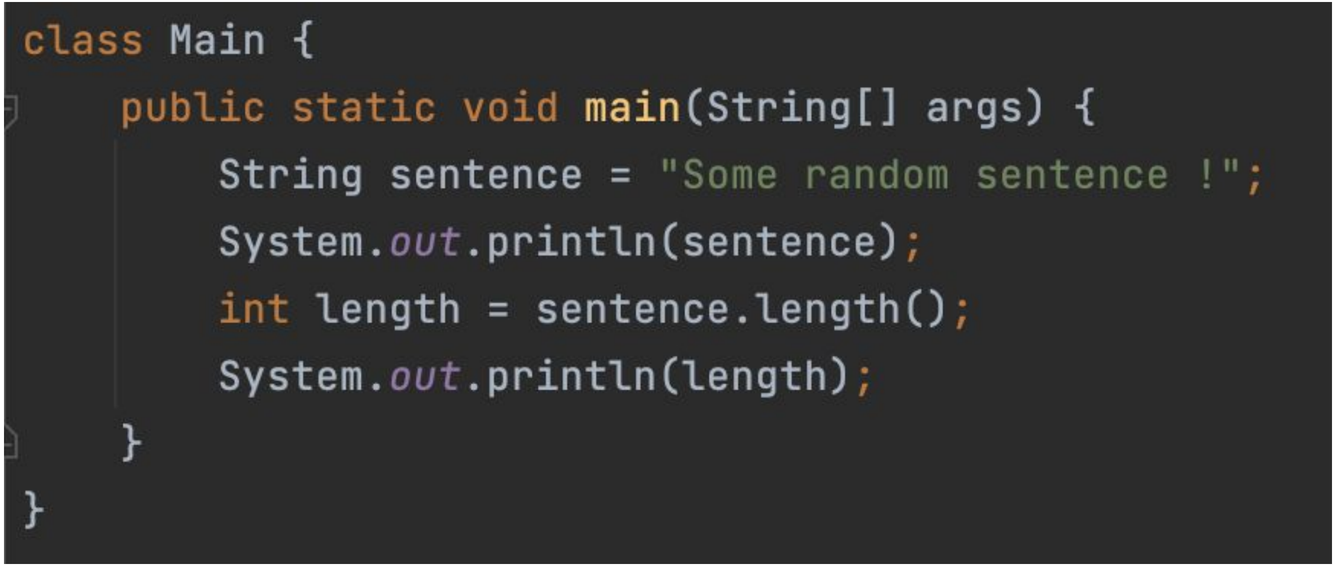
```
false
```



```java
s3.equals(s4);
```

