CSC 115
Foundations of Computer Science

Dr. Melissa Wiggins
MCC 306
(601) 925-3874
http://www.mc.edu/~mwiggins
mwiggins@mc.edu

COURSE CREDIT: 3 hrs. credit **PREREQUISITES:** High School Algebra

OFFICE HOURS: MWF 8-8:50; TR 9:25-10:40, T 1:00-2:15, W 1-3; *other hours by appt*

TEXT: *Computer Science Illuminated*, 2nd edition, Nell Dale and John Lewis
Exploration in Computer Science: A Guide to Discovery, 2nd edition, Mark Meyer
Course web site: http://cosmo.mc.edu/moodle - Computer Science - CSC 115MW

DESCRIPTION: Three hours of lecture per week. The class m may meet some of these times in the computer laboratory. This course is designed to introduce the student with no prior programming experience to the fundamentals necessary to study the science of computing. Topics include history of computing, computing as a tool and a discipline, machine level data representation, algorithms and problem solving, fundamental programming constructs & software design methodology, fundamental data structures, operating systems, net-centric computing, human-computer interaction, artificial intelligence, simulation, and social issues in computing.

OTHER MATERIALS: MCnet account and access to the Internet.

RATIONALE: This course is required of all majors and minors in Computer Science and Computing and Information Systems as well as various other degree programs on campus. The students must earn at least a C in this course to enter CSC 116.

LEARNING OBJECTIVES: After successfully passing this course, the student will have a strong breadth-first foundation upon which to continue in Computer Science. This background will serve them well in learning a new programming language.

EVALUATION: ***The instructor reserves the right to make adjustments as necessary.***

Tests: There will be a four exams worth 100 points each. (400 points)

Quizzes/Homework: There will be quizzes and/or homework assignments worth a total of 200 points.

Assignments: There will be problem solving assignments worth a total of 200 points.

Final Exam: There will be a comprehensive final examination given at the time specified by the college. This examination will be worth 200 points.

Grading Scale:

900-1000	**A**	600-699	**D**
800-899	**B**	0-599	**F**
700-799	**C**		

CLASS ATTENDANCE: The student is expected to attend classes. Regulations for class attendance are given in the Class Schedule. Remember in a MWF class 12 absences is an automatic **F**. **Three tardies counts as one absence in this class. (See Mississippi College catalog).** "For lesser numbers of absences, the student should expect a lowered grade in the course, with the maximum penalty of one letter grade for each week of absences (in a semester) or the equivalent. The calculation of the semester grade, including any penalty for absences, is the responsibility of the professor and may vary according to the nature of the course and the grading scale used. In some classes points will be deducted from the semester grade for unexcused absences; in others, the penalty may be built into the grading scale by means of frequent pop quizzes, grades for class participation and the like." Mississippi College Policy 2.10 Students should expect a penalty for absences as stated above.

MAKE-UP WORK & TESTS: Students are expected to take tests on the day they are assigned. However, it is the student's responsibility to contact the instructor in case of an emergency illness or death in the family **before** the test. At that time the student and instructor will agree on a time for the make-up exam. This time should be within 2 days of the missed test. Assignments are to be turned in on the day they are due!! All work is due at the beginning of the class period. Any work not turned in will lose 10% credit for each school day ***until the third day***. The due date at the beginning of class is day 1. No work will be accepted after the third day. Under no circumstances will work be accepted after the assignment has been graded and handed back in class. Laboratory work will be due at the end of each week's lab at which time a lab quiz will be administered. Exceptions to this may be made at the instructor's discretion. Mississippi College Policy 2.10 Students should expect a penalty for absences as stated above.

ACADEMIC INTEGRITY: This statement on academic honesty in computer science courses is an addendum to the Mississippi College policy 2.19 found at http://www.mc.edu/publications/policies/academic/219.wpd. In a computer science class individual effort is expected. Student misconduct not only includes cheating on tests, but also extends to copying or collaborating on programming assignments, projects, lab work or research unless otherwise specified by the instructor. Using other people's accounts to do your work or having others do your work is prohibited. Close proximity in lab does not mean collaboration is permitted. NOTE: Discussing logical solutions to problems is acceptable, exchange of code, pseudocode, designs, or procuring solutions from the Web, other texts, the Internet or other resources on or off campus is not acceptable.

First offense: grade of 0 for **all** parties involved unless the "guilty" party can be determined as well as any punishment deemed necessary under policy 2.19
Second offense: grade of F in the course as well as any punishment deemed necessary under policy 2.19

SPECIAL ACCOMMODATIONS: If you need special accommodations due to learning, physical, psychological, or other disabilities, please contact Dr. Buddy Wagner in the Counseling and Career Development Center. He may be reached by phone at (601) 925-3354 or by mail at P. O. Box 4016, Clinton, MS 39058.

DROPPING A COURSE: LAST DROP DATE - October 27

Students cannot withdraw after this date with a W (passing) unless the three following criteria are met:
- Extenuating circumstances (clearly outside the student's control)
- Passing the course at the time of withdrawal
- Does not have excessive absences at the time of withdrawal

NOTE: Dropping after the THIRD (3rd) WEEK will result in a grade of W appearing on your permanent record (transcripts). See http://www.mc.edu/publications/policies/academic/2.13.doc.

