Introduction to Java

ACTLEA
Active Learning

Learning objectives

- Java Basics
- Using an IDE (Integrated Development Environment) like IntelliJ IDEA
- Creating a first program

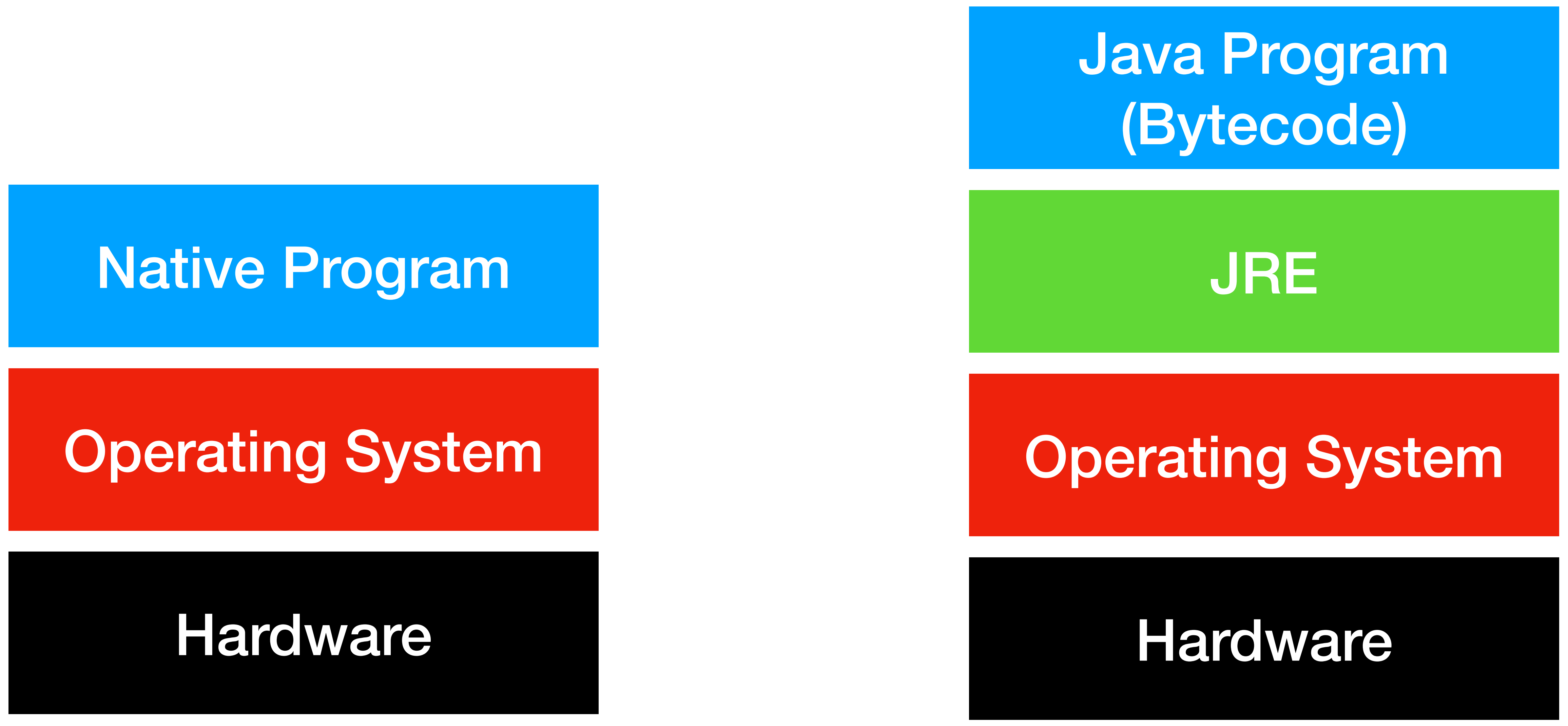
Java Basics

- Object-Oriented Language
- Portable code - write once, run anywhere
- Automatic memory management
- A language and a platform
- Very popular and trusted for enterprise applications

Java versions

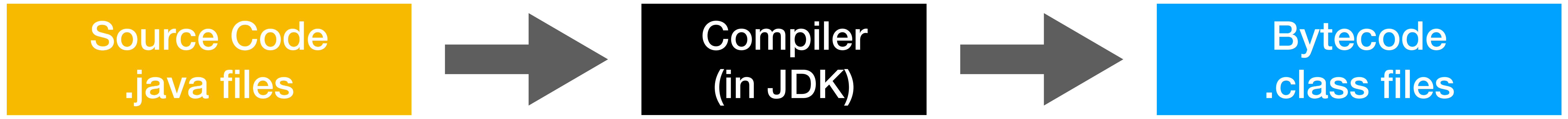
- A new version of Java is currently released every 6 months
- Some versions are LTS versions (Long Term Support)
- LTS versions are supported for a long time, other versions only for 6 months
- This course uses Java 8, one of the LTS versions