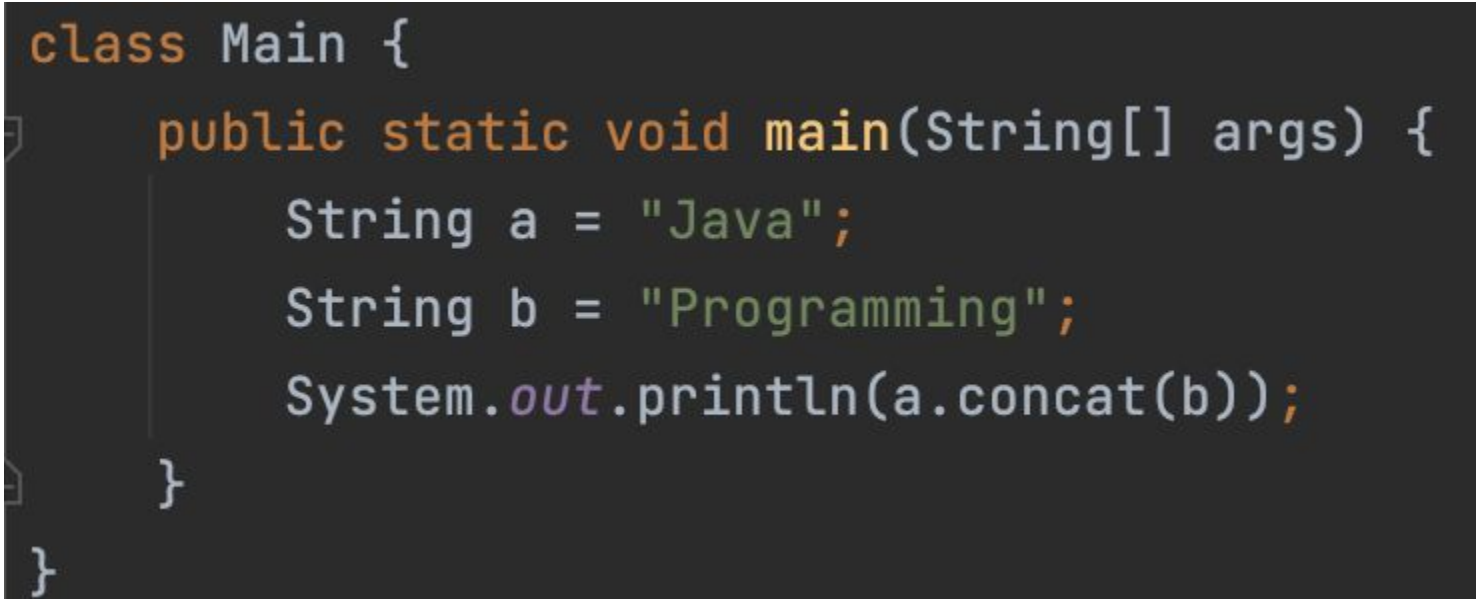
```
true
```

# What does this code output ?



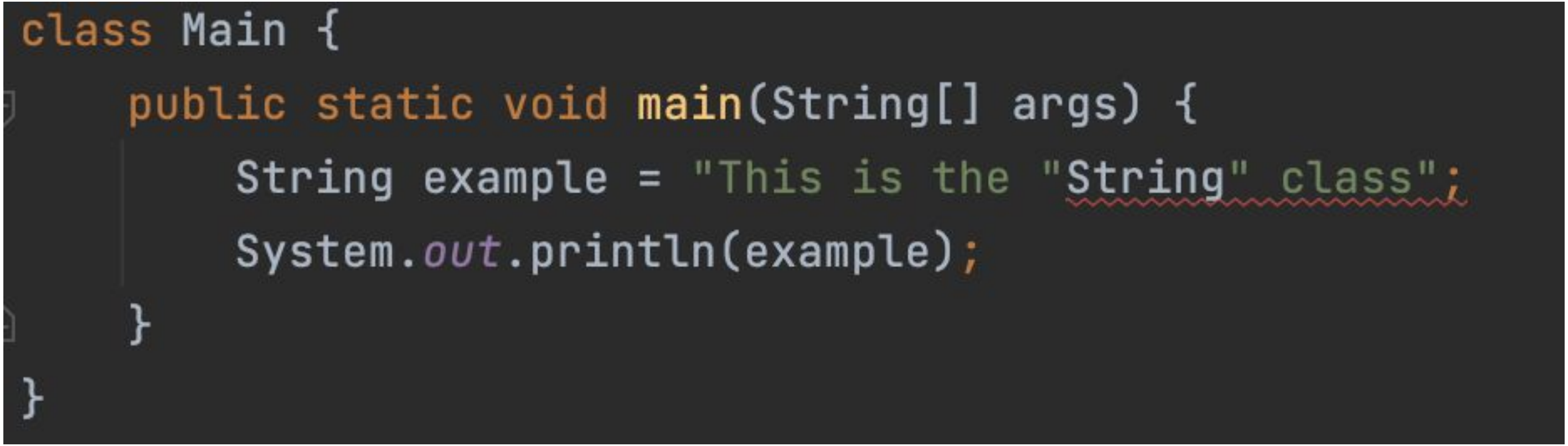
```
class Main {
    public static void main(String[] args) {
        String sentence = "Some random sentence !";
        System.out.println(sentence);
        int length = sentence.length();
        System.out.println(length);
    }
}
```

