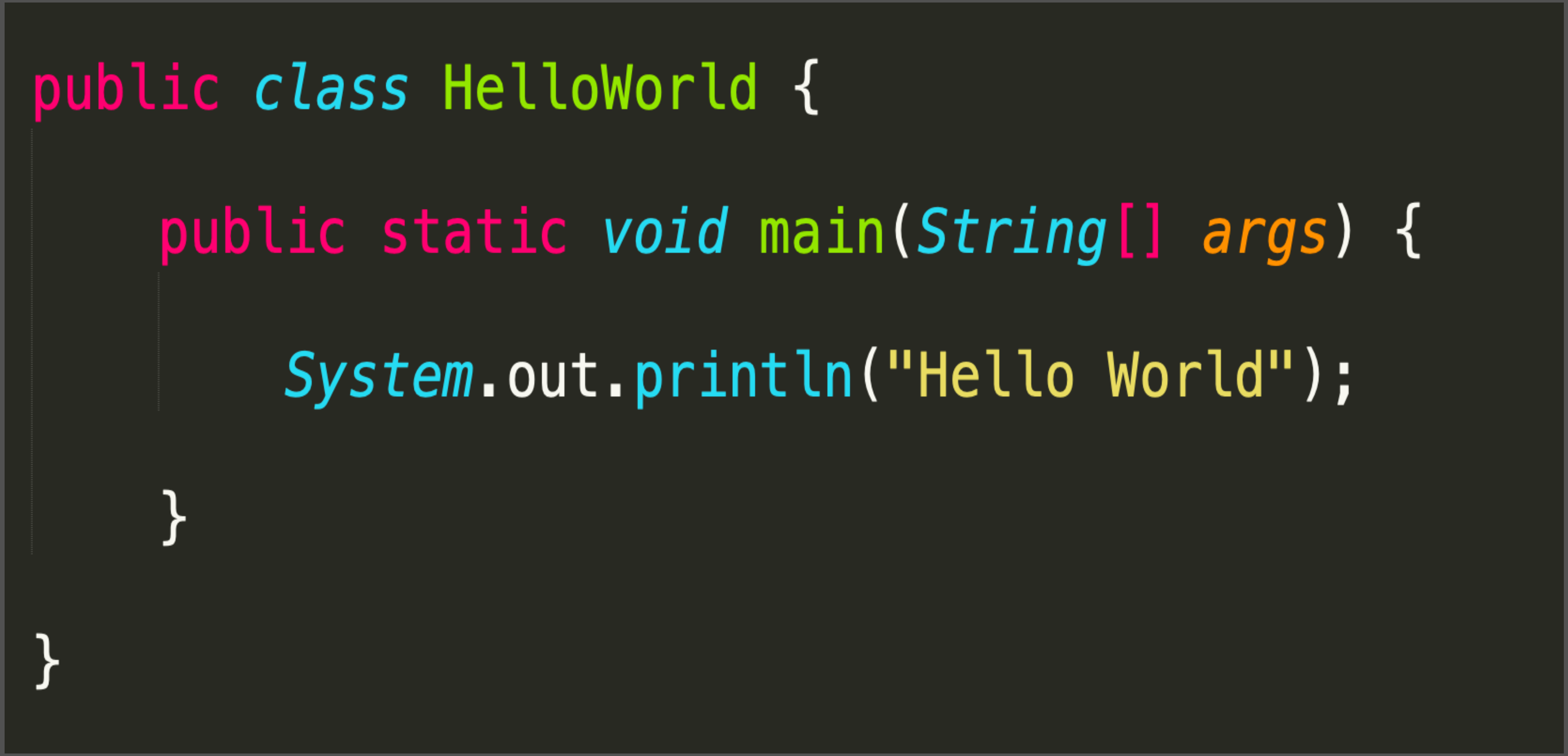