Java Runtime Environment (JRE)

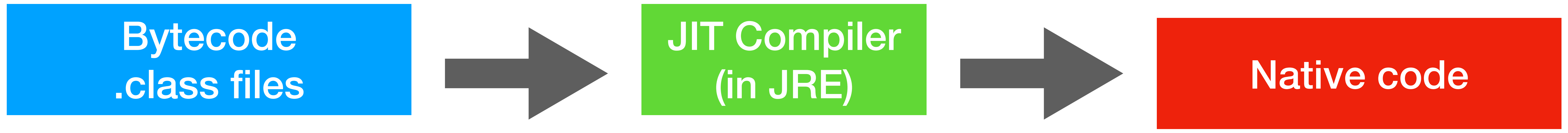


Compilation and Running

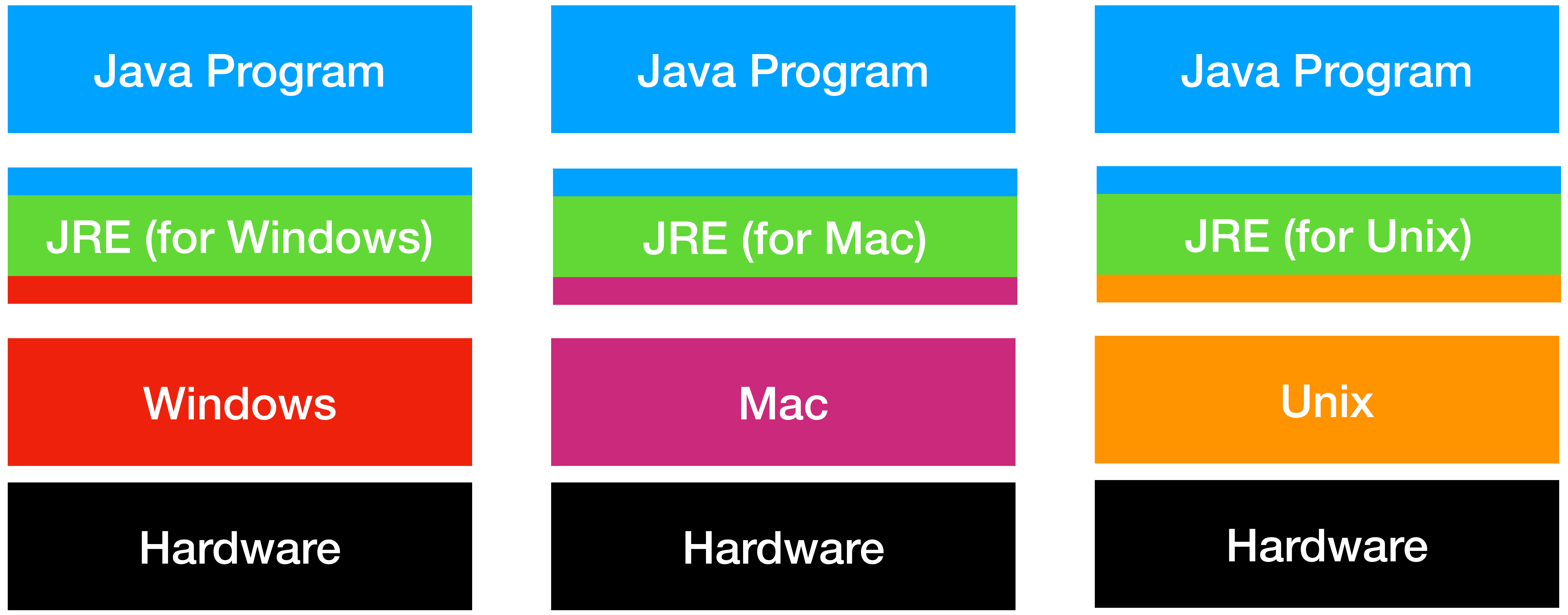
Compiling the source code into Bytecode with the JDK (Java Development Kit)



Running the compiled Bytecode with the JRE (Java Runtime Environment)