# What does this code output ?



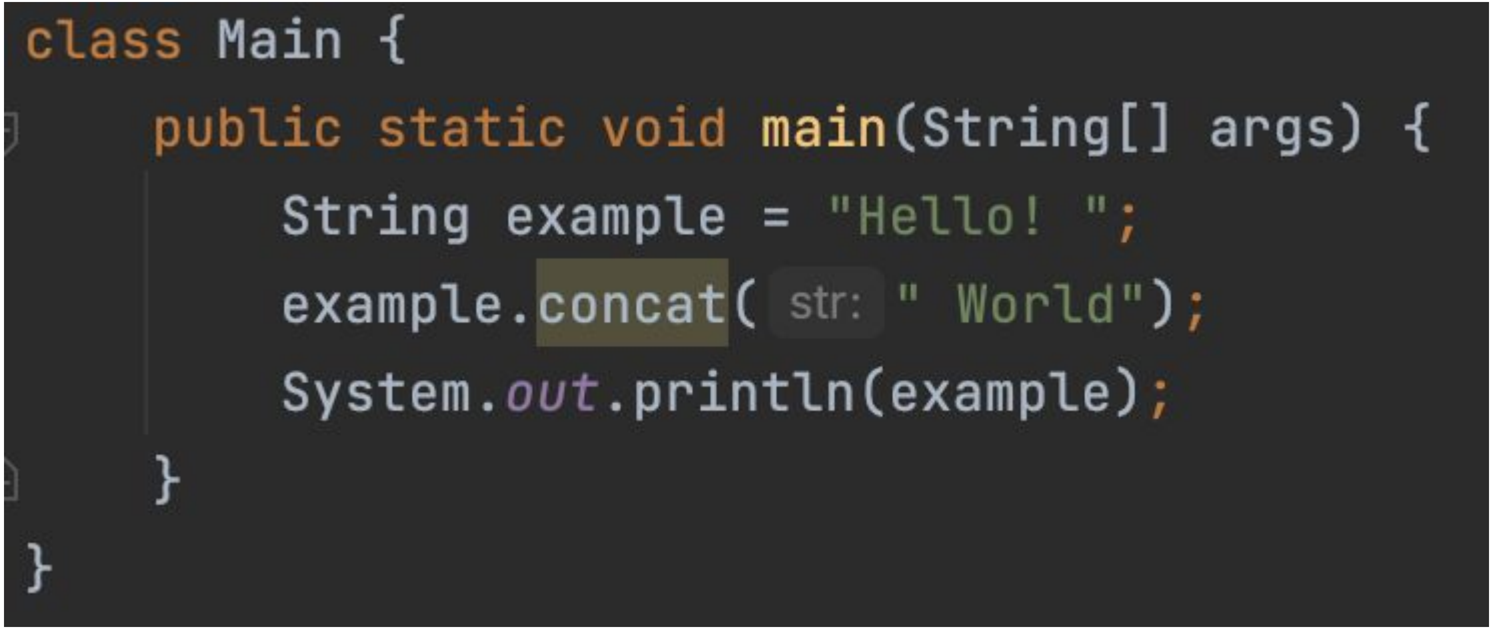
```
class Main {
    public static void main(String[] args) {
        String a = "Java";
        String b = "Programming";
        System.out.println(a.concat(b));
    }
}
```

