**CSC 20 – Exam 2 Preparation**

This page is a guide for getting ready for the module exam. You should read all of the readings and work through all of the practice assignments. This is too much work to do at the last minute, so start early and get help if you need it. If you have time, you are encouraged to do additional similar CodeStepByStep problems for more practice. You may work alone or in groups.

**Writing classes**

Objective: Student should be able to write simple classes that model simple objects, including fields to represent the state of the object and methods to manipulate the state.

Read/Study: BJP 8.1, 8.2, 8.3.

Practice problems (you may see problems like this on exams):

<https://codestepbystep.com/problem/view/java/classes/Halloween>  
<https://codestepbystep.com/problem/view/java/classes/Circle>  
<https://codestepbystep.com/problem/view/java/classes/Date>  
[https://codestepbystep.com/problem/view/java/classes/Date%2DabsoluteDay](https://codestepbystep.com/problem/view/java/classes/Date-absoluteDay)  
[https://codestepbystep.com/problem/view/java/classes/Clock%2Dadvance](https://codestepbystep.com/problem/view/java/classes/Clock-advance)  
<https://codestepbystep.com/problem/view/java/classes/Student>  
[https://practiceit.cs.washington.edu/problem/view/bjp4/chapter8/e18%2DclassRectangle](https://practiceit.cs.washington.edu/problem/view/bjp4/chapter8/e18-classRectangle)

**Javadoc**

Objective: Student should be able to write and generate minimal javadoc for all public elements of a class including relevant @ tags.

Read/Study: <http://krovetz.net/20/formatting.html>.

Practice problems (you may see problems like this on exams):

<http://krovetz.net/20/m2/m2problems.html>

**Inheritance & Polymorphism**

Objective: Student should be able to simulate a given program that uses a hierarchy of classes, inheritance, and polymorphism; and should be able to write simple classes which override some inherited functionality.

Read/Study: BJP 9.1, 9.2, 9.3, and about [Critters](http://krovetz.net/20/m2/critters.html).

Practice problems (you may see problems like this on exams):

<https://codestepbystep.com/problem/view/java/critters/Rabbit>  
<https://codestepbystep.com/problem/view/java/critters/Frog>  
<https://codestepbystep.com/problem/view/java/inheritance/inheritanceMystery1>  
<https://codestepbystep.com/problem/view/java/inheritance/inheritanceMystery2>  
<https://practiceit.cs.washington.edu/problem/view/cs2/sections/polymorphism/HarryLarryMaryJerry>

**static keyword**

Objective: Student should be able to make a shared resource in a class using the static keyword and simulate its behavior when used by multiple objects of the class.

Read/Study:  
<https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html>  
<https://www.baeldung.com/java-static>

Practice problems