Portable code

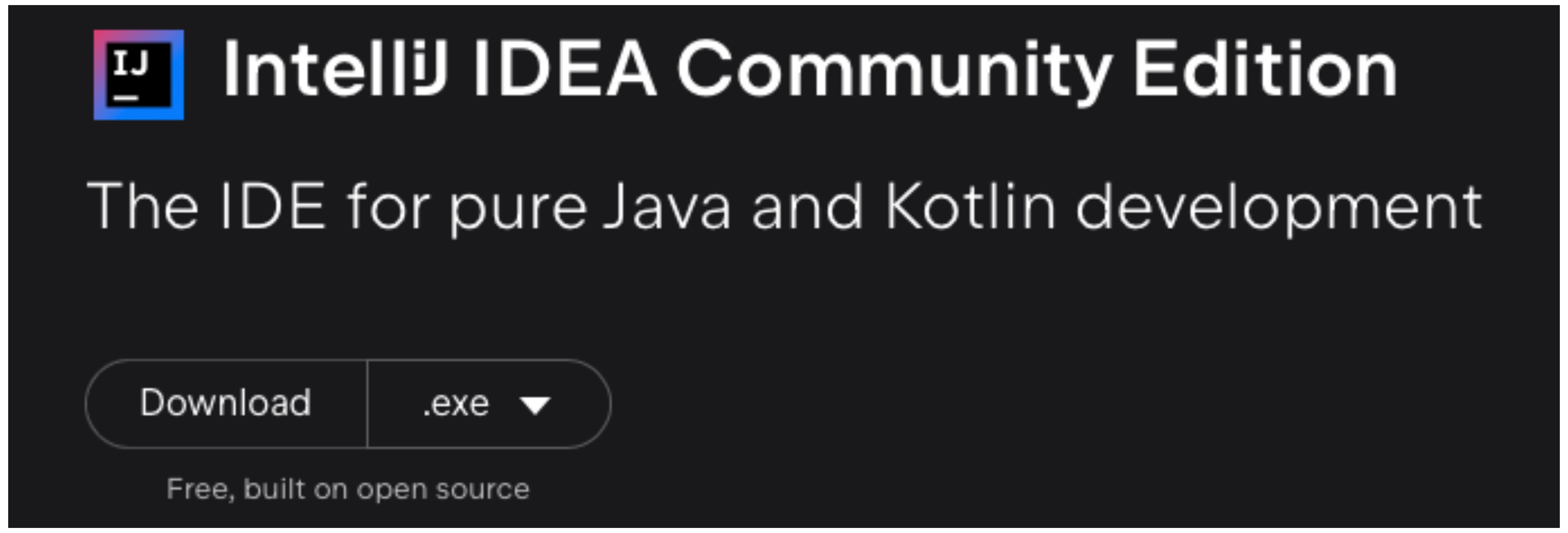


Working with the IDE (IntelliJ IDEA)

- An IDE (Integrated Development Environment) is used for programming
- IntelliJ IDEA is an IDE with many helpful tools for Java programming
- When running a program in IntelliJ, the source code will first be compiled to bytecode and then the bytecode will be run
- The debugger is really helpful when finding bugs!
- IntelliJ IDEA Community Edition is free and open source

Installing IntelliJ IDEA

- Download IntelliJ IDEA Community Edition for your operating system:
- https://www.jetbrains.com/idea/download/?section=windows
- Scroll down and download IntelliJ IDEA Community Edition:



IntelliJ IDEA Community Edition

The IDE for pure Java and Kotlin development

- Double click the .exe file and follow the installation guide

Demo 1 - Creating a program

- Creating a project in IntelliJ IDEA Community Edition
- Creating a Class
- Creating a main method
- Printing something to the console (output)

ACTLEA
Active Learning

Exercise 1 - Hello World!

