# Object-oriented Programming

Lesson 1.1

# Learning Outcomes

LO 1.1.1 **Explain** the difference between object-oriented and procedural programming paradigms

LO 1.1.2 **Define** valid OOP concepts and terminologies

LO 1.1.3 **Compile** and **run** successfully a program using Java programming language

# What is OOP?

**Object-oriented programming** is a software development paradigm that focuses on the development of self-contained software components, called **objects** which are modeled after things that appear in the real world.

# OOP vs. Procedural Programming

Object-oriented programming focuses on

**object design and interaction**

while procedural programming focuses on

**processing different types of data sequentially**.

# What is an Object?

*"An object is an **abstraction** of something in a **problem domain**, reflecting the capabilities of the system to keep information about it, interact with it, or both."* — Coad & Yourdon (1990)

# What is an Object?

*“We define an object as a concept, abstraction, or thing with crisp boundaries and meaning for the problem at hand. Objects serve two purposes: they **promote understanding of the real world** and **provide a practical basis for computer implementation**.”* — Rumbaugh et al. (1991)

# What is an Object?

*“Objects have **state**, **behavior** and **identity**.” — Booch (1994)*

# Object

1. **Identity** (***Who am I?***)
   Each object is unique

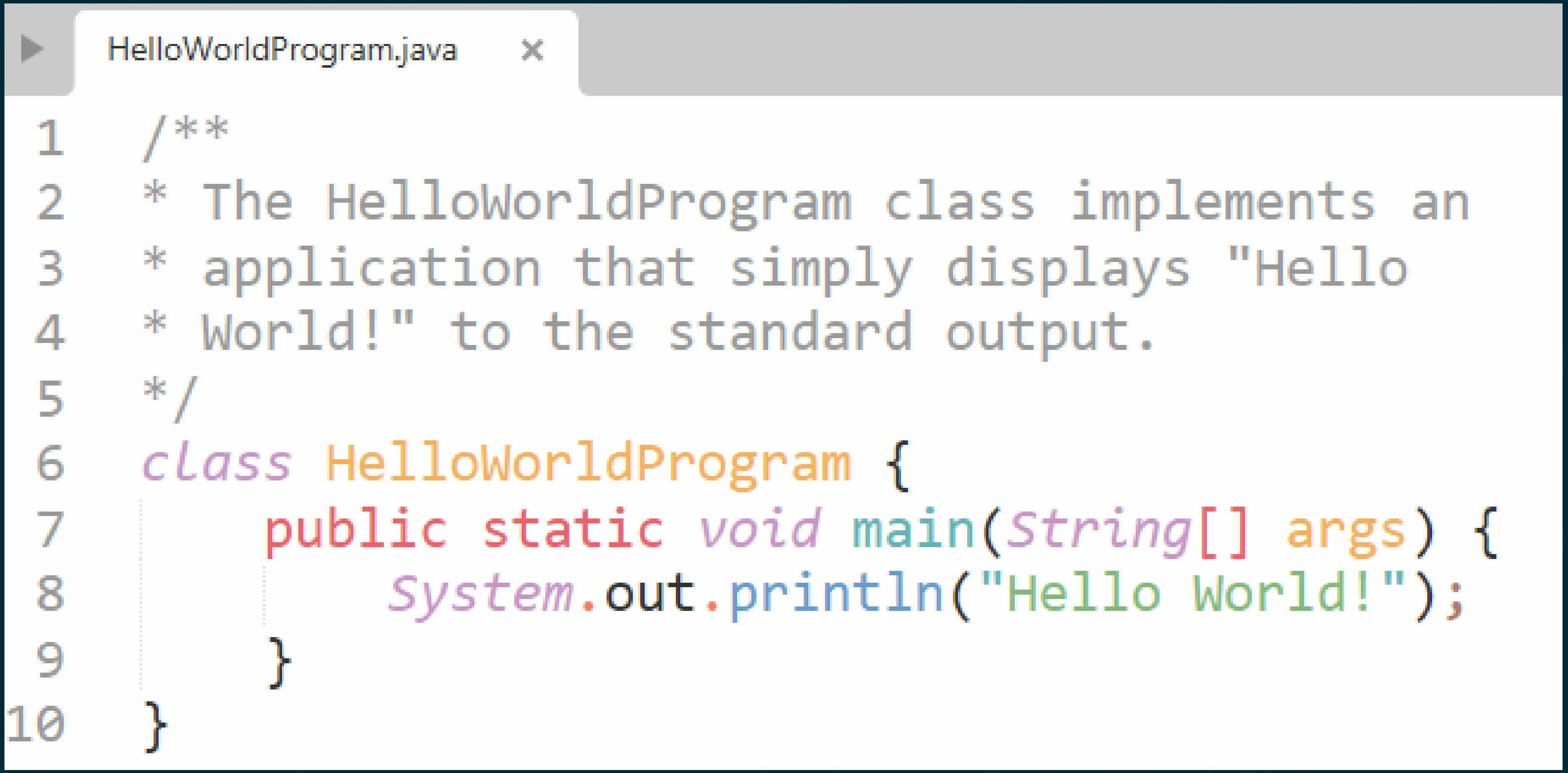
2. **State** (***What do I know?***)
   The conditions of an object at any moment that affect how it behaves

