



Java Programming II

Juan Carlos Moreno - UCLA Ex

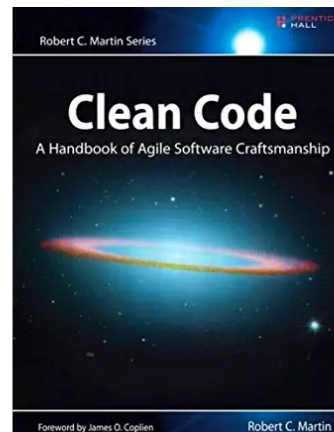
Intro

Recommended reading



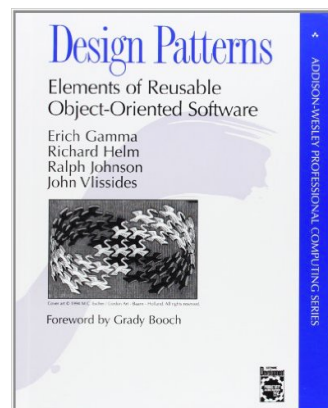
Java: A Beginner's Guide, Sixth Edition

[Herbert Schildt](#)



Clean Code: A Handbook of Agile Software Craftsmanship

[Robert C. Martin](#)



Design Patterns: Elements of Reusable Object-Oriented Software

[Erich Gamma](#), [Richard Helm](#), [Ralph Johnson](#), [John Vlissides](#), [Grady Booch](#)

Class Structure

How will we work?

