Metropolitan Community College

Course Syllabus **Spring 2012**

Course Title:	**Programming Fundamentals**
Course number:	**CSIS 123**
Credit Hours:	**3** *(2 lecture hours, 2 lab hours per week)*
Course Dependency:	**Prerequisite: MATH 40/40L or higher** **Required Co-requisite**: None
Class Schedule:	**Tuesdays 1/24/2011-5/15/2011 5:45 – 9:15 pm** **Room: Longview Business building, Room 205 (LV BU 205)**
Instructor:	**Dr. Tom Wheeler** **E-mail: thomas.wheeler@mcckc.edu** **Telephone: 816 604-5240** **Office Location: MCC-Business & Technology Room CC103**
Office Hours:	**Please contact me if you need assistance outside of the class. During the day I am normally at the Business & Technology Campus.**
Web Site:	http://n0gsg.no-ip.org/mcc

Course Description

This course provides an introduction to the principles of good program design and the characteristics common to all programming languages. The student will gain experience with writing code in C++ in preparation for more advanced course work (CSIS 223).

Textbooks and Materials

Textbook/s Title:	**Ed**	**Author**
C++ How to Program	**8th**	**Deitel & Deitel**

Useful Resources

- Practice examinations for the course can be accessed on the instructor's web site at:

  http://n0gsg.no-ip.org/mcc/csis123

  Please note that these will be built for the first time this semester, so they will become available as they are completed. You can take these exams as many times as you want to build your knowledge and skills in C++ fundamentals. The tests are automatically graded and give you instant feedback on how you're doing.

- It is strongly recommended that you install *Visual C Express* onto your own computer for home study, as this will facilitate completion of homeworks for this course. You may also use on-campus computers for homework completion if this option is not available to you. This software may be downloaded free of charge from Microsoft (http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-cpp-express).

- There are many excellent online sites for C++ programmers. You'll find links to these on the instructor's web page for this course.

Terminal Course Objectives (TCOs):

Following are the terminal objectives for this course. These are the skills that the student will be expected to attain during the progress of the course.

Metropolitan Community College is committed to the continual improvement of its curriculum and instruction to meet the needs of students and employers in a rapidly changing global economy. Students, faculty, and the College must all be actively involved to accomplish these objectives, as well as the objectives of this particular course.

Every class is to some extent a unique interactive experience, which may cause some variance within the stated objectives, in either content or level. Individual faculty, based on their experience and expertise, are encouraged to add objectives, as they deem appropriate, and to communicate these directly to the class. The outcomes of the course will depend on the design of the course, the quality of instruction, and the motivation and capabilities of the students, including time available for studying and the effectiveness of the effort.

Overall Course Objectives (MCC CSIS 123 Course Information Form)

1. State a brief history of programming.
2. Apply procedural abstraction, top-down algorithm design and step-wise refinement methods.
3. Describe and use various problem solving techniques and strategies.
4. Solve problems and develop algorithms using the control structure abstractions of sequence, selection and iteration, following a disciplined approach.
5. Apply tools to support program development, testing and debugging.
6. Code and document programs following a guideline.
7. Identify social and legal issues regarding the computing profession.

Clarified Course Objectives (These are an expansion and further explanation of the published CSIS 123 TCOs).

1. Use an Integrated Development Environment (IDE) to enter, compile, run and debug C++ programs.

2. Develop solutions to problems using structured programming techniques.

3. Given a task description, write a C++ program that completes the task.

4. Use C++ library functions to generate formatted output. (An example this would be creating a C++ program to generate a table of calculated values for a user.)

5. Write C++ source code that adheres to a set of published style rules for readability (indentation, placement of block delimiters, capitalization conventions for variable and function names, placement of useful comments, etc).

6. Effectively utilize C++ flow control structures (looping, if/else, switch, break, etc).

7. Demonstrate an understanding of C++ memory usage by correct utilization of calling techniques, arrays, strings, parameter passing methods, pointers, dynamic memory allocation, and other related C++ language features.

8. Given prescribed technical documentation guidelines, develop written reports and oral presentations with technical content.

Class Policies and Procedures:

Attendance

Each student is required to attend every lecture and laboratory session in which he or she is enrolled. Students are responsible for arriving before class begins and remaining for the duration of the course meeting. If you must be absent from class, please notify your instructor in advance to make arrangements for missed work. Students who miss more than two weeks of class will be dropped from the course. **Unexcused absence in excess of 1 week will result in a reduction of the final course grade by one letter**.

Homework Expectations

Homework is due at the beginning of class (xx00 UTC). ***Late homework is not accepted unless mitigating circumstances are present (documentation will be required).*** Homework carries the weight of one major exam (100 points) in the course. Failure to do homework will do severe damage to your grade. (UTC=Universal Coordinated Time, or Standard World Time.)

