

INTRODUCTION TO PROGRAMMING - 3 Credit(s)

YOUNG, CHA - SPRING 2011

Days: MW 06:00PM - 07:50PM Room: A 100

Course Begins: 1/17/2011 Course Ends: 5/11/2011 Last day to Withdraw: 4/8/2011

Lab Hrs: 2.00	Lecture Hrs: 2.00	PCS: 1.2	Articulated: Y	How:
IAI Core:	IAI Majors:			

Course Description:

Introduction to Programming is a study of documentation, logic, and flowcharting techniques used in typical business application programs. Structured design concepts that are general to many programming languages will be presented. In addition, the students will have hands-on application of those concepts using object-oriented programming language. They will create application programs which use the decision making, looping, and control break structures learned earlier in the course. This course may be repeated one time.

Course Prerequisite:

Credit or concurrent enrollment in CIS 110. Must place into MAT 095 or higher or permission of instructor.

Course Objectives:

Upon completion of this course, the student will be able to:

- Describe the purpose of program flowcharting
- Identify the various flowcharting symbols
- Identify and describe the three programming structures
- Describe the purpose of programming modules
- Explain how to code programming statements
- Define various data types
- Describe the differences between syntax errors, build errors, logic errors, and run-time errors
- Describe the purpose of various math operations and functions
- Demonstrate an understanding of the problem-solving capabilities of a high-level language
- Explain how files are processed in a high-level language
- Define array types and explain their uses in a high-level language
- Explain the importance of flowcharting a computer program
- Explain why there is a need for good documentation
- Describe from a programmer's point of view the advantage of structured design over non-structured programming methods
- Explain the need for good planning prior to coding a program
- Describe how a computer solves problems, what types of problems are suitable for computers, and the human role in computer applications
- Create a complete modular structured flowchart using computer software
- Design and create input and output record forms

- Design, code, compile, and execute complete high-level programs on the computer
- Debug programs until successful execution to ensure complete and accurate output

Course Outline:

- I. Overview of Computers and Logic
 - A. Developing computer programs
 - B. Machine language, assembly language, high level language, and fourth-generation languages
 - C. Steps in developing a program
 - D. Types of errors

- II. Introduction to Flowcharting and Pseudocode
 - A. Flowcharting symbols
 - B. How pseudocode may be used to represent program instructions
 - C. Character and numeric variables
 - D. Assignment statements

- III. Understanding Structure
 - A. Structured vs unstructured logic
 - B. The three basic programming structures
 - C. The priming read
 - D. Recognizing structured logic in flowcharts and correcting unstructured logic

- IV. Decision Making in Flowcharts
 - A. Evaluating Boolean expressions
 - B. Comparison operators
 - C. AND and OR logic
 - D. Writing AND and OR decisions for efficiency
 - E. Nesting decisions
 - F. Common errors made in decision logic

- VI. Looping in Flowcharts
 - A. Purposes for looping
 - B. Using a while loop with a loop control variable
 - C. Using a counter to control looping
 - D. Using a loop with a variable sentinel value
 - E. Using a for loop
 - F. Using a dountil loop
 - G. Nesting loops
 - H. Accumulating and counting

- VI. Introduction to High-Level Programming
 - A. Parts of a program
 - B. Output objects
 - C. Literals
 - D. Character, integer, and floating point variables
 - E. Arithmetic operators

- VII. Expressions and Interactivity

- A. Input objects
- B. Math expressions
- C. Type casting
- D. Named constants
- E. Formatting input and output

VIII. Making Decisions in Programming

- A. If statements
- B. If / else statements
- C. Flags
- D. Menus
- E. Logical operators
- F. The conditional operator

IX. Looping in Programming

- A. Increment and decrement operators
- B. While loops
- C. Counters
- D. Do while and for loops
- E. Break and continue statements

X. File Operations

- A. Declaring, opening, and closing file objects
- B. Using loops to read from and write to files

XI. Functions

- A. Defining and calling functions
- B. Function prototypes
- C. Passing data to functions by value
- D. Returning values from functions
- E. Local and global variables
- F. Passing data to functions by reference

XII. Arrays

- A. Accessing array elements
- B. Array declaration and initialization
- C. Processing array contents
- D. Passing arrays to functions
- E. Two- and higher dimensional arrays
- F. Arrays of strings

Special Needs Statement:

McHenry County College offers support services for students with special needs. It is the student's responsibility to meet with the Special Needs Coordinator and provide current documentation regarding his/her disability. Please stop in or call the Special Needs Department, room A-260, 815-455-8676, as soon as possible if you would like more information about the accommodations that are available. In addition, it is important for you to discuss those accommodations with me so you are able to fully participate in this course.

Academic Integrity:

As an educational community, McHenry County College values the pursuit of academic excellence and integrity. In accordance with this philosophy and Chapter 10, Act 5 of the 1994 Illinois Community College Act, academic dishonesty in any form, including cheating, plagiarism, and all other acts of academic theft, is considered intolerable. Appropriate sanctions, up to and including suspension from the college will be imposed by authorized College personnel.