- Hands on Exercises
- Working in teams
- Code reviews
- Quizes
- Final Project

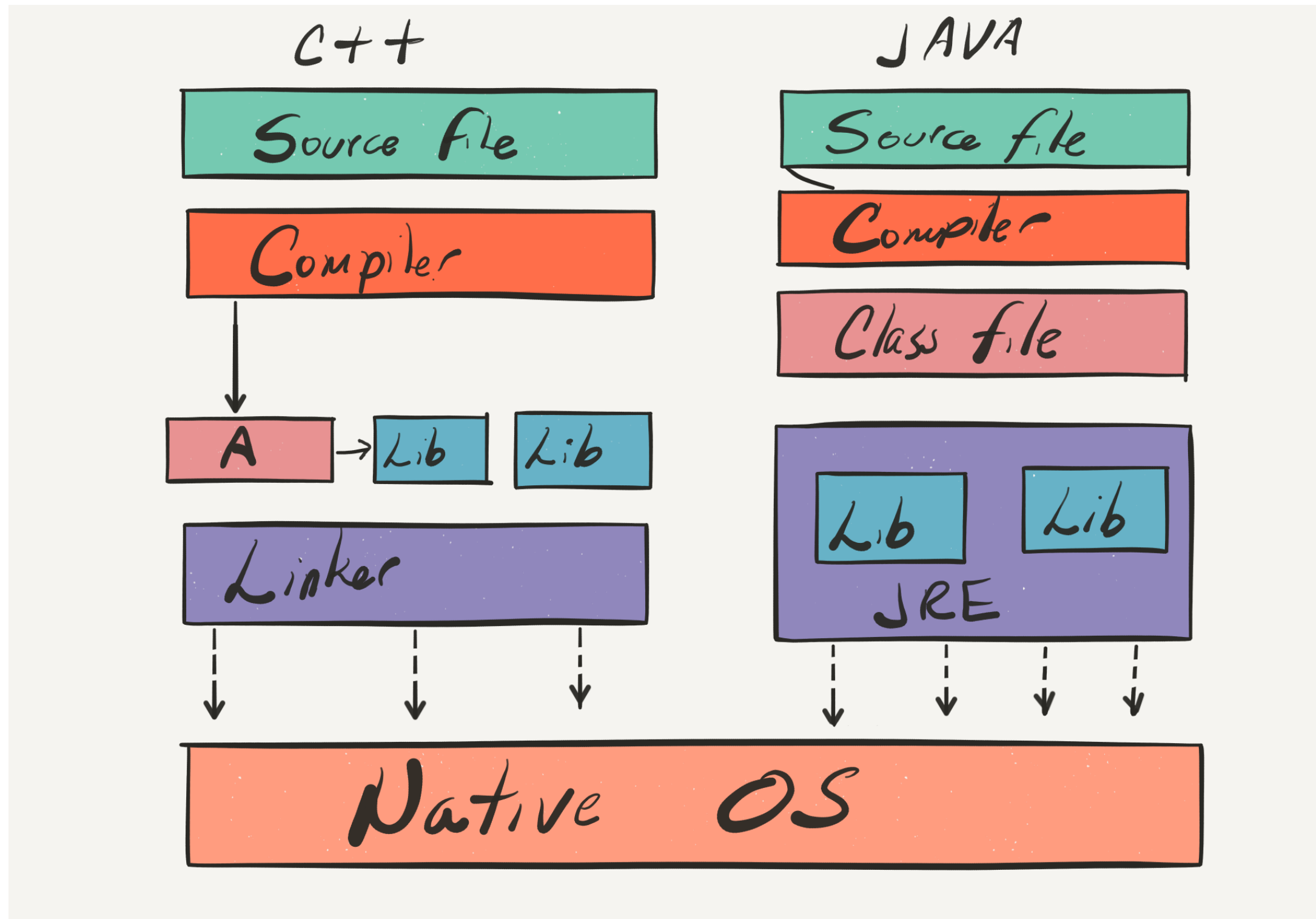


Grading

Make it as close as possible to a real-life experience

- Homeworks/Quizzes 30%
- Final Project 40%
- Calculator Project 30%

The flow of a Java Program



Statements & Blocks

Ingredients of a program

```
1      public void blockOfCode() {  
2          String statement_2 = "This is a statement";  
3  
4          String statement_1 = "This is all one "  
5                               + "single Statement";  
6  
7          if (true) {  
8              // Block of code  
9          }  
10     }
```

Variables

Initialization, Scope and lifetime

```
type name = value;
```

```
1  int other_variable = 10.5;
2  boolean variable = true;
3
4  if (variable){
5      int insideScope = 10; // Lives inside the brackets
6  }
7
8  int outsideScope = insideScope; // This is an error
9  System.out.println("My var:" + variable); // This is ok
```


Primitive Data Types

Representing data using text

Type	Meaning	Bits	Range
<i>boolean</i>	True or False		0 or 1
<i>byte</i>	8 bit integer	1 Byte	-128 - 127
<i>char</i>	Character	4 bytes	0 to 1,112,064
<i>double</i>	Very precise number	8 bytes	
<i>float</i>	Precise number	4 bytes	
<i>int</i>	Integer	4 bytes	-2 billion - 2 billion
<i>long</i>	Very long integer	8 bytes	-9 Quintillion - 9 Quintillion
<i>short</i>	Short Integer		-32,768 - 32,767

Everything is a class

Object oriented no matter what

```
1  public class RobotTom{
2      private int my_age = 25;
3      private String my_name = "Tom";
4
5      public RobotTom(String name, int age){
6          my_name = name;
7          my_age = age;
8      }
9
10     public void printMyData(){
11         System.out.println("Hello " + my_name + ", my age is: " +
my_age);
12     }
13
14     public static void main(String args[]){
15         RobotTom tom = new RobotTom("Tom", 10);
16         tom.printMyData();
17         System.out.println("Goodbye!");
18     }
19 }
```

Logical Operators

How to declare variables

==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
	Logical Or
&&	Logical And