Homework Performance Standards

- Unless specifically noted, all homework is to be done individually. If you need help solving a problem, it is acceptable to ask the instructor or a fellow student for assistance as long as you do your own work.

- For problems involving calculations, all work must be shown. If a numerical answer is obtained without doing a calculation, state clearly that this is the case. For example: "By inspection, the potential is 25 volts."

- When showing work for numerical problems, all defining equations must be stated first. The last step in the problem will be substitution of values into the equations. For example:

  Given a distance traveled of 100 km, and an elapsed time of 2 h, calculate the average speed of the vehicle.

  $D = R\,T$ (Comment: The defining equation, Distance = Rate * Time, is stated.)

  $R = \frac{D}{T} = \frac{100km}{2h} = \underline{\underline{50km/h}}$ (Comment: Note that units are clearly displayed for the answer.)

- When a numerical answer is given, it must be boxed or underlined and have correct units attached. Programs that calculate answers such as this must also display the correct units in their output.

- For program listings, your name and course number (CSIS 123) must appear at the top of the output. Each function must have a comment header stating the function name, purpose, arguments, and return conditions. Each major idea within the code must be properly commented. (These expectations will be explained and demonstrated in class.)

- Programs that that lack appropriate division of functionality will receive a grade of zero (0). An example of such code would be a program performing all of its functionality within a single function such as `main()` when it would be more appropriate to divide the workload among several related functions.

(Homework Performance Standards are continued on next page)

Homework Performance Standards (Continued)

No credit will be given for any problems that have not been worked according to these instructions, or any additional instructions given by the instructor.

Laboratory Report Instructions

The handout "LabEx.pdf" (`http://n0gsg.no-ip.org/mcc/csis123/labex.pdf`) is an example of the format that is required for laboratory reports.

Make-Up Exams

No make up exams are given in CSIS 123 for any reason.

Course Grading Standards

There are two major exams, an unspecified number of quizzes given at random intervals, various homework assignments, laboratory assignments, and a comprehensive final examination given in the last week of the course. Your grade will be determined as follows:

2 Major Exams	200 points (22% each, 44% total)
Quizzes/Homework/Labs	100 points (22%)
Final Exam	150 points (34%)
	450 points total for course (100%)

Note: No makeup exams are given. All examinations will be announced at least 1 week prior to administration. All students must take the final exam.

A final letter grade is to be awarded to each enrolled student in accordance with the 4.00 grading system shown below:

Letter Grade	***Percent of Total Points***	***Grade Points***
A	90 – 100%	4.00
B	80 – 89%	3.00
C	70 – 79%	2.00
D	60 – 69%	1.00
F	Below 60%	0.00

Academic Integrity Policy

Each student is obligated to operate with utmost academic integrity. The College has explicit policies that address academic dishonesty and repercussions for participating in such devious and dishonest acts. The guiding principle for all questions of academic integrity is as follows: "Assume the most restrictive set of conditions, unless some exception is explicitly made by the instructor." Should you have any question about a particular academic strategy or behavior, ask the instructor of the course - do NOT rely on other instructors, agencies, friends, or members of the academic community, as there is great inconsistency in individual policies.

Forms of Behavior which Violate Academic Integrity

Cheating. Using any materials, devices or strategies that are not explicitly permitted by the instructor on any exam, assignment, activity or other method of assessment for a course. This includes, but is not limited to, looking at another student's exam, using phones or other communications systems to send messages during exams, taking pictures or images of exams, talking with others during exams, using the Internet to find information, or any other system of inappropriate help. Exams are to be measures of what you, as an individual, have learned.

Unauthorized Collaboration. Unauthorized collaboration means working together on projects, papers, exams or other graded items that were assigned to individuals by the instructor. All work in this course is to be completed individually unless otherwise specified.

Plagiarism. Plagiarism means taking the work of another and claiming it as your own. Often students mean to use an external source of information and just fail to cite it correctly. The website http://owl.english.purdue.edu/owl/resource/747/02/ shows how to correctly cite outside information using the MLA style.

Falsifying Data or Resource Materials. Inventing or misrepresenting data, ideas, information, quotations, concepts, information, sources, or any other materials for use in student work.

General Misconduct. Engaging or cooperating in activities such as changing grades, copying tests or test items, revealing to other students any information or items from exams or quizzes or any other forms of assessment, falsifying one's identity for any academic purpose or activity, violating any of the college's or district's policies (or any laws) in the performance of one's academic activities.

