Introduction to Java and Object-Oriented Programming

Instructor

Brandon Krakowsky, Lecturer at the School of Engineering and Research & Education Director at the Wharton School

Course Description

This course provides an introduction to the Java language and object-oriented programming, including an overview of Java syntax and how it differs from a language like Python. Students will learn how to write custom Java classes and methods, and how to test their code using unit testing and test-driven development. Topics include basic data structures like Arrays and ArrayLists and overloading methods.

Course Learning Objectives

- Identify core aspects of object-oriented programming and features of the Java language
- Use Eclipse for writing and running Java code
- Develop programs that use Java collections and apply core object-oriented programming concepts using classes, polymorphism, and method overloading
- Test code by applying principles of test-driven development using Java's unit testing framework

Intended Audience

This course is intended for students and professionals who have minimal or no prior programming exposure. It's for motivated learners who have experience with rigorous coursework, and are looking to gain a competitive edge in advancing their career.

Course Prerequisites

- High school or college math.
- "Introduction to Python Programming" Coursera course or equivalent prior knowledge of basic programming concepts.

Penn Engineering
ONLINE LEARNING

Course Outline

Module 1: Introduction to Java, Classes, & Eclipse

- Learning Outcomes
  - Identify core aspects of object-oriented programming and features of the Java language
  - Use Eclipse for writing and running Java code
  - Understand and apply core object-oriented programming concepts using classes
  - Write code to develop a game that does basic math and converts time
  - Design a banking system for managing customers and their bank accounts
- Topics
  - Introduction to Java
  - Configuring Java & Tools
  - The Java Language
  - Classes

Module 2: Unit Testing, Arrays, & ArrayLists

- Learning Outcomes
  - Test code by applying principles of test-driven development using Java's unit testing framework
  - Create multi-dimensional Arrays to store collections of items
  - Discover Java's Collection Framework using ArrayLists
  - Apply the toString method to print objects and the equals method to compare objects
  - Write a program to manage seats in a classroom of students
- Topics
  - Unit Testing
  - About Equality
  - Arrays
  - Collections: ArrayLists
  - About ToString

Module 3: Static Variables, Methods, & Polymorphism Using Overloading

- Learning Outcomes
  - Differentiate the use cases for instance variables and static variables
  - Define static helper methods in a class
  - Define static variables to store hard-coded information
  - Manipulate static variables to share data across instances of a class
  - Discover polymorphism and write a program illustrating method overloading
  - Design a customer tracking system using static variables and methods

- Topics
    - Static Variables
    - Static Methods
    - More Examples of Static Variables and Methods
    - Polymorphism - Overloading

Course Assessment

This course will use a variety of assessments. Ungraded code-along videos allow students to practice along with the instructor, and self-assess their ability to apply the concepts and skills they learned, before attempting the graded assessments. Graded assessments include:

- Quizzes to check your knowledge in each module
- Programming assignments to test your level of understanding

To earn a certificate in this course, learners must earn a passing score on all assessments:

- Homework Assignments: 60% or above
- Quizzes: 75% or above

Recommended Resources

- *Head First Java*, by Kathy Sierra: https://www.amazon.com/Head-First-Java-Kathy-Sierra-dp-0596009208/dp/0596009208/
- *Java in Easy Steps*, by Mike McGrath: https://www.amazon.com/Java-easy-steps-Mike-McGrath-dp-1840788739/dp/1840788739/

Effort

We expect this course will take you 5-7 hrs per week to complete, for a total of 3 weeks.

Communication and Support

You can communicate with course staff and other students through the discussion forums. Please reach out to us through the discussion forum with any questions about the course content. Please allow at least 48 hours to receive a response from a TA or course staff.

Note: All communication on the discussion forum must follow the Coursera Honor Code . Never post code or solutions to assignments on the discussion forum. If you are having difficulty with code or solutions, a TA may provide an email address to send it in for private assistance.