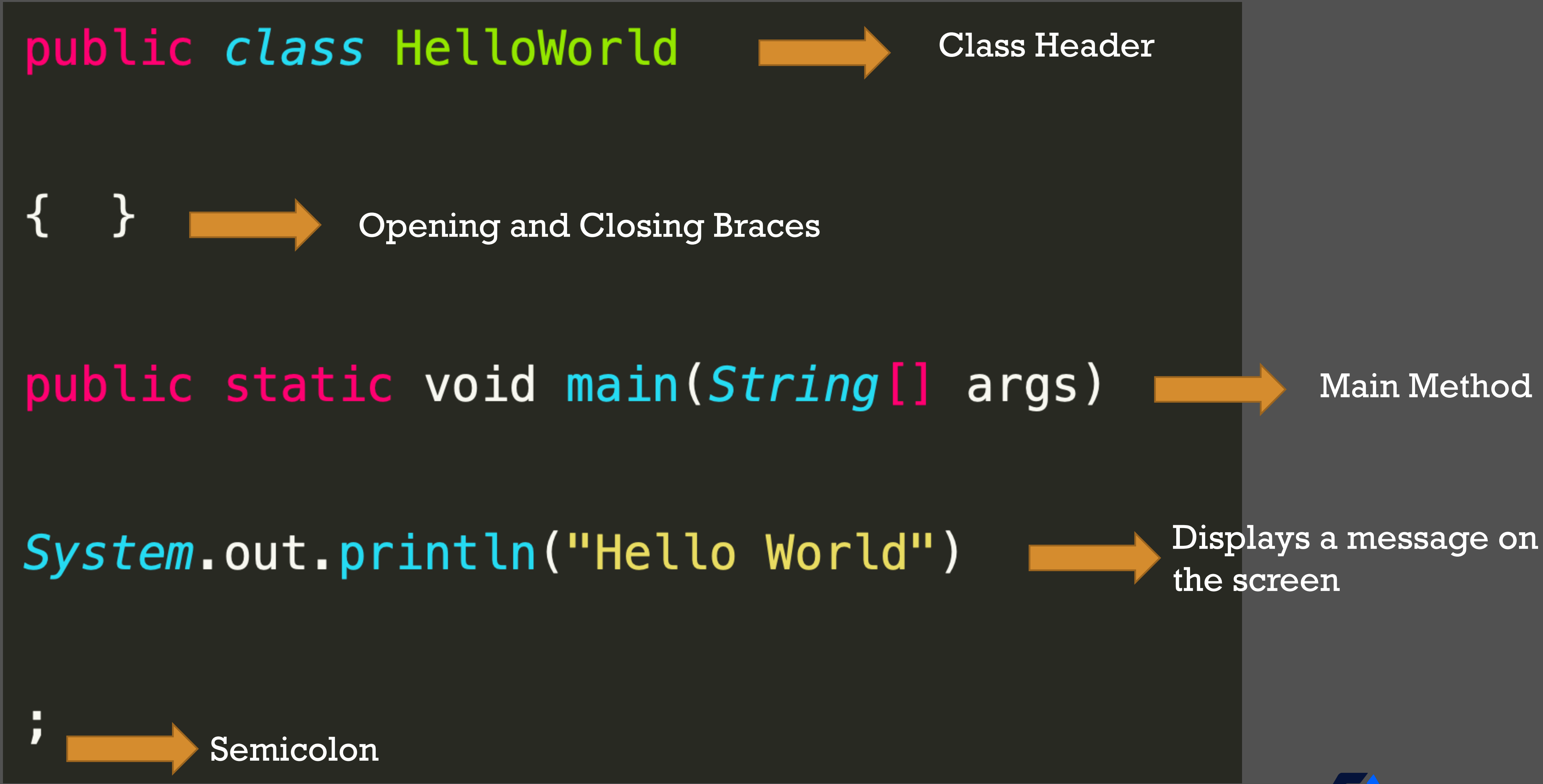
Hello World