Logical Operators

Class Exercise

Or

A	B	A or B
0	0	0
0	1	1
1	0	1
1	1	1

And

A	B	A and B
0	0	0
0	1	0
1	0	0
1	1	1

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

Operator Priority

Operator precedence

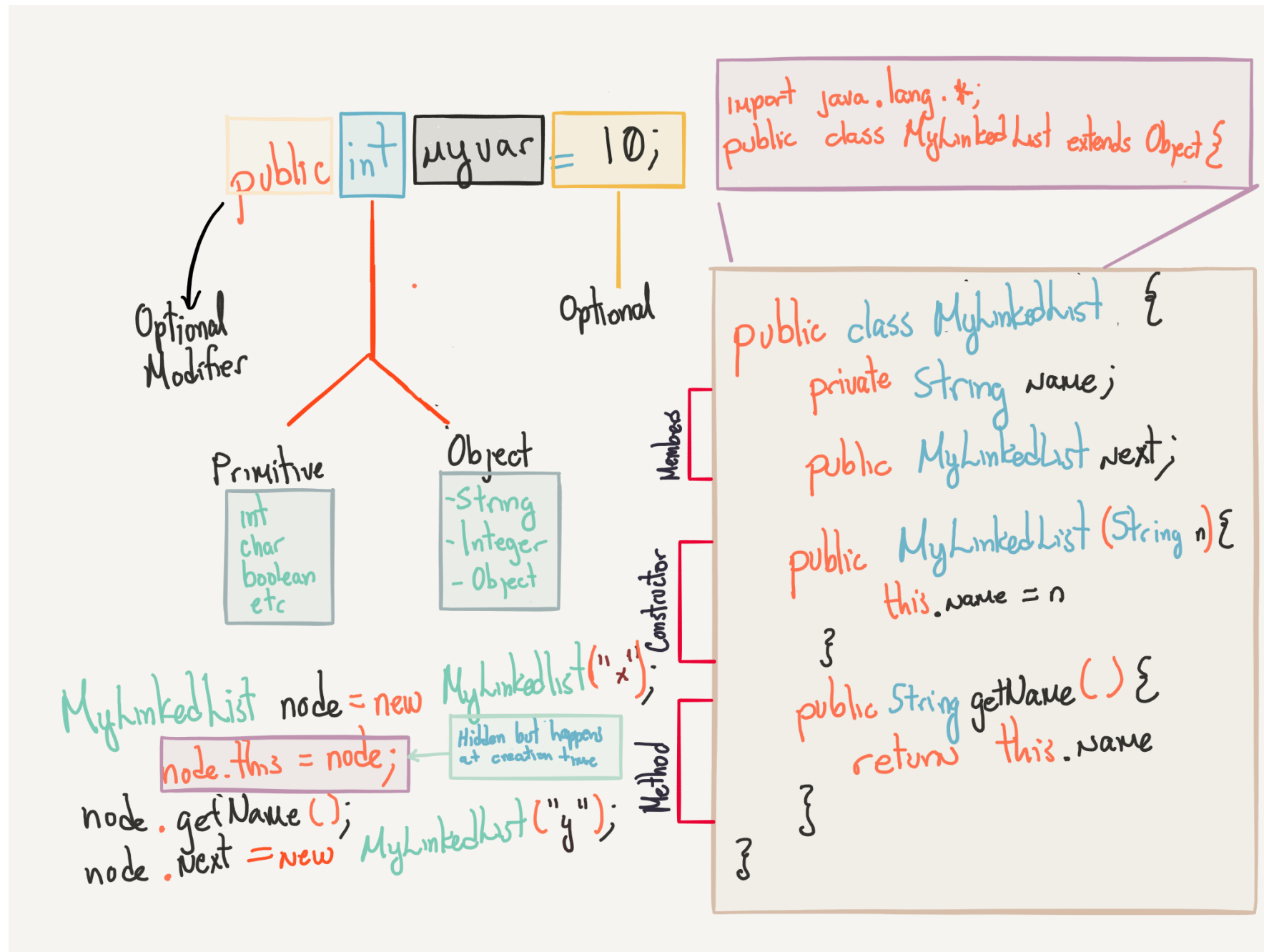
High	++ or --
	* then /
	+ then -
	== then !=
	&&
	
Low	=

$$100 + 2 * 10 / 4 = 105$$

$$2 * 10 = 20 \quad \rightarrow \quad 20 / 4 = 5 \quad \rightarrow \quad 5 + 100$$

Code Structure

Lets understand the what a class looks like



Control Structures

```
if (condition)  
{  
    //statement  
}
```

```
while (condition)  
{  
    //statement  
}
```

```
for(init;condition;increment) {  
    // statement  
}
```

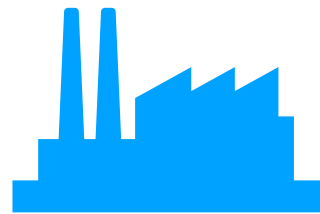
```
for (type name : iterator) {  
    // statement  
}
```

Hello World

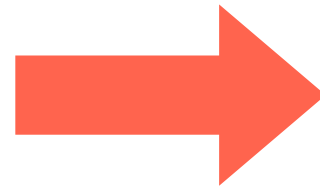
```
public class HelloWorld {  
    public static void main(String[] args) {  
        // Prints "Hello, World" to the terminal window.  
        System.out.println("Hello, World");  
    }  
}
```


