

Guide: Learning Java

Java in a Week

Get a copy of "Thinking in Java" and start reading it.

Day 1: A Basic Java Program

1. Install Eclipse, IntelliJ or the IDE of your choosing
2. Determine the version of Java you are using
3. Understand the difference between a JRE and a JDK
4. Understand CLASSPATH and PATH
5. Understand .java and .class files
6. Write a simple program that prints out "Hello World"
7. Modify the program to print out "Hello *your name*" 5 times
8. Modify the program to take in two command line parameters. Now you should run it with two arguments (java *YourProgram* name number). The program should printout "Hello" name the number of times specified. For example:

Java MyProgram Bob 4

Should print out:

Hello Bob

Hello Bob

Hello Bob

Hello Bob

Day 2: Working with Objects

1. Review control flow options
 - a. If/else
 - b. For
 - c. While
 - d. Switch
2. Understand the difference between primitives and objects
3. Understand the difference between a Class and an Object (instance)
4. Create a new class called Student
5. Create some Student attributes
 - a. First name
 - b. Last name
 - c. Id#
 - d. Age
 - e. Zipcode
6. Create some Student methods
 - a. addATestResult
 - b. getATestResult
 - c. getGPA
7. Create a new class called Tester
8. Create a new Student object
9. Add some grades
10. Print out the students name
11. Print out the grade from one of the tests
12. Print out the students GPA

Day 3: Designing with Objects

1. Review Packaging
2. Review Abstract class
3. Review Interface
4. Review Error Handling
5. Design some classes to represent tools a teacher might need
 - a. Student Roster
 - b. Seating Chart
 - c. Grade Book
6. Required Features
 - a. Code must be packaged
 - b. Seating chart must use mutli-dimensional array
 - c. Must have two types of Students
 - i. Regular (with getGPA)
 - ii. Honors (with getHonorsGPA and get HonorsCourses)
 - d. Must have Two methods for calculating letter Grade
 - i. Straight 10 point increments (100-90, 90-80, 80-70)
 - ii. Some type of weighted average
 - e. Read in test Results from a file and then write out high, low and average to a file. File format will be as follows:

TestName	Math5
Bob	90.0
Sam	72.0
Mary	85.0

Day 4-5: Application

1. Discuss private, protected, public
2. Discuss scoping
3. Discuss polymorphism
4. Discuss inheritance
5. Choose a favorite game. Design and implement that game in Java. Graphics are not required. Automated player is not required. Simply design the classes that represent the game and include the functionality for two people to play it.