Java Programming: Arrays, Lists, and Structured Data

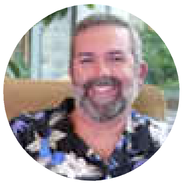
Duke University

About this Course
Build on the software engineering skills you learned in “Java Programming: Solving Problems with Software” by learning new data structures. Use these data structures to build more complex programs that use Java’s object-oriented features. At the end of the course you will write an encryption program and a program to break your encryption algorithm.

After completing this course, you will be able to:
1. Read and write data from/to files;
2. Solve problems involving data files;
3. Perform quantitative analyses of data (e.g., finding maximums, minimums, averages);
4. Store and manipulate data in an array or ArrayList;
5. Combine multiple classes to solve larger problems;
6. Use iterables and collections (including maps) in Java.

⌃ Show less

Taught by: Andrew D. Hilton, Assistant Professor of the Practice
Electrical and Computer Engineering



Taught by: Robert Duvall, Lecturer
Computer Science

Taught by: Owen Astrachan, Professor of the Practice
Computer Science

Taught by: Susan H. Rodger, Professor of the Practice
Computer Science

Basic Info	Course 2 of 4 in the Object Oriented Programming in Java Specialization
Level	Beginner
Commitment	4 weeks of study, 4-8 hours/week
Language	English Volunteer to translate subtitles for this course
How To Pass	Pass all graded assignments to complete the course.
User Ratings	★★★★⯪ 4.6 stars

Syllabus

WEEK 1

Welcome

Welcome to "Java Programming: Arrays, Lists, and Structured Data"! We are excited that you are starting our course to learn how to write programs in Java, one of the most popular programming languages in the world. In this introductory module, you will hear an overview of this course and be introduced to the supporting resources available.

1 video, 2 readings

1. **Video:** Welcome
2. **Reading:** Programming Resources
3. **Reading:** Feedback surveys

Show less

Cryptography: Keeping Information Secret

In this module, you will learn about the basics of cryptography, the science of keeping information private and secure. You will learn about simpler cryptographic systems, which were used from the Roman Empire through the early 1900s. You will learn how to implement these ciphers, as well as how to break them. To solve these problems, you will work more with processing Strings, but also learn about arrays—a way to store an indexable sequence of elements. You will be able to: (1) combine Strings using concatenation; (2) build Strings within a Java program using StringBuilder; (3) use arrays to store and manipulate collections of data; (4) refactor your programs for improved organization using object-oriented principles; (5) and practice effective algorithm design.

Show less

21 videos, 5 readings, 3 practice quizzes

1. **Reading:** Module Learning Outcomes / Resources
2. **Video:** A Brief History of Cryptography
3. **Video:** Introduction
4. **Video:** Creating and Manipulating Strings
5. **Video:** Counting Loops
6. **Video:** Character Class
7. **Video:** Developing an Algorithm
8. **Video:** Translating into Code
9. **Video:** Testing and Debugging
10. **Video:** Summary
11. **Reading:** Programming Exercise: Implementing the Caesar Cipher
12. **Practice Quiz:** Implementing the Caesar Cipher
13. **Video:** Introduction
14. **Video:** Arrays
15. **Video:** Random Numbers and Arrays
16. **Video:** Counting with Arrays
17. **Video:** Developing an Algorithm

18. **Video:** Summary
19. **Reading:** Programming Exercise: Breaking the Caesar Cipher
20. **Practice Quiz:** Breaking the Caesar Cipher
21. **Video:** Introduction
22. **Video:** Rewriting with Encapsulation
23. **Video:** Fields
24. **Video:** Visibility
25. **Video:** Constructors
26. **Video:** Summary
27. **Reading:** Programming Exercise: Object Oriented Caesar Cipher
28. **Practice Quiz:** Object Oriented Caesar Cipher
29. **Reading:** End of Module Survey

Show less

Graded: Cryptography

WEEK 2

GladLibs: Stories from Templates

After completing this module, you will be able (1) to program a word frequency counter to analyze any input text file, (2) to select and substitute words from a list into a document template using both ArrayList and HashMap, (3) to create new lists to use in templates, (4) to recognize brittle code, and (5) to improve code with flexible, object-oriented design. You will gain these skills in the framework of developing a randomly generated story that we call GladLibs. You may discover that bald lions change peoples' lives, or that fluffy dinosaurs get things done in a jiffy. We hope you have fun developing your Java skills this week!

Show less

13 videos, 5 readings, 2 practice quizzes

1. **Reading:** Module Learning Outcomes / Resources
2. **Video:** Introduction
3. **Video:** High-level Design Concepts
4. **Video:** ArrayList

5. **Video:** ArrayList for Unique Words
6. **Video:** ArrayList Advantages and Issues
7. **Video:** Summary
8. **Reading:** Programming Exercise: Telling a Random Story
9. **Practice Quiz:** Telling a Random Story
10. **Video:** Introduction
11. **Video:** Brittle Code
12. **Video:** Adding New Labels
13. **Reading:** Programming Exercise: Using GladLibs
14. **Video:** HashMap
15. **Video:** HashMap for Unique Words
16. **Video:** HashMap for Flexible Design
17. **Video:** Summary
18. **Reading:** Programming Exercise: Improving GladLibs
19. **Practice Quiz:** Using and Improving GladLibs
20. **Reading:** End of Module Survey

Show less

Graded: GladLibs

WEEK 3

Web Server Logs: From Logs to Visits

In this module, you will learn about how web server logs store information about visitors to a website, and you will write programs to access information like user IP address, date and time of access, and more. Using Java programs you write in this module, you will be able (1) to read information from a web server log, (2) to count the number of unique visitors to your website, and (3) to count the number of times each visitor uses your website.