Copyright Policy:

The College will maintain current procedures and guidelines to ensure that all staff and students comply with applicable copyright laws and other intellectual property protection laws. The College will encourage staff and students to engage in the development of intellectual property and facilitate ownership protections with respect to such development of intellectual property.

The College expects that staff and students will act responsibly and ethically in a manner consistent with all copyright laws and College copyright procedures and guidelines. This policy authorizes the College to adopt and maintain such procedures and guidelines necessary to ensure compliance with copyright laws and to facilitate ownership protection with respect to the development of intellectual property.

Student Code of Conduct and the Judicial Process:

Consistent with the McHenry County College mission is an expectation that students will govern themselves in terms of appropriate behavior with emphasis on self-respect and respect for others. It is the practice of the College to respect the properly exercised rights of its students. The College recognizes a student's rights within the institution to freedom of speech, inquiry and assembly; to the peaceful pursuit of education; and to the reasonable use of services and facilities of the College.

The College has adopted a Student Code of Conduct and judicial process in order to maintain a learning environment of respect, civility, safety, and integrity for all members of the College community.

Whenever possible, sanctions for violations of the Student Code of Conduct may be educational in nature. However, violations affecting the health and safety of members of the College community are deemed to be the most serious. Therefore, acts of violence, threats or dangerous behavior are most likely to result in a suspension from the college. Violations of the academic dishonesty policy may also result in suspension or expulsion from the institution and/or reduced or failing grade.

Children on Campus:

For the safety of children on campus, children (e.g., less than 16 years of age) are not permitted on campus unattended by a parent/guardian, except when they are attending classes offered by the College for children. The College requires that no children be allowed into a classroom/laboratory environment, including the Testing Center, Learning Center and computer labs, solely for the purpose of a parent/guardian to provide direct supervision of his/her child.

Teaching Schedule:

The scheduling of the activities and teaching strategies on this syllabus, but not the objectives or content, may be altered at any time at the discretion of the instructor.

INSTRUCTOR: Chuck Young
OFFICE: A124
OFFICE PHONE: 815-455-8544
OFFICE HOURS: To be announced

TEXTBOOK: Gaddis, Tony, Standard Version of Starting Out With C++, 6th Ed.

ATTENDANCE:

Computer programming is not like most other classes. An addition to studying the course material, you are learning skills you will need to apply. The pace of the course is intense.

It is important to attend every class. Due to the accelerated nature of this course and the relatively few class meetings we have, attendance is expected at every class meeting.

You are responsible for everything said and handed out in class. If it is necessary to miss a class, it is your responsibility to get the class notes from a classmate to cover the material you missed and contact the instructor for any missed handouts. If you miss a topic, it is highly unlikely you will understand any following topics until you make it up.

Up to ten points may be added to your point total based on attendance and class participation. Class participation involves preparing for class by reading assigned material; taking notes, listening to and participation in class lectures and discussions; and working on assigned materials during lab times. Students using chatrooms, Facebook or MySpace, playing Solitaire or other games, or displaying lack of interest or attentiveness by other means may be asked to leave.

GRADING:

The final grade will be based upon an accumulation of points from tests and application assignments. A tentative breakdown follows:

2 Midterm Tests @ 100 points	200
Final Test	100
Assignments & Programs	400
Quizzes	100
In-class exercises	60
Total points available	860

The grading scale to be used is as follows:

A = 90% of total possible points; B = 80 - 89%; C = 70 - 79%; D = 60 - 69%; E = 0 - 59%

There will be at least one extra credit assignment available worth 10 points. In addition, up to ten more points may be added to your point total at the end of the course based on attendance and class participation.

PROGRAMMING ASSIGNMENTS:

There will be ten programming assignments, worth 30-50 points apiece, for a total of 400 points. Assignments may include creating defining diagrams, flowcharts, pseudocode, printer spacing or screen charts, hierarchy charts, and C++ programs.

Assignments will be printed and submitted in class. This will include printouts of defining diagrams, pseudocode/flowcharts, programming code, and desk checks or printouts of the programs output. In addition, you will also need to turn in an electronic copy of the code you wrote. This will usually be done using a dropbox in Angel.

Each assignment will be due on the day scheduled for completion of the assignment. The assignments will normally be returned and discussed in the next week's class. For that reason, late assignments will be accepted up to one week late subject to a 10% penalty. Assignments later than that will still be accepted but will only receive 50% of the normal number of points.

I will waive the 10% penalty and grant an extension of time to turn in a program if you can show me that you have a non-working program that is substantially complete on the due date.

TESTS:

There will be two midterm tests and a final test for a total of 300 points (100 points each) as shown on the schedule.

Tests will be “open-book”; you may use your text book and any notes you have taken for reference during the test. Test questions will usually include a combination of multiple-choice, fill-in-the-blank, short essay, and problems. A problem may ask you to write a single flowchart or C++ instruction, a series of instructions, a structure, a module, or an entire program.

