**CSCI 105 Fundamental Ideas in Computer Science**
**Spring 2009**
**TR 8:50 – 10 AM, REEVE 212**
**T 12:45 – 2:45 PM, PPHAC 114 (Lab)**

**Instructor**: Sun Chung
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)**: An introduction to several of the major ideas in the discipline of computer science. Emphasis is placed upon contributions which computer science has made to contemporary society. Topics covered include physical and logical aspects of computers, algorithms and problem solving, an introduction to programming, simple computer architecture, and additional topics central to the discipline. These topics are supplemented by laboratory exercises in which students create small programs or utilize existing programs.

**Textbook (required)**: Carl Reynonds and Paul Tymann, *Schaum's Outline of Principles of Computer Science*, 1st edition, McGraw-Hill, 2008.

**Goals include the following**:

By the time you complete the course, you will be able to

1. Describe the hardware and software aspects of computers and fundamental ideas in computer science.

2. Write computer programs using the Java language.

3. Use existing programs such as database management software.

4. Discuss social issues such as ethics, security, digital-divide, and open source software.

In addition, you will have been engaged in activities that satisfy the Quantitative Reasoning requirement for the Learning in Common (LinC) curriculum, including:

(a) Converting conceptual information into problems that can be solved quantitatively;

(b) Appropriate techniques for analyzing and solving quantitative problems that lead to the formation of a conclusion;

(c) Pictorial and graphical representation of data and data analysis, including those showing relationships among and/or between multiple variables;

(d) Significant use of appropriate technology as a tool for quantitative analysis;

(e) Formal, written interpretation of results and/or solutions of some problems.

| | | |
|---|---|---|
| **Grading**: | Quizzes & Assignments | 45 |
| | Midterm 1 | 15 |
| | Midterm 2 | 15 |
| | Final | 25 |
| | Total | 100 |

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.