**Catalog Description of CS 1120**

This is the standard Computer Science II course using the C# computer language. The emphasis is on designing and programming object-oriented computer solutions to problems, as well as on the data structures used for this purpose. An introduction to the analysis of algorithms is made. Students must register for both a lecture section and a laboratory section.

**Prerequisites**

**By Courses:** CS1110 - Computer Science I or equivalent with a grade of C or better (prerequisite); Math1220 or Math 2000 (co-requisite)

**By Topic:** Basic concepts of high-level language programming - conditional structures; looping structures; arrays; program logic - to solve problems; Basics of object oriented programming - be able to create and use elementary objects; C# language for both procedural and introductory object oriented programming; Basics of the software life cycle; Validating quality of software produced; Introductory sorting and searching algorithms; Algorithms for elementary problem solutions; Documenting programs effectively and efficiently.

**Objectives**

- To learn about various phases of the software life cycle.
- Understand the concepts of classes and object oriented programming.
- Understand the concept of recursion and structured programming.
- Learn basic mathematical techniques for analyzing algorithm complexity.
- Learn common data structures.
- Learn and use version control and documentation tools.
- Version Control/ Unit Testing/ Function Testing.
Performance Objectives

- Student will be able to write well-designed and well-documented C# programs.
- Student will be able to use recursive solutions for recursive problems.
- Student will be able to program various searching (linear, binary) and sorting (merge, quick) algorithms and be able to analyze their efficiencies.
- Student will be able to design, create design and use class and object hierarchies.
- Student will be able to use method or operator overloading.
- Student will be able to design complex data structures—including two-dimensional arrays (tables), linked list, stacks and queues.
- Student will be able to use collections or generics to perform operations on these data structures.
- Student will be able to work in a pair or team programming environment.

Text

Required

Grading
During the term there will be two mid-term exams and a final examination (see schedule below). Laboratory assignments will be given in the regularly scheduled laboratory. Pop-quizzes may be given at anytime in lab or lecture without prior notification. Your grade will be computed from your performance on these components using the following weights:

Midterm Exams: 30%
(Thursday, February 11, 2010, 8:00-9:00pm and Thursday, March 25, 2010, 8:00-9:00pm in rooms C226, C227 and C228)

Final Exam: 20%
(Friday, April 30, 2010, 8:00-10:00am in rooms C224, C226 and C227)

Labs: 40%
Participation, lecture quizzes & lab quizzes: 10%
PMT: 0%
(PMT: The week of April 19 in regular labs)
(Second-chance PMT: tentative: Friday, April 23, 2010, 03:00-05:00pm, C0224)

The following grading scale will be used.
A - 90; BA - 85; B - 80; CB - 75; C - 70; DC - 65; D - 60.

Missed Exams
If you miss an exam (Midterm Exams or Final Exam), the decision as to whether or not it is made up and how it is made up will be made on an individual basis. To be excused there must be significant circumstances beyond the student's control. Generally this will require documentation, such as a doctor’s note in the case of an illness. Normally, if your absence from an exam is excused, you will have to take a make up exam. Contact the instructor asking for a make up exam as soon as it is possible (if possible inform the instructor even before the exam that you will miss).

Laboratory
Students taking this course are required to register for a lab section. Lab grades are based on student performance on programming assignments and quizzes. Additionally, the Programming Skills Mastery Test (PMT) is given in lab.

Programming Assignments
Lab assignments will be given on a regularly scheduled basis. Many of these assignments will need to be worked on outside of the regular scheduled labs. Each assignment will have a due date/time. For
each day an assignment is late, 5% of the total possible points for the assignment will be deducted. If an assignment is more than 20 days late, it is no longer worth any points. Weekends and holidays are all counted when calculating lateness. No assignments may be submitted after 11:59 PM on the day preceding the last day of the classes, that is, Thursday, April 22. By this time all work should be complete and submitted.

**Quizzes**
There will be regular quizzes given in the lab. Additionally, pop-quizzes may be given at anytime in the labs or lectures without prior notification. If you miss a quiz for any reason, you will receive a 0 on it.

**PMT**
During the last lab session (the week of April 19-23) the semester, students will be given the Programming Skills Mastery Test (PMT). The test will consist of a short programming problem. Students must program the solution in an essentially complete and correct form in the allotted time. This problem must be solved within the allotted time to earn a passing grade in the course. Students that fail the PMT on their first attempt will be given the second, and last, chance with a different problem.

**Special Policies**
Producing competent programmers is a primary goal of this course, and therefore a minimum performance in lab is required for students to pass the course.

- You must pass the lab with at least 60% of the total possible lab points in order to pass the course regardless of exam scores.
- You should strive to complete all assignments. In order to pass the laboratory, you may have at most two assignments incomplete. Even if an assignment is so late that the credit would be 0, it can still satisfy the completeness policy if it is completed and submitted.
- You must pass the PMT to pass the course.