3. **Behavior** (***What can I do?***)
   The way in which an object responds to messages

# Object

| Object | Identity | State | Behaviour |
|---|---|---|---|
| A person | 'Catriona Gray' | — Miss Universe<br>— Sexy<br>— Beautiful | — Speak<br>— Walk<br>— Smile |
| A shirt | 'My favorite shirt' | — Worn<br>— Crumpled<br>— Faded | — Shrink<br>— Rip<br>— Decolorize |
| A bottle of ketchup | '*This* bottle of ketchup' | — Opened<br>— Empty | — Squirt |

# Java Programming Language



```java
/**
* The HelloWorldProgram class implements an
* application that simply displays "Hello
* World!" to the standard output.
*/
class HelloWorldProgram {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

# Java Programming Language

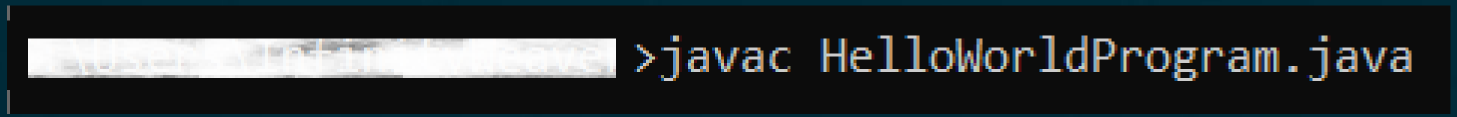
1. **Write valid code then save it as .java file extension**
