Goskills

Object Oriented Programming in Java

GoSkills online course syllabus

Skill level
Beginner

Lessons
34

Accredited by
Verified by GoSkills

Pre-requisites
No prior experience needed

Video duration
1h 6m

Estimated study time
1h 51m

Instructor
June Clarke

Introduction

1 It's time up your game in Java

The World of Objects

2 What is an object in Java?
You may have guessed that Objects are the foundation of Object Oriented Programming.

3 What is encapsulation?
Encapsulation is one of the fundamental concepts of object-oriented programming.

4 The world of object oriented design
To be a successful programmer, you need to write code that's readable and reusable.

5 Prerequisites
This course is best experienced in the quiet moments between you and your IDE. This lesson will point you in the right direction for getting set up.

6 Exercise files
This lesson will show you how to get the exercises for this course from GitHub, and how to import them into Eclipse so that you make the most of this course through hands-on practice.

Inside Classes

7 The anatomy of a Class

Classes are the main building blocks of Java code.

8 The mighty constructor

To make every object in Java, you need to invoke the mighty constructor!

9 Using the constructor

Watch a coding demo that will show you how to create a constructor and call it to create multiple instances of a Duck class.

10 Overriding the toString() method

Now that you're making your own classes, learn how to print their details elegantly, by overriding the toString method.

11 Exercise: Objects in Java

Practice makes perfect.

Getters & Setters

12 Private member variables

There are a number of ways to get access to the data inside of a class, but which is the correct way to maintain modularity and encapsulation in your code?

13 The inner workings of setters

Setters and Getters are key techniques to achieving encapsulation.

14 Using setters

There are 3 ways to modify the data within a class.

15 Testing with JUnit

The mark of a top notch software engineer is thoroughly tested code.

16 Exercise: Getters, Setters, & JUnit

Engage your knowledge of Getters & Setters with an exercise to create Minions.

Static

17 What is static?

You've probably noticed it on your main methods already, but what does it really mean, this mysterious word static?

18 Static coding demo

A static member variable belongs to the class itself, not to any instance of a class.

19 Breaking out of the main() method

Break free from the static context! Learn to use an instance of a class to gain access to the sweet methods inside.

20 Exercise: Static

Experiment on your own with an exercise where static will help you make sure that there is only one Papa Smurf and only one Smurfette.

Inheritance

21 What is inheritance?

Yes!

22 Extending behaviors

Watch me put inheritance into practice with a coding demo where I'll show you how you can use inheritance to add to, or extend the behavior of the Random() class.

23 The mother object

Learn about the omnipresent Object class from whom all objects are born.

24 Exercise: Build a better Arraylist

Put pedal to metal with an exercise! Use your inheritance skills to build a better ArrayList.

UML

25 The Unified Modeling Language (UML)

When you want others to understand and use your code, UML is your friend.

26 Drawing with UML

UML has a variety of standard methods to create diagrams. Learn the various connector types and what they mean.

27 Exercise: Create a UML diagram

You can put down your IDE for this one, and pick up a pencil.

28 Exercise Solution: UML Diagram

Verify your UML by hearing one possible solution that reveals the composition and inheritance relationships within a house.

Polymorphism

29 What is Polymorphism?

With polymorphism, subclasses can define their own behavior while still fitting within a specification, that opens up a world of possibilities for new object oriented patterns and architectures.

30 Polymorphism in action

See polymorphism brought to life with a code example that allows countries to define their own national dance.

31 The purpose of polymorphism

When subclasses have a common parent, we can do cool tricks with them, some of which I'll show with code in this lesson.

32 Implements vs. Extends

Get a little deeper with a brief nod to interfaces as an alternative to inheritance. And learn how parent and child classes can share information.

33 Exercise: Polymorphism

Flex your new polymorphic muscles by implementing a class hierarchy that describes the relationship between doctors, surgeons, and general practitioners.

Conclusion

34 Continuing object oriented programming in Java