Cybertek

Printing to Console



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



```
public class HelloWorld
```

Class Header

```
{  }
```

Opening and Closing Braces

```
public static void main(String[] args)
```

Main Method

```
System.out.println("Hello World")
```

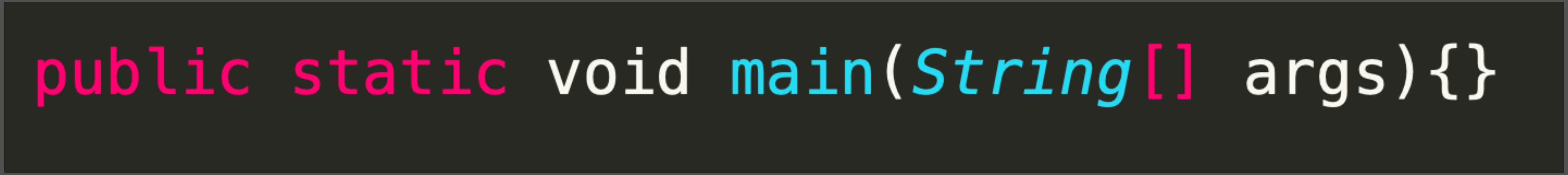
Displays a message on the screen

```
;
```

Semicolon

main( ) method

- Every Java application must contain a main method whose signature looks like this:



```
public static void main(String[] args){}
```

- When the Java interpreter executes an application (by being invoked upon the application's controlling class), it starts by calling the class's main method. The main method then calls all the other methods required to run your application.

REMEMBER

JAVA

IS

VERY, VERY

CASE SENSITIVE

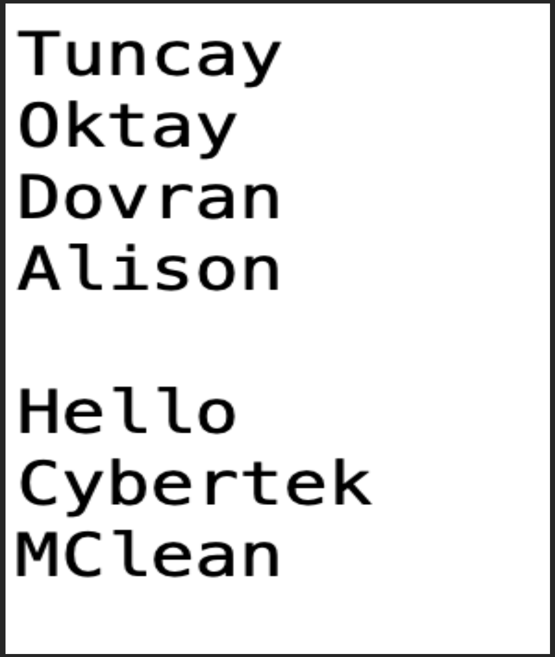
$$A \neq a$$

Task

- Create a new class called classmates.java
- Print your name in the console

Task

- Create a new class called PrintExercise.java
- Write a code to print following output in the console

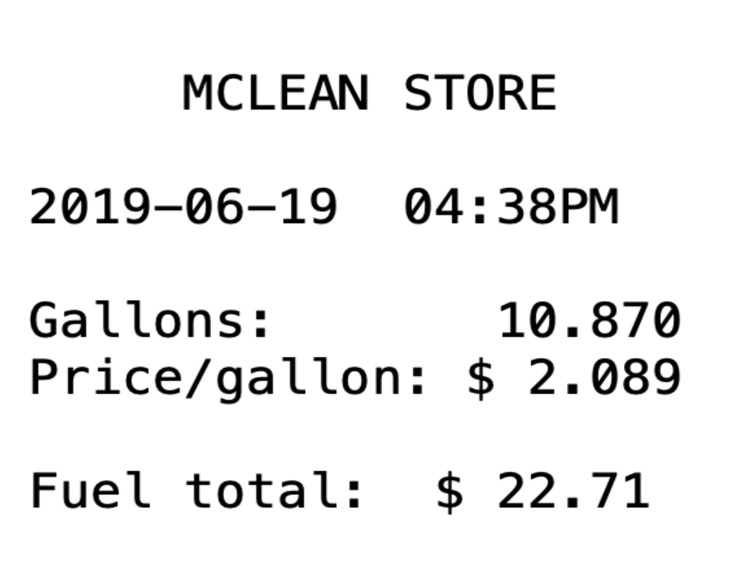


```
Tuncay
Oktay
Dovran
Alison

Hello
Cybertek
MClean
```