    a. The class name must be the same as the filename
    b. The parameter of the main program is always String[]

# Java Programming Language

## 2. Compile written class file (.java)

a. Open the cmd prompt and change the directory to the location of the Java workspace you are trying to compile

b. Compile the .java file using *javac* command in the cmd prompt



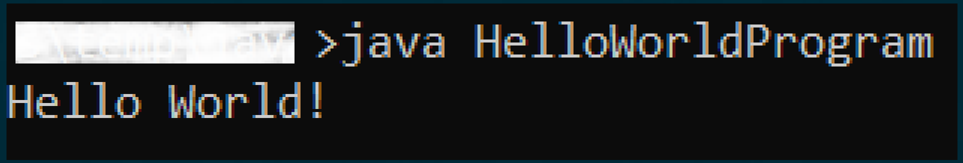
```
[redacted]>javac HelloWorldProgram.java
```

c. If errors have been found, read and fix the errors then compile again

# Java Programming Language

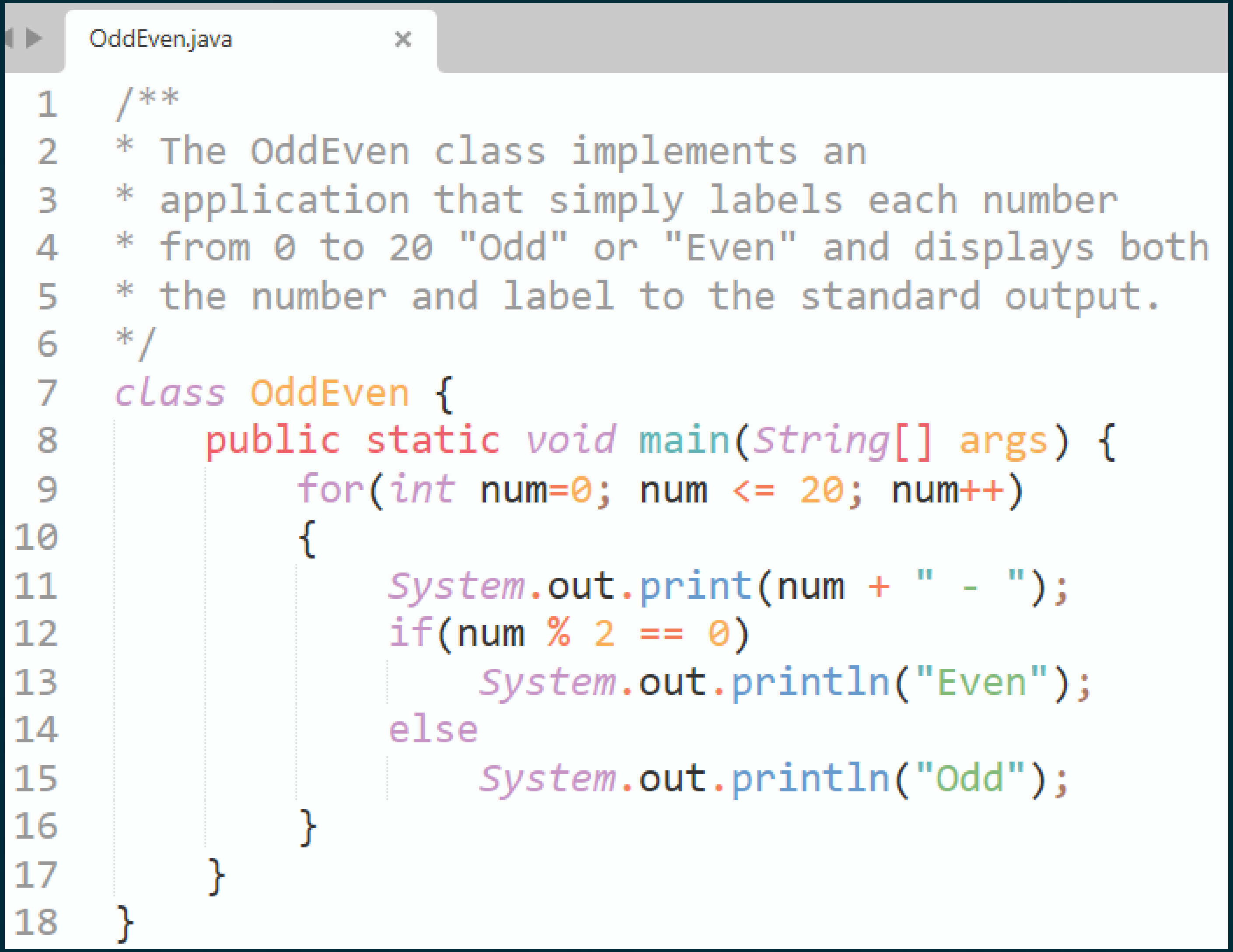
## 3. Run compiled program (.o file generated after successful compilation)

Run the compiled program using *java* command in the cmd prompt



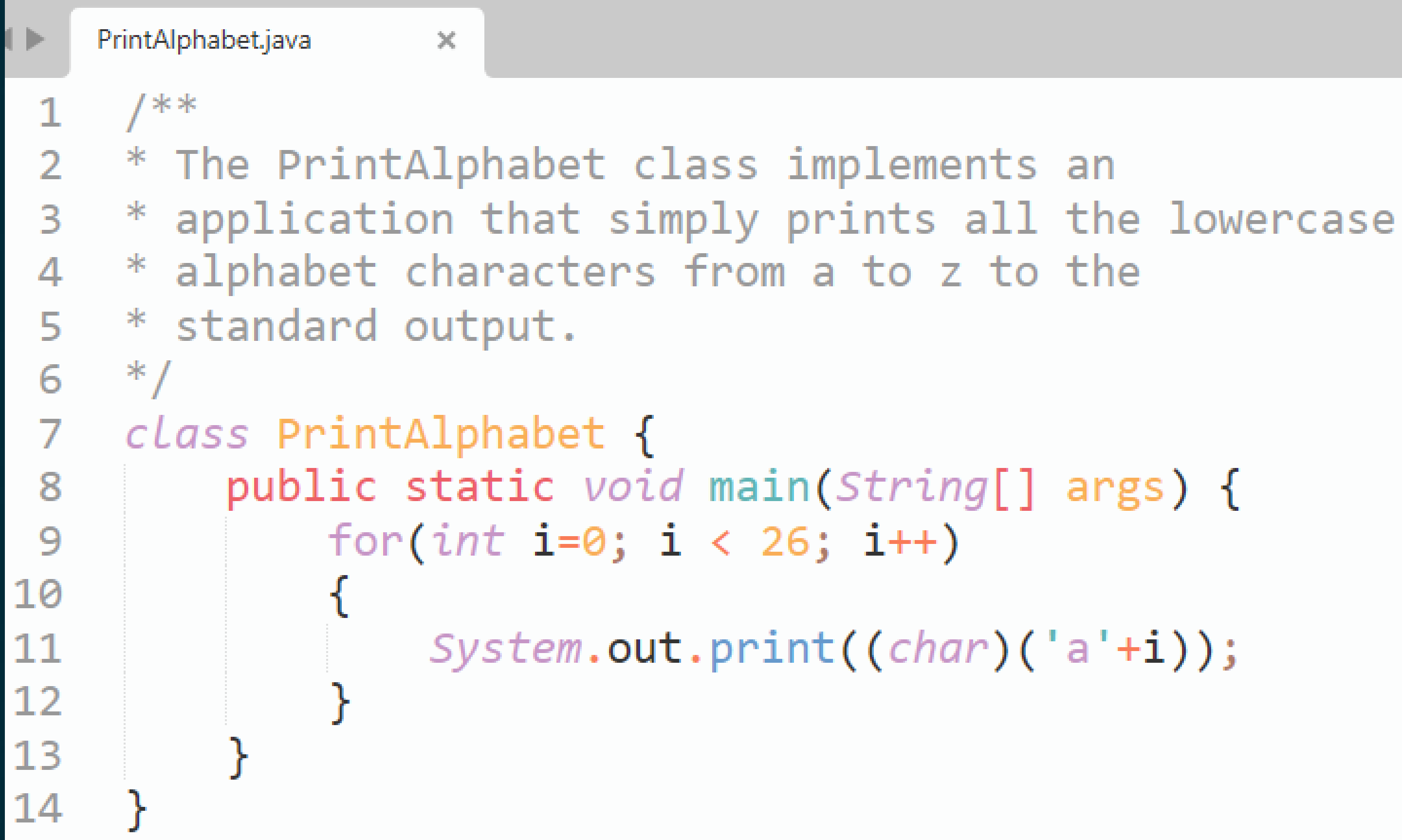
```
[redacted] >java HelloWorldProgram
Hello World!
```

# Java Programming Language



