Software 1 Eight Week Challenge!

Welcome to the 8 Week Challenge for passing Software 1 (Java 8)! This course covers basic software concepts. To complete this course, you will be required to pass the 1Z0-808 Oracle Certified Associate Java SE8 Programmer exam with a minimum passing score of 65%. This exam can be challenging, and has many trick questions, but we’ll guide you through it! You will receive daily assignments, for eight weeks, including an assignment that requires you to code in Java (coding is reinforcing!) and have one check in with your course mentor per week (Note: you can always book additional time with your course mentor if you want additional help!). If you complete our plan, you should be very ready for your certification exam!

If at any time you want to opt out of the daily emails, send an email to [cmsoftware@wgu.edu](mailto:cmsoftware@wgu.edu) to request you be taken off the list.

We will be using the following references for this Challenge:

1. UCertify Oracle Certified Associate Java SE 8 Programmer I exam by Enthuware: <https://lrps.wgu.edu/provision/74426853>
2. Java SE 7 Programmer I & II Study Guide" book by Kathy Sierra and Bert Bates: <https://lrps.wgu.edu/provision/80137499>

(Note- the Sierra and Bates reference is keyed to Java 7, not Java 8, but for the purposes of this test, the only differences are in Interfaces, and the additional topics of Lambda Expressions and the LocalDateTime class.)

1. Lynda.com’s Java SE8 new Features by David Gassner: <https://www.lynda.com/Java-tutorials/Java-SE-8-New-Features/156621-2.html>
2. Previously recorded webinars and cohorts by the Software 1 Course mentors

Week 1

Day 1:

1. Watch the video- Getting Started with the Software 1 Eight Week Challenge: <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=0623d377-f06c-409a-afba-872a7e77c60f>
2. Take the UCertify pre-assessment.

Day 2:

1. Go over the questions you got wrong during the pre-assessment. Note the areas where you need to concentrate most heavily.
2. Read the Sierra and Bates reference, Chapter 1 (skip enums)

Day 3:

1. Do the Sierra and Bates Self Test Problems, at the end of Chapter 1.
2. Go over the answers you got wrong.

Day 4:

1. Read UCertify Chapter 1 & do the Quizzes and Exercises in the Learn Mode.

Day 5:

1. Do Coding Exercise 1-1 on page 23 of the Sierra and Bates reference (Creating and Abstract Superclass and Concrete Subclass. Create an additional package. Put a Fruit subclass in an additional package.
2. Extend that exercise to test instance variables of the four different access types- public, protected, default and private.

Send your mentor your Day 5 code. Meet with a course mentor for a half an hour to review your progress, and ask any questions about this week’s material that you didn’t understand.

Week 2

Some rules on polymorphism:

1.      Instance variables are determined at compile time by the reference type.

2.      Instance methods are determined at run time by the object type polymorphically (overriding).

3.      When there’s a superclass reference to a subclass object, if the member isn’t in the superclass, the reference won’t be able to reach it, unless you cast the reference type back to the subclass object type.

4.      Static members are class members.  You can’t reach instance members from a static context (method) unless they are connected to a specific object.

5.      When there’s a superclass reference to a subclass object, the static method is determined by the reference, not the object.  Statics don’t care about objects. Statics don’t override.

Day 1: Read the Sierra and Bates reference, Chapter 2. Read UCertify, Chapter 5

Day 2: Do the Sierra and Bates SelfTest questions at the end of the chapter 2

Day 3: Do the UCertify Quizzes and Exercises for Chapter 5

Day 4: Watch this cohort on Inheritance:  [**https://wgu.adobeconnect.com/p9it0bpbf69/**](https://wgu.adobeconnect.com/p9it0bpbf69/?OWASP_CSRFTOKEN=a9660ecc76979527cadde04a1a1960b3df5270b4a92d17a24a01aebaa15c6283)

Day 5:

1. There are a lot of questions that hinge on your understanding of superclass references to subclass objects.  Modify your code from last week’s Coding Exercise so it demonstrates:
   1. Instance variables are determined at compile time by the object reference
   2. Instance methods are determined at run time by the object type polymorphically
   3. When using a superclass reference, you can only reach the members that exist in the superclass
   4. Static methods don’t override, they shadow.  So static methods are determined by the reference type
2. Program in the initializer block problem in Sierra and Bates Chapter 2 problem 5. Now change it so Hawk doesn’t extend Raptor. How does it change? Now create a Raptor in the new class Hawk. Demonstrate what happens when you move the main method to each of the different classes.

Send your mentor your Day 5 code. Meet with a course mentor for a half an hour to review your progress, and ask any questions about this week’s material that you didn’t understand.