Cybertek

Task

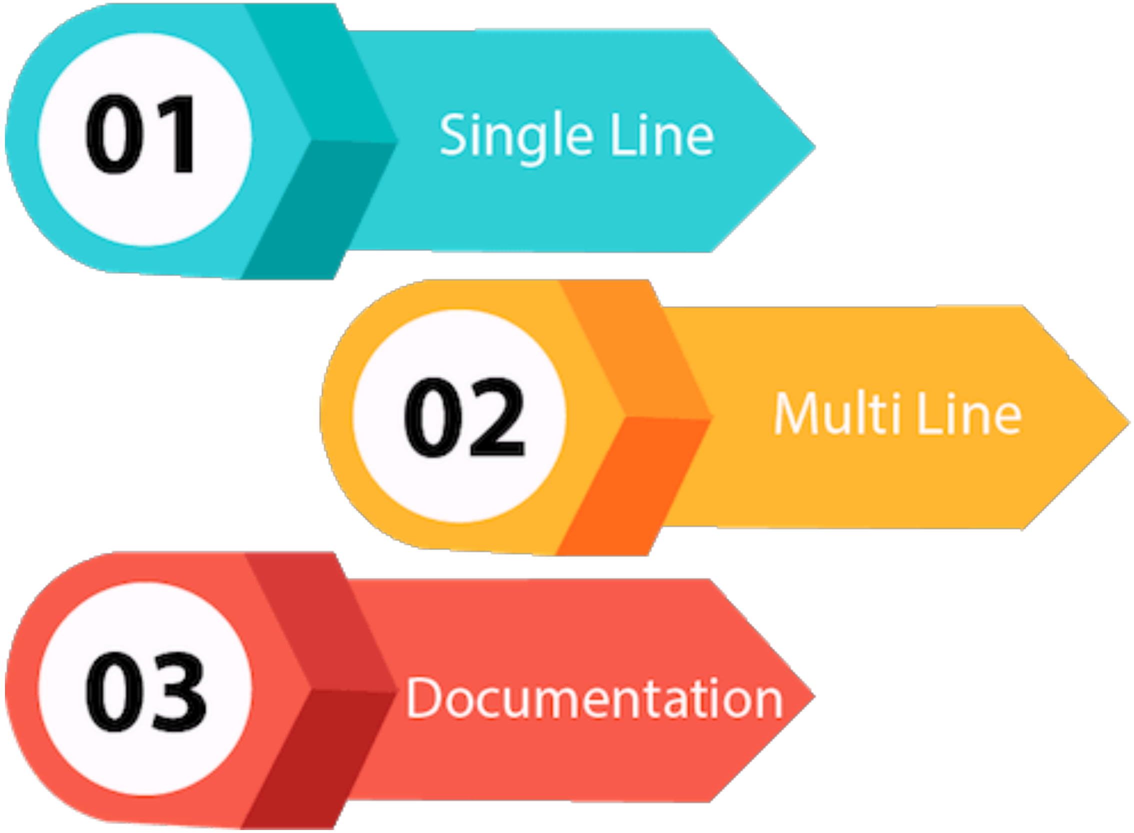
- Create a new class called GasReceipt.java
- Write a code to print following output in the console



```
     MCLEAN STORE

2019-06-19  04:38PM

Gallons:       10.870
Price/gallon: $ 2.089

Fuel total:  $ 22.71
```

Comments

Java comments can be used to make the code more readable.