- Create a Hello World project in IntelliJ IDEA

- Hint:

```
public class Main {

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

Java naming rules

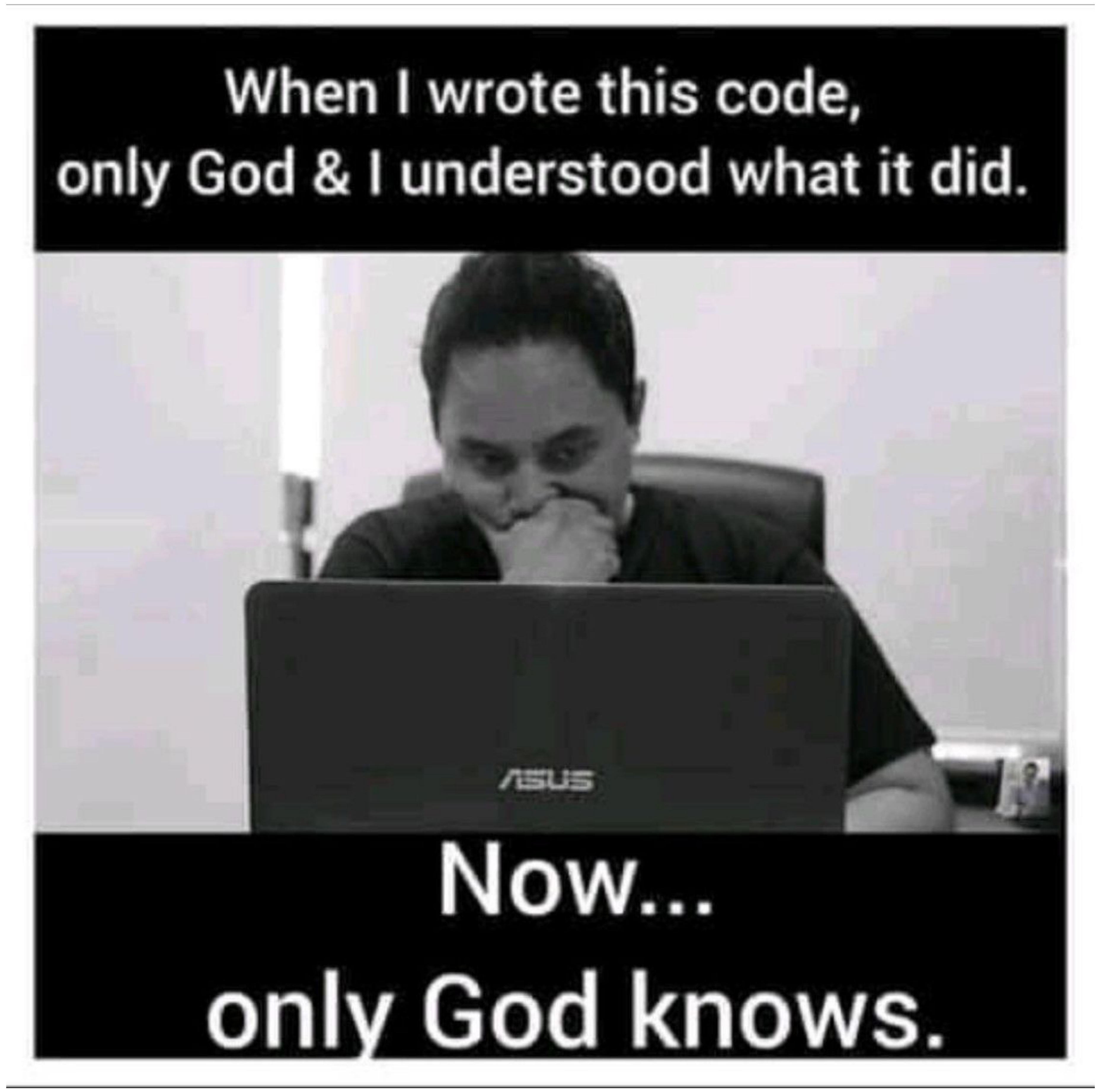
- Names of packages, classes, methods and variables cannot contain spaces or certain special characters
- Names can contain numbers, but not start with a number
- Names cannot be reserved words (words with specific meaning in Java)

Java naming conventions

- Packages: Only lowercase letters (for example com.example.helloworld)
- Classes: CamelCase starting with an uppercase letter (for example Main, MyClass, HelloWorld, Demo1)
- Methods: camelCase starting with a lowercase letter (for example main, testMethod, getName, setAge)

Java naming best practice

- Give descriptive names!
- The code should be easy to understand, both by others and by yourself in the future


When I wrote this code, only God & I understood what it did.
Now... only God knows.

Demo 2 - Packages, classes, methods

- Creating and using packages
- Creating classes and organize classes in packages
- How packages can be used to avoid naming conflicts
- Creating methods within classes

Exercise 2 - Classes and packages

- Create two classes with the same name in the same project (both can contain a main method with the Hello World solution)

- There should not be any errors in the project and the main method of each class should be able to run successfully

- Hint: To avoid a naming conflict, put the classes in different packages

CamelCase…

WOW! HOW DID YOU GET LIKE THAT?
Java devs
EVERY TIME I press the shift key
Java devs
...I DO ONE PUSH-UP.
Java devs
JESUS CHRIST.
SHEN COMIX
made with mematic

Learning objectives

- Java Basics
- Using an IDE (Integrated Development Environment) like IntelliJ IDEA
- Creating a first program