Week 3

Day 1: Read Sierra and Bates Chapter 3, UCertify Chapter 4. Also this: <http://www.javaranch.com/campfire/StoryCups.jsp> and <http://www.javaranch.com/campfire/StoryPassBy.jsp>

Day 2: Do the Sierra and Bates SelfTest questions at the end of Chapter 3 & the UCertify Quizzes and Exercise for Chapter 4

Day 3: Watch this cohort on General Topics:[**https://wgu.adobeconnect.com/p7wo9sknihu/**](https://wgu.adobeconnect.com/p7wo9sknihu/)

Day 4:

1. Use your code from the prior weeks and write a method, Tastiness, with a return type of void. Have it overloaded with one version that accepts an int parameter, and another version that accepts an Apple as a parameter. Have the version that accepts the int parameter change its value inside the method, and the one that accepts the Apple parameter change its value inside the method. Now print out the value before the method is called, call the method, print out the value in the method, and print out the value in the main method after the method is called. You should see the difference between pass by reference and pass by value.

Next overload this method with a String as a parameter and an Integer as a parameter. Do these classes behave differently from the Apple class? Review the implications of immutability.

1. Do exercise 3-1 in Sierra and Bates p.178.
2. Do exercise 3-2 in Sierra and Bates p.207.

Day 5:

Download the Sierra and Bates practice test engine, and install your system. (See About the Download in Appendix C.) Be sure to give them a WGU email for the link! Do the first OCA practice test, and review the correct answers to the problems you didn’t get right.

Send your mentor your Day 4 code. Meet with a course mentor for a half an hour to review your progress, and ask any questions about this week’s material that you didn’t understand.

Week 4

Day 1: Read Sierra and Bates Chapter 4, UCertify Chapter 2

Day 2: Do the Sierra and Bates SelfTest questions at the end of Chapter 4 & the UCertify Quizzes and Exercise for Chapter 2

Day 3: Program in the problem described here: <http://stackoverflow.com/questions/18811636/instanceof-enthuware-ocajp> Practice the possible combinations of instanceof

Day 4: Read the section on this question, and program in this problem, and variations on it: <https://coderanch.com/t/662136/ocajp/certification/Evaluation-order-Enthuware-test> Here’s a chart with the operator precedence in Java: <http://introcs.cs.princeton.edu/java/11precedence/>

Day 5: Do the second Sierra and Bates OCA practice test, and review the correct answers to the problems you didn’t get right.

Send your mentor your Day 3 code. Meet with a course mentor for a half an hour to review your progress, and ask any questions about this week’s material that you didn’t understand.

Week 5

Here’s some information on multidimensional arrays.  Here are some great analogies from this resource: <http://programmers.stackexchange.com/questions/246718/how-does-the-fourth-dimension-work-with-arrays>

|  |  |
| --- | --- |
| 13down vote | In programming, arrays are quite easy to implement, but maybe not to understand.  Generally, each level of arrays means to have the content n-fold. That means   * int x[4] are 4 blocks, each of them containing an int. * int x[5][4] are 5 blocks, each of them containing an int[4]. * int x[3][5][4] are 3 blocks, each of them containing an int[5][4]. * int x[2][3][5][4] are 2 blocks, each of them containing an int[3][5][4].   How you are referring to them is up to you, but for better understanding, you have something like   * COLUMN for the last one * ROW for the second-last one * PAGE for the third-last one   Till here, I read it somewhere. In order to stay here, we can as well define   * BOOK for the fourth-last one * and maybe SHELF for the fifth-last one. (Or, if you prefer, SHELFROW so that we can continue.)   That said, I never saw array with more than 4 or maybe 5 dimensions in "wild life".  This way, you can define and imagine int x[6][2][3][5][4] as a collection of 6 "shelves", each having 2 books, each having 3 pages, each having 5 rows, each having 4 columns. |

And

When dealing with finite arrays, it's easy to find space.

Imagine a sheet of paper with a grid printed on it; you can write some information in each cell of the grid. That's a 2D array: row and column.

Put several of those sheets of paper in a file folder; that's a 3D array: page, row, and column.

Put several of those folders in a file box. 4D array: folder, page, row, column.

Arrange boxes in a rectangular grid on a wooden pallet. 6D array: box-row, box-column, folder, page, row, column.

Stack more grids of boxes on top of those. 7D array: box-depth, box-row, box-column, folder, page, row, column.

Start cramming pallets into a shipping container: 9D array. (Assuming each stack is as tall as the inside of the container, so you can only get 2 more dimensions here.)

Stack up shipping containers on the deck of a container ship: 12D array.

Your fleet of container ships is now a 13D array.