15 videos, 5 readings, 3 practice quizzes

1. **Reading:** Module Learning Outcomes / Resources
2. **Video:** Introduction
3. **Video:** Understanding Log Files

4. **Video:** LogEntry Class with toString
5. **Video:** Parsing Log Files
6. **Video:** Summary
7. **Reading:** Programming Exercise: Reading Log Files
8. **Practice Quiz:** Reading Log Files
9. **Video:** Introduction
10. **Video:** Developing an Algorithm
11. **Video:** Translating to Code
12. **Video:** Equality
13. **Video:** Summary
14. **Reading:** Programming Exercise: Finding Unique IP Addresses
15. **Practice Quiz:** Finding Unique IP Addresses
16. **Video:** Introduction
17. **Video:** Developing an Algorithm
18. **Video:** Translating to Code
19. **Video:** HashMap for Unique IPs
20. **Video:** Summary
21. **Reading:** Programming Exercise: Counting Website Visits
22. **Practice Quiz:** Counting Website Visits
23. **Reading:** End of Module Survey

Show less

Graded: Web Server Logs

WEEK 4

MiniProject: Vigenère Cipher

In this module, you will develop a program to break the Vigenère Cipher, a more complex version of the Caesar Cipher. You will improve your program in three stages: first decrypting messages where you know the language and key length, then adding the capability to handle messages with unknown key length, then extending the program to handle messages in a range of possible languages. Through this project, you will be able (1) to effectively use HashSet and HashMap, (2) to expand an algorithm from solving a simpler problem to handle broader, more complex problems, and (3) to design and modify program code involving a more complex collection of classes, methods, and data.

Show less

4 videos, 6 readings, 2 practice quizzes

1. **Reading:** Module Resources
2. **Video:** Introduction
3. **Video:** Known Language and Key Length
4. **Reading:** Programming Exercise: Known Language and Key Length
5. **Practice Quiz:** Known Language and Key Length
6. **Video:** Unknown Key Length
7. **Reading:** Programming Exercise: Unknown Key Length
8. **Practice Quiz:** Unknown Key Length
9. **Video:** Unknown Language
10. **Reading:** Programming Exercise: Unknown Language, Unknown Key Length
11. **Reading:** Extend Your Program
12. **Reading:** End of Module Survey

Show less

Graded: Breaking the Vigenère Cipher

View Less

How It Works

GENERAL

How do I pass the course?

To earn your Course Certificate, you'll need to earn a passing grade on each of the required assignments—these can be quizzes, peer-graded assignments, or programming assignments. Videos, readings, and practice exercises are there to help you prepare for the graded assignments.

What do start dates and end dates mean?

Most courses have sessions that run multiple times a year — each with a specific start and end date. Once you enroll, you'll have access to all videos, readings, quizzes, and programming assignments (if applicable). Peer-graded assignments can only be submitted and reviewed once your session has begun. If you choose to explore the course without purchasing, you may not be able to access certain assignments. If you don't finish all graded assignments before the end of the session, you can enroll in the next session. Your progress will be saved and you'll be able to pick up where you left off when the next session begins.

What are due dates? Is there a penalty for submitting my work after a due date?

Within each session there are suggested due dates to help you manage your schedule and keep coursework from piling up. Quizzes and programming assignments can be submitted late without consequence. However, it is possible that you won't receive a grade if you submit your peer-graded assignment too late because classmates usually review assignment within three days of the assignment deadline.

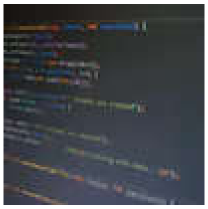
Can I re-attempt an assignment?

Yes. If you want to improve your grade, you can always try again. If you're re-attempting a peer-graded assignment, re-submit your work as soon as you can to make sure there's enough time for your classmates to review your work. In some cases you may need to wait before re-submitting a programming assignment or quiz. We encourage you to review course material during this delay.

Show less

Course 2 of Specialization

Grow Your Portfolio as a Software Engineer. Learn about Object Oriented Design in four project-based courses.



Object Oriented Programming in Java

University of California, San Diego, Duke University

View the course in catalog

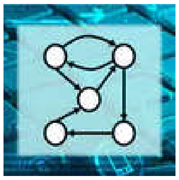
Related Courses

Java Programming: Principles of Software Design

Duke University

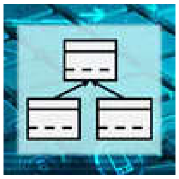
Java Programming: Solving Problems with Software

Duke University



Advanced Data Structures in Java

University of California, San Diego



Object Oriented Programming in Java

University of California, San Diego

Data Structures and Performance

University of California, San Diego