CSC207 Practice Exam Questions:

Last Modified: `Sunday 20 November 2016`

The exam will test a subset of the topics discussed in this practice exam. It will have fewer questions with a different format.

Solutions will not be posted to these questions. Instead, we will be discussing the answers during the last lecture of the semester. You can also discussion the solution to these questions in any office hours for this course, including:

Date	**Time**	**Instructor**	**Location**
Mon., Dec 5	3–4 PM	Lindsey	BA3219
Tues., Dec 6	1–3 PM	Paul	BA4234
Wed., Dec 7	10–12 PM	Lindsey	BA3219
Wed., Dec 7	2–4 PM	Justin	BA3219
Fri., Dec 9	11–2 PM	Lindsey	BA3219
Mon., Dec 12	3–4 PM	Lindsey	BA3219
Tues. Dec 13	1–3 PM	Paul	BA4234
Wed., Dec 14	10-12 PM	Lindsey	BA3219
Wed. Dec 14	2–4 PM	Justin	BA3219
Fri., Dec 16	11-2 PM	Lindsey	BA3219

Questions:

1. What are the main components of a class in Java? What are the standard accessibility modifiers for each (e.g., protected, private, etc.)? When would you want to use a non-standard accessibility modifier for a variable? for a method? for a constructor?

2. In what ways are methods similar to constructors? In what ways are they different? (Consider: the syntax for coding each, how they show up in the Java memory model, features such as calling other constructors, whether or not Java provides them by default, returning values, the applicability of terms like "instance" and "static", etc..)

3. Is it possible for a class to have more than one parent class? More than one child class? To implement more than one interface?

4. In which ways are abstract classes and interfaces similar? In what ways are they different? List the conditions which are necessary and sufficient for a: (a) class to be declared abstract and (b) for a method to be abstract.

5. What do the following keywords mean when used in front of a method: `final`, `static`, `abstract`.

6. List all of the primitive types. How does the memory model reflect the differences between primitive types and objects. When are two primitive variables "equal"? When are two objects "equal"? Why is the word "equal" in quotes? Are all non-primitive types subclasses of the `Object` class? What features of class `Object` did we use most frequently in the lectures?

7. What do we mean by "casting", "autoboxing", and "wrapper class". Compare and contrast these terms.

8. Give four examples of subclasses of `Collection` in Java. Describe a different circumstance for each in which you would require the features of that particular type of collection. For example, how when would you use an `ArrayList`? How are collections similar to arrays? How are they different? Is it possible to define a non-generic subclass of `Collection`? Why or why not?

9. Write a main method that creates three instances of the generic class from Test #2, Question 3. The first instance should use Strings as the generic type, the second should Integers, and the third should use a type that you create yourself.

10. How did we use instances of each of the following classes with regards to the `Person/Student/StudentManager` examples: `Logger`, `Handler`, `Scanner`, `FileInputStream`, `BufferedInputStream`, `ObjectInputStream`, `FileOutputStream`, `BufferedOutputStream`, `ObjectOutputStream`. What is `Serializable` and how did we use it to store information about instances of class `Student`?

11. Create a JFrame that displays the information from a `StudentManager` so that each student has a checkbox. When the checkbox is checked, the student's information is sent, using the `println()` method, to the screen.

12. On the course website "Labs and Lectures" page under Week 8, you can find the files: `DemoCheckedAndUnchecked.java`, `UnexpectedNegativeException.java`, and `UnexpectedNegativeException.java`. Modify the first file to accomplish each of the following results, separately. It is not possible to accomplish them simultaneously.

    (a) The file compiles but does not run to completion.
    (b) The file compiles and runs, even though one of the methods throws an exception.
    (c) The file compiles and runs, but prints the call stack trace to the screen twice, at different parts of the program. In other words, the two traces should describe different points in the code.
    (d) The file compiles and runs, even though an exception is thrown from inside a call block.
    (e) The file compiles but does not run. However, between the moment when the last exception is thrown and the end of execution, the message "This is a message." appears on the screen.

13. We discussed the following design patterns in class: Observer, Singleton, Iterator, and Strategy. When would you want to use each pattern? Describe a situation where the Observer pattern would be useful. Do that again for each of the other patterns. Describe an alternative solution to the Observer pattern, the Iterator pattern, and to the Strategy pattern.

14. Modify the file "`Inherit.java`" from the website under the Week 7 Readings to demonstrate the meaning of the table on the first page of "`Practice Quiz 2`" under the Week 6 Readings. In other words, create modified versions of the file to check when Java shadows and overshadows for static/instance methods and static/instance variables.

15. Try the questions in the "`regex_practice`" file. You will be able to find it under the Week 11 Readings on the course website.

16. What is a floating point variable? What examples of floating point issues have we seen?

17. Complete a set of CRC cards for the Restaurant activity that we did during lecture. Looking at your cards, is it possible to deduce where each class is instantiated? As the user, where is the entry point into your program? If you move the main method, how does that impact your design?

18. Pretend that you are trying to explain JUnit to someone who knows nothing about it. What is an assertion? What are the different types of assertions and how do they work? What does it mean to pass a test? What is the difference between a fail and an error? What is a "unit test"? What are the three steps to running a test? How do you select test cases? Give examples to illustrate your explanations.