**Test 1 Review Lesson**

**Objectives**

* Review material covered in Moodle chapter 1, 2, 3, 4, to prepare for test 1
* The textbook chapters covered are chapter 1, 2, 3, 4, plus textbook Appendix D (page 260 ~ 267).

**Schedule**

1. Test #1 is a paper and pencil close book test, and it has 100 points, covering textbook chap 1 to 4, plus page 260 ~ 267 of Appendix D, and additional study materials that are provided in Moodle folder 1 to 4. **A letter-size paper (8.5 by 11 inches) with hand-written notes on both sides is allowed in the test**. This notes paper must be a complete piece of paper, and it cannot be a partial paper with tearing-apart lines. This notes paper needs to be turned in together with your test paper, and you cannot bring the notes out of the testing center. If you turn in the notes paper with tearing-apart lines, the score will be 0 for this test. Computer, phone, calculator, book is NOT allowed.

2. make sure you have finished and understood homework 1 to 4, before you go for test 1. In test 1, there will be two coding questions as described below:

* One coding question is similar to coding question #1 in file “***Test1StudyGuide.docx”***, which tests your ability to define a method, call a method with suitable plugged-in parameters, and make use of the return value of the method. These skills were covered in “Beginning Programming” class. One example file is ***HouseApplicationSample.java*** in zip file “UML-classDiagram.zip” in Moodle folder “chap 4”.
* The other coding question is similar to coding question #2 in file ***Test1StudyGuide.docx”***, which tests your ability to code and OOP class from scratch and then code an application class. This question is also similar to homework 4.
* In test 1, you need to write down the java code solutions in the test paper using a pencil. There is no computer for you to input source code in the test.

3. you DEFINITELY want to finish document “***Test1StudyGuide.docx***”, in order to prepare for test 1. The format of the test 1 is alike to the questions in this study guide. If you submit the answer of study guide to Moodle, you will have an opportunity of earning 10 bonus points. Let’s stay GREEN by submitting the electronic version ONLY, and DO NOT turn in any paper copy. The submitted study guide will be reviewed by the instructor, but NOT graded. If you have one question not answered in the study guide, 1 point will be taken off until all 10 bonus points are off.

4. even if you decide NOT to submit the answers for the study guide, you are **REQUIRED** to work on all the questions in the study guide. The questions in the study guide are the same type of questions that will show up in test 1. **It is very unlikely** that a student will achieve a passing grade without spending time on this study guide.

5. use Eclipse, work on coding exercises in section 1, 2, and 3 addressing the encapsulation principle of OOP from this link: <http://www.ntu.edu.sg/home/ehchua/programming/java/J3f_OOPExercises.html#show-toc>

For this link, first work on exercise 1.1 by yourself. You need to generate the source code based on the given UML class diagram for the Circle class. Then you can compare your answer with the answer given on the website. Then you need to create an application class called TestCircle, which has the main method, and in the main method, you create two objects of Circle class using two different constructors, respectively, and then you need to call some methods of the Circle class using the Circle objects you just create, and output the results of the method callings. Again, compare your TestCircle class with the answer given on the website.

Then you can work on exercise 1.2 to 1.8, and for each exercise, you need to first create the OOP class based on the given UML class diagram of the OOP class, and then for each OOP class, you need to create an application class that has the main method, and then create several objects of the OOP class type using different constructors, respectively, and then call some method of the OOP class using the objects you just created, and output the results of the method callings, like what you did in exercise 1.1 for class TestCircle.

For exercise 2.1 to 2.8, and 3.1 to 3.7, the first exercise in each section (section 2 and 3) has the java source code answer provided, and you can verify your work with the answer provided. Then you need to work on the rest of the exercises in each section, solely based on the UML class diagrams given, just like what you do for exercise 1.2 to 1.8 in in section 1.

Another link to study OOP is: <http://www.ntu.edu.sg/home/ehchua/programming/java/J3a_OOPBasics.html> , there are more explanations about the encapsulation principle of OOP.

6. **before** or **on the due day of test 1**, schedule a time slot in Testing Center in main building, and finish the close book test. In Moodle, click item ***Test 1 study guide drop box***, and you will see **the due day for test 1**. Before you can take any test in Testing Center, you have to show the staff your **OCCC student ID**. If you forget to bring or lost your student ID on the test day, the staff will NOT allow you to take the test, even after I make a call to them and try to verify your identity. Trust me, and I learn this in a hard way. Driver license or passport does NOT count in the Testing Center.

7. Plan about **two and a half hours (150 minutes)** to finish the test, or you can plan more hours if you think 2.5 hours is not enough. Testing Center does not time you when you take a test, and you sure can work on the test for as many hours as you can, as long as the facility is open. However, bathroom break is not allowed during the test.

The opening hours of Testing Center can be found in this link: <http://www.occc.edu/testing/test-center-hours.html>

You have to **start the test at least one hour before Testing Center is close**, otherwise, you will be denied to take the test.