Day 1: Read Sierra and Bates Chapter 5, UCertify Chapter 3. Here are some additional references on arrays:

Here are some references on two dimensional arrays and a little more:

<http://www.homeandlearn.co.uk/java/multi-dimensional_arrays.html>

<http://www.java-samples.com/showtutorial.php?tutorialid=265>

<http://www.dummies.com/how-to/content/java-use-arrays-with-two-dimensions-or-more.html>

<https://www.youtube.com/watch?v=dj15BrhCHIc>

[http://pkris001.freeshell.org/OCA.pdf Section 4.3](http://pkris001.freeshell.org/OCA.pdf%20Section%204.3)

Day 2: Do the Sierra and Bates SelfTest questions at the end of Chapter 5 & the UCertify Quizzes and Exercise for Chapter 3

Day 3: Watch this cohort on Arrays and ArrayLists:[**https://wgu.adobeconnect.com/p6xkxabamj2/**](https://wgu.adobeconnect.com/p6xkxabamj2/?OWASP_CSRFTOKEN=8baf16daf3de2e822f2e0852d23bf8a09061d60c20c2944bd6496d8a2276caac)

Day 4: Program examples of code that use the methods of the String, StringBuilder, arrays and ArrayLists. Become comfortable with using substring and arraycopy in particular.

Day 5: Do Practice Test A in UCertify in the test mode and review the problems you didn’t get right.

Send your mentor your Day 4 code. Meet with a course mentor for a half an hour to review your progress, and ask any questions about this week’s material that you didn’t understand.

Week 6

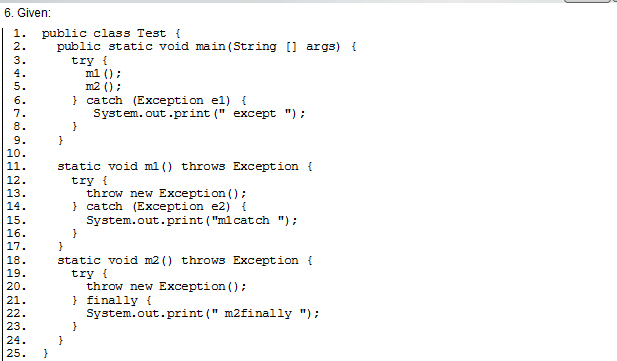
Day 1: Read Sierra and Bates Chapter 6, UCertify Chapter 6

Day 2: Do the Sierra and Bates SelfTest questions at the end of Chapter 6 & the UCertify Quizzes and Exercise for Chapter 6

Day 3: Watch this cohort on Exceptions:[**https://wgu.adobeconnect.com/p483opywjja/**](https://wgu.adobeconnect.com/p483opywjja/?OWASP_CSRFTOKEN=8baf16daf3de2e822f2e0852d23bf8a09061d60c20c2944bd6496d8a2276caac)

Day 4:

1. Program in the following problem and look at the results for it and variations on it:



1. Program Sierra and Bates Exercise 6-1 (p.310), 6-2(p.333), 6-3 (p.341) and 6-4 (p.353).

Day 5: Do Practice Test B in UCertify in the test mode and review the problems you didn’t get right.

Send your mentor your Day 4 code. Meet with a course mentor for a half an hour to review your progress, and ask any questions about this week’s material that you didn’t understand.

Week 7

Day 1: View Lynda.com’s Java SE8 new Features by David Gassner, Chapters 2 & 4: <https://www.lynda.com/Java-tutorials/Java-SE-8-New-Features/156621-2.html>

Day 2: Watch this cohort on Lambdas, Interfaces and the LocalDateTime class: [**https://wgu.adobeconnect.com/p6fi55epyde/**](https://wgu.adobeconnect.com/p6fi55epyde/?OWASP_CSRFTOKEN=a597d32067666e98d06d8c47e91e163b0a3c89ab819bbcbfb411ce0c5220eb6d)

Day 3: Do Practice Test C in UCertify in the test mode and review the problems you didn’t get right.

Day 4: Do Practice Test D in UCertify in the test mode and review the problems you didn’t get right.

Day 5: Do Practice Test E in UCertify in the test mode and review the problems you didn’t get right.

Meet with a course mentor for a half an hour to review your progress, and ask any questions about this week’s material that you didn’t understand. Decide together if you are ready to schedule your attempt.

Week 8

Day 1: Do Practice Test F in uCertify in the test mode and review the problems you didn’t get right.

Day 2: Do the uCertify Post-Assessment in the test mode, and review the problems you did not get right.

Days 3-5: Retake earlier pre-assessment and practice tests and review the problems you did not get right.

Meet with a course mentor for a half an hour to review your progress, and ask any questions about this week’s material that you didn’t understand. If you have not done so yet, decide together if you are ready to schedule your attempt, and if you are not ready, decide with your mentor on next steps.

Good luck on your certification exam!