Although the schedule shows tests as covering just the most recent topics, bear in mind that some material from previous chapters will necessarily be included as well. This is due to the nature of program development, which does not allow you the luxury of forgetting material after you have used it once. This will occur particularly in problems involving writing several instructions.

ANGEL:

Angel is a Web-based, distance-learning program we will use extensively throughout this course.

You can access Angel from any computer connected to the Internet. With Angel, you will be able to

- Access course documents including the syllabus, assignments, and other materials
- Take quizzes and see the results immediately
- Turn in assignments automatically
- Send emails
- See announcements of class activities
- Check your grade

Angel's URL is <http://mchenry.angellearning.com>

QUIZZES:

There will be up to 10 quizzes given during the semester. Quizzes are all to be taken online and outside of class. You may access the testing software from the computer labs or from a home computer over the Internet. Each quiz is worth 10 points. These quizzes will all be on Angel and consist of objective questions (multiple choice, true-false, matching, etc.). Angel will automatically post your grades in the Check Grade section. These quizzes may be retaken as many times as you want until the due date listed; after that time, the quiz will disappear forever and may not be made up. You can review your quiz results on Angel anytime during the course.

COMPUTER LAB:

The Computer Labs in A127 and E108 are available for student use at regular hours throughout the term. Check the posted hours outside the lab for the exact schedule. Keep in mind that the lab assistants are not tutors. It is NOT their responsibility to train you on the computer or complete your assignments for you.

CHEATING:

All work must be your own! Any student caught cheating on an exam or assignment will receive a grade of 'F' for the course. Although students are encouraged to discuss programming assignments with one another, it is expected that each program submitted by a student will be his or her own work.

OFFICE HOURS AND LAB TIME:

Up to one hour per week of our class time will be open lab time. This is time for you to work on programs and for me to work individually with students to answer questions and try to fix "bugs" in the programs. I also schedule several office hours per week during which I can work with you.

SPECIAL NEEDS AND THE SAGE LEARNING CENTER:

I am a firm believer in "Equal Opportunity". If you have any type of disability (learning, physical, medical, etc.) and feel you need any type of special accommodations to participate fully in this class, please let me know. Any information shared will be held in strict confidence. You will also need to contact the Sage Learning Center.

The Sage Learning Center, a general tutoring and learning facility located in A247 near the Atrium entrance, offers students currently enrolled in credit academic courses assistance in a variety of subject areas. Comprehensive support is provided in math and in writing and English. Assistance is also provided, whenever possible, in many other disciplines. The SLC is staffed by professional and peer tutors who work on a scheduled drop-in and appointment basis with individuals and small groups. Computer-assisted instruction, study groups, assorted handouts, videos, audiotapes, CDs, and DVDs as well as a resource library are also available. The SLC is open from 8 a.m. to 7 p.m. Mondays through Thursdays and from 8 a.m. to 4:30 p.m. on Fridays.

GENERAL INFORMATION:

I reserve the right to alter or adjust the syllabus as needed. Any changes to the syllabus will be announced in class at any point in the semester. These changes will usually involve alterations to the schedule.

Course Schedule Spring 2011

PLEASE NOTE: READING ASSIGNMENTS SHOULD BE COMPLETED PRIOR TO THE SCHEDULED CLASS MEETING (EXCEPTION: 1ST CLASS MEETING).

Date	Topic	Reading
Week 1 (Jan. 17, 19)	Introduction to Course Programming Logic using Flowcharts and Pseudocode	From handouts
Week 2 (Jan. 24, 26)	Developing an algorithm Writing pseudocode Creating a flowchart	From handouts
Week 3 (Jan. 31, Feb 2)	Creating projects in Visual C++ (handout) C++ Basics	Chapter 1
Week 4 (Feb. 7, 9)	Identifiers, Variables, Assignments, Outputs in C++	Chapter 2
Week 5 (Feb. 14, 16)	Math expressions and inputs in C++	Chapter 3 (skip section 3.14, material on files for now)
Week 6 (Feb. 21, 23)	Test 1	
Week 7 (Feb. 28, Mar. 2)	Selection	Chapter 4 (skip material on Conditional Operator and Switch Statement, sections 4.15 and 4.16)
Week 8 (Mar. 7, 9)	More on Selection	
Week 9 (Mar. 14, 16)	Loops	Chapter 5
Week 10 (Mar. 21, 23)	More on Loops	
(Mar. 28, 30)	Spring Break	No classes
Week 11 (Apr. 4, 6)	Text files	Chapter 3: section 3.14 and from handouts
Week 12 (Apr. 11, 13)	Test 2 Functions in C++	Chapter 6
Week 13 (Apr. 18, 20)	More on Functions	
Week 14 (Apr. 25, 27)	Arrays	Chapter 7 (skip material on vectors, section 7.12)
Week 15 (May 2, 4)	More on Arrays	
Week 16 (May 9, 11)	Final Test at 6:00pm	