# What does this code output ?



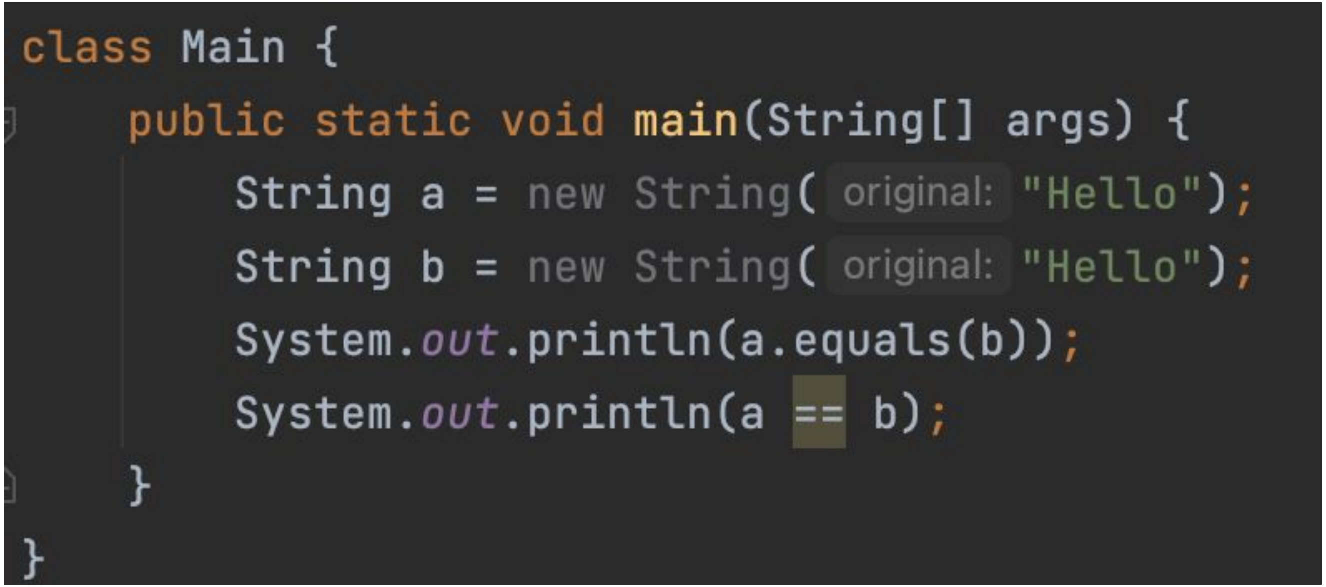
```java
class Main {
    public static void main(String[] args) {
        String example = "This is the "String" class";
        System.out.println(example);
    }
}
```