Running it

Under the hood



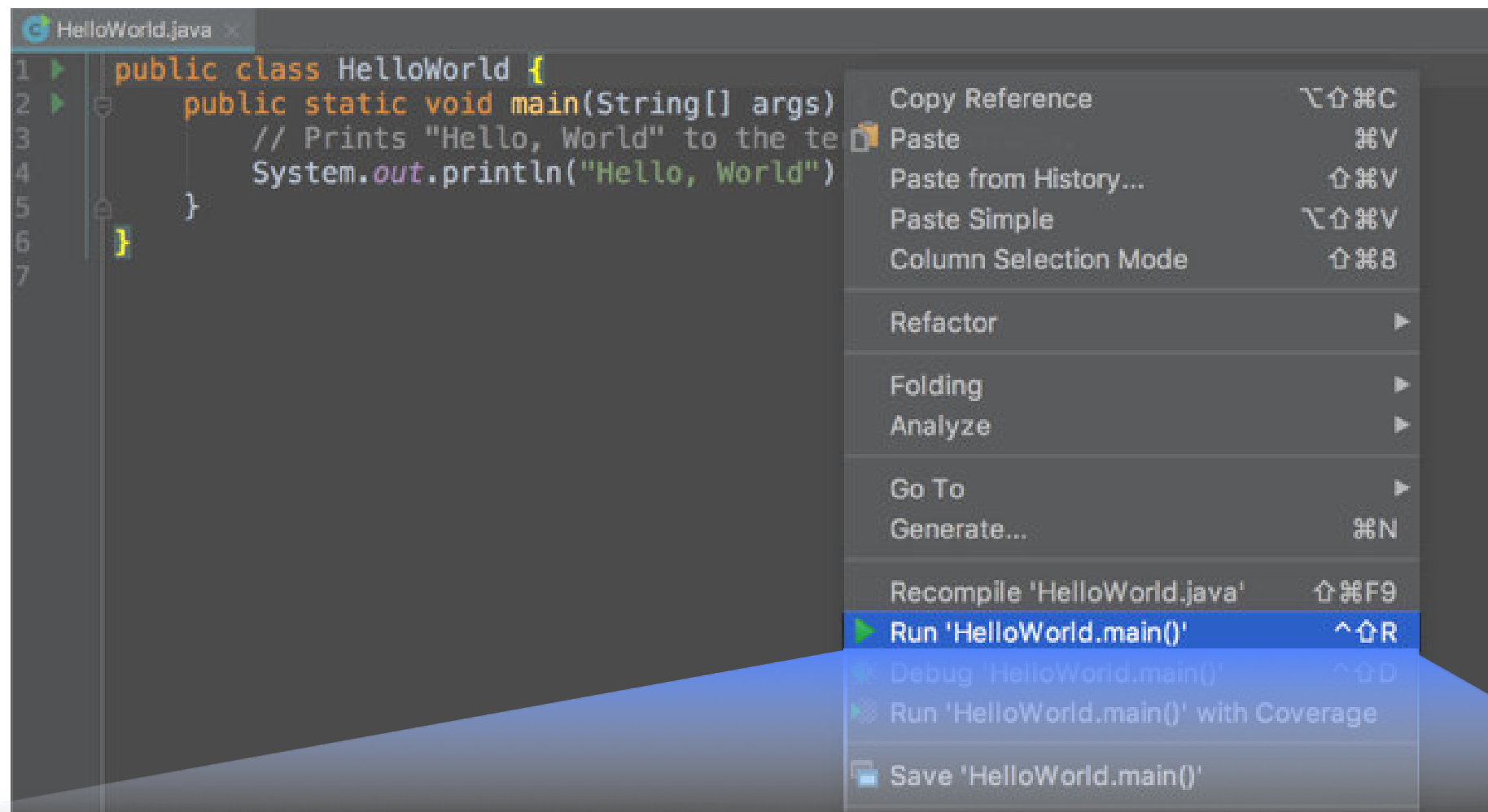
Build



Run

```
[moreno (17:19) ~$ javac HelloWorld.java  
[moreno (17:19) ~$ java HelloWorld  
Hello, World  
moreno (17:19) ~$ █
```

