

**CSCI 256 Principles of Programming Languages**  
**Spring 2009**  
**MWF 2:20 – 3:30 PM, HOSCI 123**

**Instructor:** Sun Chung  
[sbchung@moravian.edu](mailto:sbchung@moravian.edu)

**Office:** PPHAC 213, (610) 625-7786

**Office Hours:** MWF 9 – 10, T 10 – 11, and by appointment

**Course Description (from the Catalog):** A study of features of programming languages and of the methods used to specify and translate them. Topics covered include block structure, naming, procedure invocations and parameter passage, data types, data accessing, syntactic analysis, and correspondence of source language and object language constructs. Prerequisite: CS 244.

**Textbook (required):** Robert W. Sebesta, *Concepts of Programming Languages*, 8<sup>th</sup> edition, Addison Wesley, 2008.

**Goals include the following:**

By the time you complete the course, you will have done the following.

- Explore programming language design and implementation issues.
- Explore the various types of programming languages including imperative, functional, logic, and domain specific languages.
- Write programs in representative languages including Fortran, COBOL, LISP, and Prolog.

|                        |                         |           |
|------------------------|-------------------------|-----------|
| <b><u>Grading:</u></b> | Homeworks & Assignments | 25        |
|                        | Midterm 1               | 25        |
|                        | Midterm 2               | 25        |
|                        | <u>Final</u>            | <u>25</u> |
|                        | Total                   | 100       |

Please note the following:

1. Makeup tests will be given only for documented emergencies.
2. It is within the instructor's purview to apply qualitative judgment in determining grades for an assignment or for the course.

**Disability Accommodations:** If you have disabilities, please let me know and I will do my best to provide you with adequate accommodations. In addition, please note the following policy of the College: “Students who wish to request accommodations in this class for a disability should contact Mr. Joe Kempfer, Assistant Director of Learning Services for Disability Support, 1307 Main Street (extension 1510). Accommodations cannot be provided until authorization is received from the office of Learning Services.”

**Attendance:** I expect perfect attendance. However, if you have to miss class, you are responsible for all material covered that day, so be sure to check with someone in the class.

There is no penalty for up to two classes missed. With the third class missed, the grade will be lowered by one level (from A to B, from B to C, etc.). A final grade of F will be given in the case of six classes or more missed.

**Assignments:** There will be a number of programming assignments. In addition, there will be problem solving homework. Some of the assignments will be done individually, and others in pairs or groups. Programs must be handed in by 11:59 PM on the due date for full credit. A penalty of 25% will apply if the assignment is handed in the next day (and 50% for another day). After that, the assignment will not receive any points. A total of three grace days are given. A program must compile to be graded.

**Academic Honesty:** I feel very strongly that you do your own work. Copying another program will earn a zero for both the copier and the source. Merely changing the names of the variables or reordering a few instructions is not original work and will be considered a copy.

Legal ways to help others in the class include:

1. Talking in general terms on how to design the program (not specific instructions).
2. Helping extensively on “systems” questions (naming files correctly, using the compiler, printing out programs, etc.).
3. Finding syntax errors.
4. Debugging running programs (or crashing programs) by helping someone find the place where the error occurs and giving suggestions on how to fix the error. Giving suggestions means giving general ideas and does not include mailing your code, writing down instructions, or otherwise writing the program for your classmate.
5. Acting as a tutor by showing someone a different example and working with them on solving a different problem.

The syllabus is subject to change.