**Use of Electronic Devices**
You are expected to stay alert and pay attention to the directions/announcements in the class. Cellphones, PDAs, and other electronic devices should NOT be used during the lecture and should be turned-off. If available, please do bring your laptop to the class. Email checking or web-surfing of non-course related material is NOT permitted during the class. You may surf the web only when specifically told to do so. In order to maintain the integrity of the classroom and if I feel it is distracting you or others, I may ask you to turn-off your laptop / desktop or stop web-surfing.

**Incomplete Grades**
Please note that the incomplete grade - I - is intended for the student who has missed a relatively small portion of work due to circumstances beyond his/her control. In general, performance on work done must be at a level of C or better in order to qualify for an incomplete. An I grade will not be given to replace an otherwise low or failing grade in the class.

**Academic Honesty**
The following statement has been approved and distributed by the Western Michigan University Faculty Senate:

"You are responsible for making yourself aware of and understanding the policies and procedures in the Undergraduate and Graduate Catalogs that pertain to Academic Honesty. These policies include cheating, fabrication, falsification and forgery, multiple submission, plagiarism, complicity and computer misuse. [The policies can be found at http://catalog.wmich.edu under Academic Policies, Student Rights and Responsibilities.] If there is reason to believe you have been involved in academic dishonesty, you will be referred to the Office of Student Conduct. You will be given the opportunity to review the charge(s). If you believe you are not responsible, you will have the opportunity for a
hearing. You should consult with your instructor if you are uncertain about an issue of academic honesty prior to the submission of an assignment or test.

We also encourage you to browse http://osc.wmich.edu and www.wmich.edu/registrar to access the Code of Honor and general academic policies on such issues as diversity, religious observance, student disabilities, etc

Unless otherwise told, you may not bring aids to exams. Submission of another person's work in part or whole is not permitted. Learning can certainly occur with discussion of class material and assignments with other students, and we will be doing considerable collaborative activity, but at all times take care that you don’t represent the work of another as your own.

If you are copying another's work in part or whole, either by hand or electronically, you are going too far.

If two or more people are working so closely together that the outcomes, particularly on significant portions of computer programs, are essentially line-by-line the same in logical structure, they are going too far.

You should not give your completed work to someone else or accept another's completed work to "review or look at" in either hardcopy or electronic form. This too easily facilitates copying.

Easy availability of information, material, source codes, lecture notes etc on the Internet may make it possible to find solutions to your assignments on the Internet or elsewhere. It is okay to refer to those, understand them and use them to enhance your solutions, generate your own ideas etc. However, you must give proper and full credit to original authors of the work, if you include their ideas and/or solutions. Failing to do so is part of academic and professional dishonesty. It will not be tolerated in this class. Do not give in to temptations.

If you are found responsible for violation of academic honesty in the course, you will receive a penalty up to and including an E grade in the class in addition to a report to the Office of Student Conduct.

### Course Topics

#### Programming Topics

- **Coding Standards**
  - Declarations vs. definitions; Headers and code;

- **Compilation Process**
  - Preprocessor; Compilation; Linking

- **Classes**
  - What is a class? Terminology: objects, classes, instances; Examples of classes from modeling;
  - Member functions/methods; Member variables; Access functions (& pass by reference);
  - Constructors and destructors; Scope; Constant objects, functions, arguments; Static members;

- **Inheritance and Polymorphism**
  - Has-a vs. is-a designs; Subclassing; Access: public, private, and protected; Abstract base classes; Virtual functions; Late binding

- **Operator Overloading**
  - Why overload? How overloading works; Operators; Unary and binary operators; Comparison operators;

- **Stream I/O**
  - Stream classes and objects; Basic operations on streams; Read(), ReadLine(), etc.; Detecting the end of input; Using text file streams

- **Generics/Collections**

- **Tools**
  - Debugger tools

### Concepts
Program Design Using ADTs
Cohesion and coupling; Abstraction - functional and data; Problem statements and the noun/verb division; Producing a basic design document; Moving from design to code

Recursion
Recursive functions: fact and fib; Recursive algorithms: binary search; Reversing string; Problem-solving and traversing search spaces

Array Data Structures
Arrays of objects; Arrays of pointers to objects; Dynamic arrays

Linked Lists
Insertion: beginning, middle, end; Deletion: beginning, middle, end; LL as a modeling tool (trains, etc.)

Stacks
Uses and push/pop; Implementation as a LL; Sample Algorithms: postfix, delimiter match

Queues
Uses and enqueue/dequeue; Sample algorithms: palindromes, string to int conversion, simulation

Program Analysis
Analytic vs. experimental methods; Big-O concepts; Basic code analysis: loop structures; Analysis of bubble, insertion, and selection sorts; Analysis of binary search, towers of hanoi, quicksort, and mergesort

Algorithms
Linear and binary search; Bubble, insertion, selection, quick, and merge sorts; Postfix evaluation; Reversing linked lists; Palindrome recognition

Dates of Interest

01/11/2010 - Classes begin
01/18/2010 - Martin Luther King, Jr. Day (no classes)
02/26/2010 - Sprit Day (no classes)
03/01/2010 - Spring Break begins
03/08/2010 - Classes resume
03/22/2010 - Last day to withdraw (no refund, W recorded on the transcript)
04/26/2010 - Final examination week starts

Last updated on 01/11/2010