

# CSCI 3155: Homework Assignment 9

Due Thursday, December 3, 2009

For this assignment, you will again work with a **partner**. You will write up and turn in this assignment in pairs. Choose one of you to upload your write up to the moodle. Please name the file

`hw9-YourIdentiKey-YourPartnersIdentiKey.pdf`

You should also upload your Java source code named as follows:

`hw9-YourIdentiKey-YourPartnersIdentiKey.java`

Also, please include both of your PL-Detective user ids on your write-up and indicate which one should be considered the submission id. As always, you are welcome to discuss in larger groups. Just be sure to acknowledge those with which you discussed.

## Bookkeeping

**Exercise 1:** Indicate in a sentence or two how much time you spent on this homework, how difficult you found it subjectively, and what you found to be the hardest part. Any non-empty answer will receive full credit.

If you would like share something about yourself that I do not already know, please do so. And if your opinions have changed since the last submission, indicate one thing you like about the class so far and one thing you would change about it.

## Java Generics

For this set of exercises, you will need to have a good grasp of generics in Java. Please use any Java book or online resource to learn about generics in Java.

For any Java source code you write, you must make sure it compiles on the CSEL machines to be graded. Your write-up should include the snippets from your code, your test traces, and a discussion about them.

**Exercise 2: Skill 17.1 and 17.2.** Write a routine in Java that takes an array, the length of the array, and an element and returns the position of the element in the array. For example, given the array with values [2, 4, 6, 8] and element 6, the routine should return 2 (since counting from 0, 6 occurs at position 2 in the array). Your routine should use generics to enable your routine to be reusable for different element types. Be sure to test your code with a Java compiler before you submit it.

**Exercise 3: Skill 17.3.** Give a concrete example where you feel that parametric polymorphism is more suitable than inclusion polymorphism. Give a concrete example where you feel that inclusion polymorphism is more suitable than parametric polymorphism. Justify your answers.

**Exercise 4: Skill 17.4.** Write a generic “greater-than” function that (a) takes two objects as arguments, each of which has a “value” method which returns an integer; and (b) returns the argument whose “value” method returns the larger integer. Your generic function should constrain its type argument so that types without a “value” method cannot be used.

## Language Design

**Exercise 5: Synthesis.** This question will touch on most of the concepts that we have covered so far (notable omissions generics) and thus it may be a particularly good one to use in your preparation for the final. For this assignment, I will not limit the number of attempts to the PL-Detective. Use as many as you want as long as you use the link given below.

In this question, you will design the full semantics of the MYSTERY language. Use this link to submit programs to the PL-Detective:

<http://www-plan.cs.colorado.edu/diwan/plddesign.htm>

As part of this question, you will need to pick semantics for each dimension provided by the PL-Detective (e.g., scoping). The PL-Detective link given above has a link to a web page which describes all the options in detail.

For each of your choices (e.g., scoping), you must justify your choice using arguments based on the characteristics in Section 1.3 of text. We will grade you based on the quality of your arguments and to the extent you have thought about how language features interact (e.g., some of the choices in parameter-passing and type-passability may interact with each other). More specifically, we will see if your arguments (i) provide good and complete justification for your choices; and (ii) demonstrate that you understand the implications of your choices (e.g., if a given combination of choices requires many things to be checked at run time, your discussion should point that out).

If you questions about MYSTERY’s fixed semantics, you should figure these out by probing the system (much as you have been doing in the earlier assignments).

It is worth remembering that there is no single “perfect” language: if there was such a language then there would be no need for any other language. In other words, there is no single “right” answer that we are looking for. We are looking for evidence that you understand the concepts that we have covered so far, have thought about their interactions, and can provide good arguments to justify your choices.