# What does this code output ?



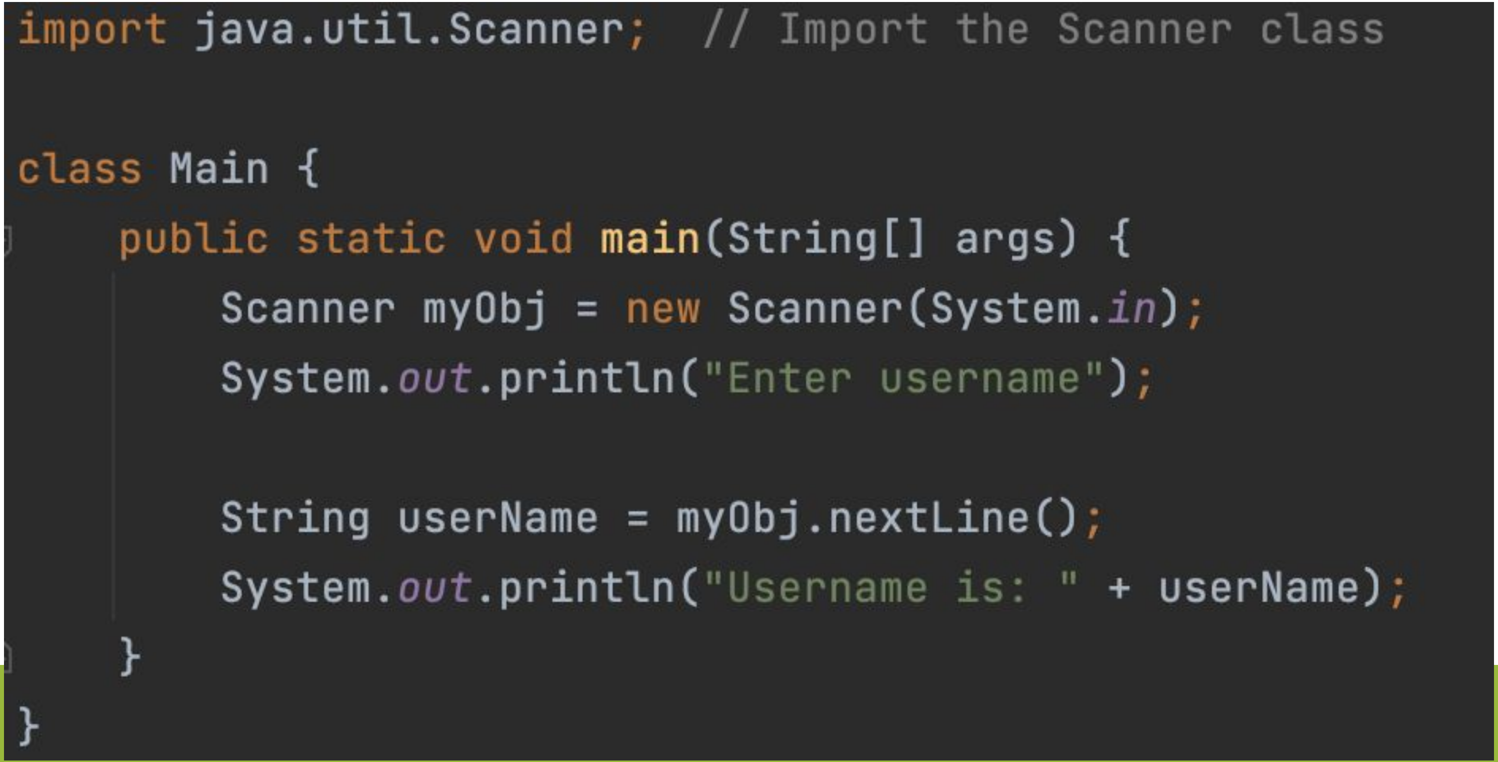
```
class Main {
    public static void main(String[] args) {
        String example = "Hello! ";
        example.concat( str: " World");
        System.out.println(example);
    }
}
```

# What does this code output ?



```
class Main {
    public static void main(String[] args) {
        String a = new String( original: "Hello");
        String b = new String( original: "Hello");
        System.out.println(a.equals(b));
        System.out.println(a == b);
    }
}
```