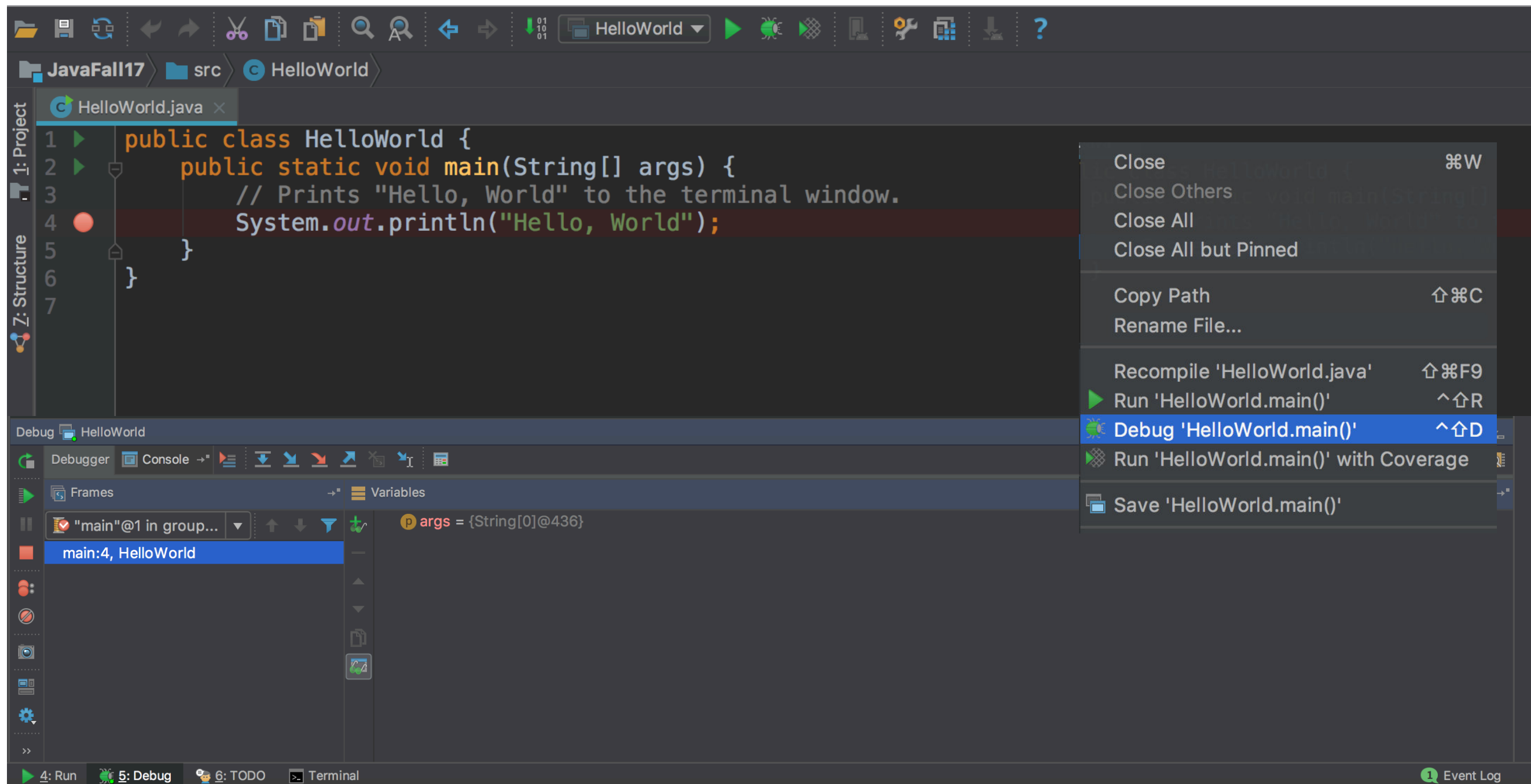
Running it



```
/Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar" -jar /Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/bin/java  
objc[840]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/bin/java and /Library/Java/JavaVirtualMachines/jdk1.8.0_20.jdk/Contents/Home/bin/java  
Hello, World
```

```
Process finished with exit code 0
```

Debugging it



Hello World ++

Classroom exercise

Learn how to

- Create new file
- Use IDE
- Expand code

Troubleshooting

- Handling compiler errors
- Handling runtime errors
- Debugging

```
public class HelloWorldPlusPlus{  
    public static void main(String[] args) {  
        String name = args[0];  
        System.out.print("Hello, " + name);  
        System.out.println(". How are you?");  
    }  
}
```