01 Single Line

02 Multi Line

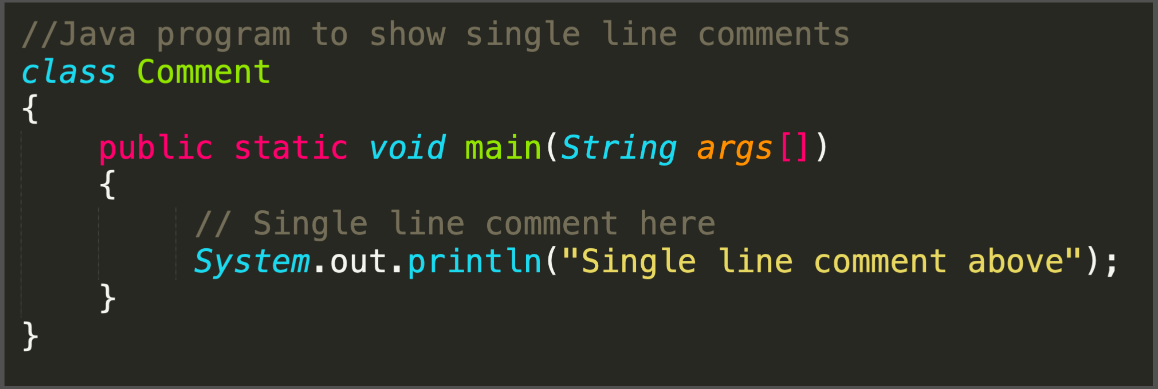
03 Documentation

Single-Line Comments

In a single-line comments, anything that follows the two forward slash characters // on that line will not be processed by the JavaScript interpreter. Single line comments are often used for short descriptions of what the code is doing.

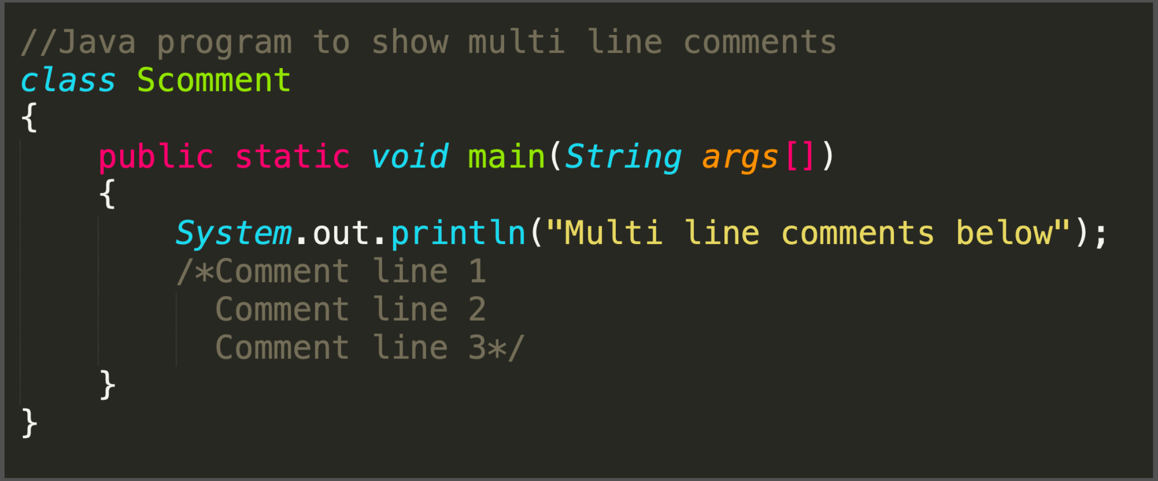
Multi-Line Comments

To write a comment that stretches over more than one line, you use a multi-line comment, starting with the /* characters and ending with the */ characters. Multi line comments are often used for descriptions of how the script works, or to prevent a section of the script from running when testing it.



```java
//Java program to show single line comments
class Comment
{
    public static void main(String args[])
    {
        // Single line comment here
        System.out.println("Single line comment above");
    }
}
```

Single Line Comment



```java
//Java program to show multi line comments
class Scomment
{
    public static void main(String args[])
    {
        System.out.println("Multi line comments below");
        /*Comment line 1
         Comment line 2
         Comment line 3*/
    }
}
```

Multi Line Comment

Cybertek

println() vs print()

- System.out.println() : Appends a newline at the end of the data output
- System.out.print() : Does not append newline at the end of the data output

Example	Output

System.out.print(“one”);
System.out.print(“two”);
System.out.println(“three”);
	onetwothree

System.out.println(“one”);
System.out.println(“two”);
System.out.println(“three”);
	one
	two
	three

System.out.println();	[new line]

Common Escape Sequences

Escape Sequence	Name	Description
\n	Newline	Advances the cursor to the next line for subsequent printing
\t	Horizontal Tab	Causes the cursor to skip over to the next tab stop
\\	Backslash	Causes a backslash to be printed
\’	Single quote	Causes a single quotation mark to be printed
\"	Double quote	Causes a double quotation mark to be printed