# What does this code output ?

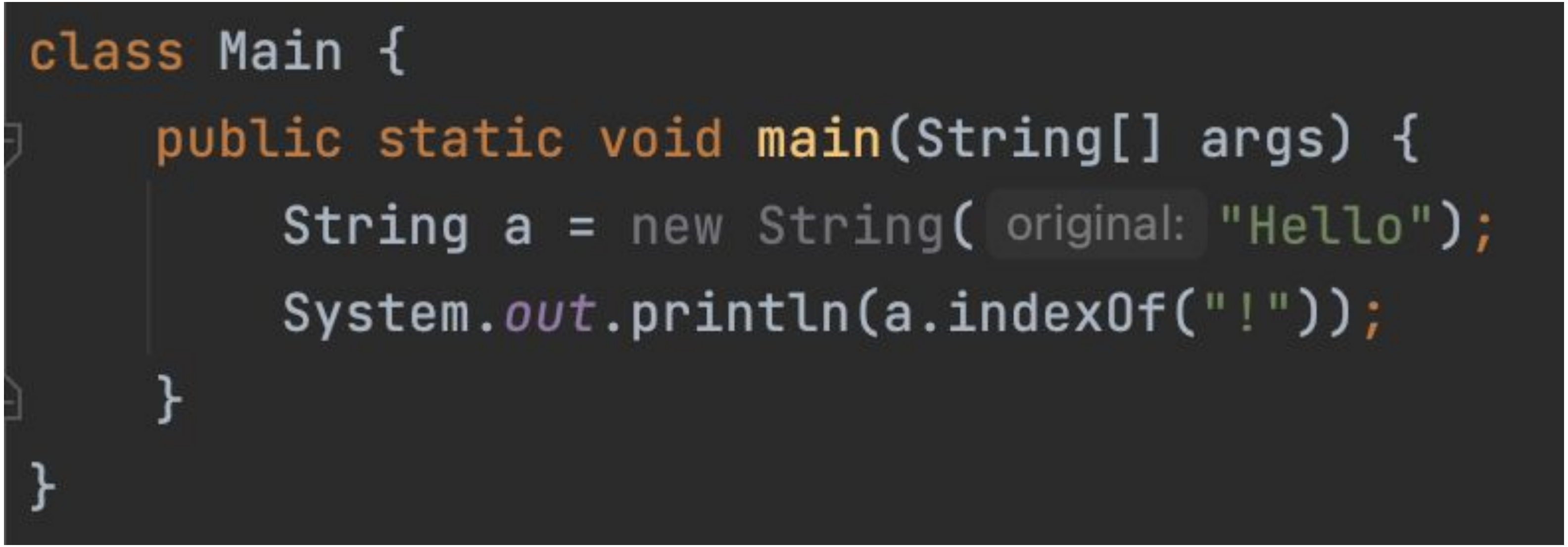


```
import java.util.Scanner;  // Import the Scanner class

class Main {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        System.out.println("Enter username");

        String userName = myObj.nextLine();
        System.out.println("Username is: " + userName);
    }
}
```

# What does this code output ?



```
class Main {
    public static void main(String[] args) {
        String a = new String( original: "Hello");
        System.out.println(a.indexOf("!"));
    }
}
```

# Exercise 1 *(GenAI 🟠 Orange)*

Create a program that outputs the current date

# Exercise 2 *(GenAI 🟠 Orange)*

Create a program that takes a String written through user input and outputs the String in all uppercase letters

# Exercise 3 *(GenAI 🟠 Orange)*

Create a program that takes a String written through user input and replaces all the vowels by an exclamation point.

# Exercise 4 *(GenAI 🟠 Orange)*

Create a program that takes a String written through user input and returns a boolean that checks if the String Starts with “Java”.

# Exercise 5 *(GenAI 🟠 Orange)*

Create a program that takes one number through user input and that outputs the square root of that number.

# Exercise 6 *(GenAI 🟠 Orange)*

Create a program that takes 2 strings written successively written through user input and outputs a boolean that checks if the 2 Strings are the same.

# Exercise 7 *(GenAI 🟠 Orange)*

Create a program that stores the date of your birthday (this year 2023) in a variable and then returns true if your birthday is yet to come and false if your birthday has already passed.

# Exercise 8 *(GenAI 🟠 Orange)*

Define a string in your program and then ask the user to input a character, display a boolean that is True if the character is in the String and False otherwise.

# Exercise 9 *(GenAI 🟠Orange/🟢Green)*

Write a Java program to get the whole and fractional parts from a double value

# Exercise 10 *(GenAI 🟠 Orange)*

Create a table that summarizes the following string methods in java. For each method search for the input(s), return type and provide an example.

.charAt()

.contains()

.indexOf()

.substring()

.replace()

.replaceFirst()

.replaceAll()

.concat()

https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html

# Pause & Recall

Close your eyes and try to recall as many things as possible that were covered during this lesson.

Created by Victoruler
from Noun Project

Alternatively, you can keep your eyes open and write down as many things you remember on a piece of paper.

This will help strengthen your memory of key concepts 💪

# Homework

Finish all the exercises

Create flashcards for the following terms:

String
Package
Reference type
instance
Static methods
Instance Methods
Concatenation
.charAt()
.contains()
.indexOf()
.substring()
.replace()
.replaceFirst()
.replaceAll()
.concat()