```java
/**
* The OddEven class implements an
* application that simply labels each number
* from 0 to 20 "Odd" or "Even" and displays both
* the number and label to the standard output.
*/
class OddEven {
    public static void main(String[] args) {
        for(int num=0; num <= 20; num++)
        {
            System.out.print(num + " - ");
            if(num % 2 == 0)
                System.out.println("Even");
            else
                System.out.println("Odd");
        }
    }
}
```

# Java Programming Language



```java
/**
* The PrintAlphabet class implements an
* application that simply prints all the lowercase
* alphabet characters from a to z to the
* standard output.
*/
class PrintAlphabet {
    public static void main(String[] args) {
        for(int i=0; i < 26; i++)
        {
            System.out.print((char)('a'+i));
        }
    }
}
```

# LO 1.1.1 Explain the difference between object-oriented and procedural programming paradigms

What is the difference between OOP and procedural programming paradigms?

# LO 1.1.2 Define valid OOP concepts and terminologies

Select an object from the real world and define its properties according to Booch’s definition.

# LO 1.1.3 Compile and run successfully a program using Java programming language

Implement a Java program that displays the Collatz conjecture sequence starting from 17. The *Collatz conjecture* is a sequence of numbers transforming through the function, $\mathrm{f(n)}$ and ends in the value $1$ since it will be a loop of $1\ 4\ 2$ from there.

$$f(n) = \begin{cases} \dfrac{n}{2}, & \text{if } n \equiv 0 \pmod{2} \\ 3n{+}1, & \text{if } \mathrm{n} \equiv 1 \pmod{2} \end{cases}$$

Example: $17,\ 52,\ 26,\ 13,\ 40,\ 20,\ 10,\ 5,\ 16,\ 8,\ 4,\ 2,\ 1$