Consequences

Students should assume that the harshest consequences allowed under district policy, as outlined in the Student Handbook (http://www.mcckc.edu/pdf/exploreMCC/MCC_StudentHandbook.pdf), will be provided. Violation of Academic Integrity on any single part of a course will result at minimum a grade of zero (failing) for the involved element (lab report, homework assignment, quiz / exam, etc). All academic integrity violations will be documented by the College. It is incumbent on the part of the student to abide by any and all codes, traditions, rules, and guidelines for Academic Integrity.

Agreement

Attendance in the course constitutes agreement and subjection to the policies on Academic Integrity presented above and in the Student Handbook and Policies of this college and district.

Changes to Syllabus

The contents of this syllabus are subject to change with appropriate notice to the students.

Weekly Course Schedule:

Week	Date	Topics	What's Due?	Reading Assignment
1	1/24	The development process: Problem to Solution Introduction to the C/C++ environment. Scaleable applications. Objects. Documentation of program code; shop standards.		Ch1: pp. 1-32
2	1/31	C++ Data types and expressions; development of a simple application that utilizes user input; formatted output (printf); debugging introduction.	Lab #1	Ch2: pp. 37-55
3	2/7	C++ Data types and expressions, continued. C++ operators, continued. Structured decision making (if-then, switch) and elementary loop structures; debugging.	HW #1	Ch 4: pp. 101-138 Ch 5: pp. 152-185
4	2/14	Division of Program Tasks: Functions and parameter passing (without pointers). Debugging programs with functions.	Lab #2 HW#2	Ch 6: pp. 194-248
5	2/21	**Exam #1**		
6	2/28	Exam #1 review. Arrays and pointers. C++ memory models. Passing arguments to functions by pointer. Significance of array names. Dereferencing pointer arguments. Pointer best practices.		Ch 7: pp. 267-313 Ch 8: pp. 330-361
7	3/6	Arrays and pointers, continued. Special arrays – characters and strings. Practice with pointers and introduction to dynamic memory allocation methods.	HW #3 Lab #3	
-	**3/12**	**Spring Break – No class**		
8	3/20	Sorting routines. Library functions for characters and strings: sprintf(), strlen(), strcat(), etc. Prevention of unsafe operations.	HW #4	Internet Reading
9	3/27	Applications of arrays and strings: Sorting with `qsort()`; searching with `bsearch()`.	Lab #4	Ch 19: pp 724-740 (skim!)
10	4/3	Organizing data with structures and classes		Ch 9: pp. 379-407
11	4/10	Structures – applications		
12	4/17	**Exam #2**		
13	4/24	Files and streams – introduction.	Lab #5 HW #5	Ch 17: pp. 658-686
14	5/1	Serialization of structures and arrays. Development of larger projects (multiple files and libraries). Generating reusable code.		
15	5/8	Introduction to Object Oriented Programming (OOP). Terminology, memory models, typical practices. Review for final exam.	Lab #6	
16	**5/15**	**Final Examination**		

* This schedule is tentative and subject to change to meet the needs of the class members.

Note 1: Homework problems are on the instructor's web site at http://n0gsg.no-ip.org/mcc/csis123.
Note 2: LAB due dates: Labs are due by the end of class on each class meeting night.
Note 3: HOMEWORK due dates: Homework is due at the beginning of each class.

Getting to Know Each Other

Assumptions About You:

- You are probably working at least part time (and probably full-time) while attending school. Your time is very valuable and must not be wasted.

- You have the ability and motivation to be successful in a highly complex field (computer science) and you are willing to put in the time and effort to reach this goal.

- You possess a great deal of knowledge already as well as the ability to think critically. You will be building upon and adding to this knowledge.

About This Course:

- This is an introductory course that covers many important concepts. These concepts are essential foundation stones for building advanced knowledge in computer science. Each new idea builds upon the previous ideas, so regular attendance is very important.

- You will be very busy during this course. The average student will need to spend three to six hours outside of class each week to keep up. This requires dedication to the task and a willingness to stick with it when the going gets tough. Your instructor will do everything possible to help you succeed, but you are ultimately responsible for your own learning.

- If you are having trouble please ask the instructor for help. Don't put it off!

- This course will be student-centered. Traditional lecture will be used but it will be kept to a minimum.

About Me:

- I have more than 30 years of experience in the field of computer science and more than 20 years of teaching experience. I have high expectations for my students. At the same time, I realize that each one of us is an individual and brings a unique perspective and skill set to our class. My goal is to help you be as good as you are able in this subject.

- I believe that learning must be a lifelong process, and that one of the primary purposes of school must be to help all of us become better learners. Fundamental topics in technology are an enabler of lifelong learning.

- I expect to learn much along with you in this course.