INCOMPLETE GRADES: An Incomplete may be given to a student who has been providentially hindered from completing work required in a course, provided that:
1. semester attendance requirements have been met;
2. most of the required work has been done;
3. the student is doing passing work; and
4. the student has made prior arrangements with the professor to complete the remaining work at a later date.

The grade of I must be removed promptly or it becomes an F; it cannot be removed by repeating the course.

TENTATIVE COURSE OUTLINE*

Day	Date	Course Topic & Reading Assignment	Other Course Assignments
1	Aug 23	Overview of Course and Policies	
2	25	Ch. 1 - The Big Picture	
3	28	Ch. 1	
4	30	Ch. 1	
5	Sept 1	Ch. 2 - Binary Values and Number Systems	
6	4	**No Class**	**Labor Day Holiday**
7	6	Ch. 3 - Data Representation	Lab 1 Due
8	8	Ch. 3	
9	11	Ch. 3 / Ch. 4 - Gates and Circuits	Lab 2 Due
10	13	Ch. 4	
11	15	Ch. 4 / Ch. 5 - Computing Concepts	Lab 3 Due
12	18	Ch. 5 / Ch. 6 - Problem Slvg & Alg Design	
13	20	Ch. 6	Lab 4 Due
14	22	***Exam 1 - Chaps 1-4 & Labs***	
15	25	Ch. 6	Lab 5 Due
16	27	Ch. 7 - Low-Level Programming Languages	
17	29	Ch. 7	Lab 6 Due
18	Oct 2	Ch. 8 - High-Level Programming Languages	
19	4	Ch. 8	Lab 7 Due
20	6	Ch. 9 - Abstract Data Types & Algorithms	
21	9	Ch. 9	Lab 8 Due
22	11	***Exam 2 - Chaps 5-8***	
23	13	Ch. 9	
24	16	Ch. 10 - Operating Systems	Lab 9 Due
25	18	Ch. 10	
26	20	Ch. 10	
27	23	**No Class**	**Fall Recess**
28	25	Ch. 10	
29	27	Ch. 11 - File Systems & Directories	****Last Drop Date** / Lab 10 Due
30	30	Ch. 11 / Ch. 12 - Information Systems	
31	Nov 1	Ch. 12	Lab 11 Due
32	3	Ch. 13 - Artificial Intelligence	

33	6	Ch. 13	Lab 12 Due
34	8	***Exam 3 - Chaps 9-12***	
35	10	Ch. 14 - Simulation & Other Applications	Lab 13 Due
36	13	Ch. 14 / Ch. 15 - Networks	
37	15	Ch. 15	Lab 14 Due
38	17	Ch. 16 - The World Wide Web	
39	20	Ch. 16	Lab 15 Due
40	22	**No Class**	***Thanksgiving Recess***
41	24	**No Class**	***Thanksgiving Recess***
42	27	Ch. 17 - Limitations of Computing	
43	29	Ch. 17	
44	Dec 1	***Exam 4 - Chaps 13-16***	
45	4	Ch. 17	**Dead Days**
46	6	Wrap-up and Review	**Dead Days**
47	11	***Comprehensive Final Exam 8:00-10:00 a.m.***	

*Instructor reserves the right to modify as necessary.

Course Objectives

Computing Systems and Computer Science
The student will be able to
•Describe the layers of a computer system.
•Describe the concept of abstraction and its relationship to computing.
•Describe the history of computer hardware and software.
•Describe the changing role of the computer user.
•Distinguish between system programmers and application programmers.
•Distinguish between computing as a tool and computing as a discipline.

Number Systems
The student will be able to
•Distinguish among categories of numbers.
•Describe positional notation.
•Convert numbers in other bases to base ten.
•Convert base-ten numbers to numbers in other bases.
•Describe the relationship between bases 2, 8, and 16.
•Explain the importance to computing of bases that are powers of 2.

Data Representation
The student will be able to
•Distinguish between analog and digital information.
•Explain data compression and calculate compression ratios.
•Explain the binary formats for negative and floating-point values.
•Describe the characteristics of the ASCII and Unicode character sets.
•Perform various types of text compression.
•Explain the nature of sound and its representation.
•Explain how RGB values define color.
•Distinguish between raster and vector graphics.
•Explain temporal and spatial video compression.

Gates and Circuits
The student will be able to
•Identify the basic gates and describe the behavior of each.
•Describe how gates are implemented using transistors.
•Combine basic gates into circuits.
•Describe the behavior of a gate or circuit using Boolean expressions, truth tables, and logic diagrams.
•Compare and contrast a half adder and a full adder.
•Describe how a multiplexer works.
•Explain how an S-R latch operates.
•Describe the characteristics of the four generations of integrated circuits.

Computing Components
The student will be able to
•Read an ad for a computer and understand the jargon.
•List the components and their function in a von Neumann machine.
•Describe the fetch-decode-execute cycle of the von Neumann machine.
•Describe how computer memory is organized and accessed.
•Name and describe different auxiliary storage devices.

•Define three alternative parallel computer configurations.

Problem Solving and Algorithm Design
The student will be able to
•Determine whether a problem is suitable for a computer solution.
•Describe the computer problem-solving process and relate it to Polya's How to Solve It list.
•Distinguish between following an algorithm and developing one.
•Apply top-down design methodology to develop an algorithm to solve a problem.
•Define the key terms in object-oriented design.
•Apply object-oriented design methodology to develop a collection of interacting objects to solve a problem.
•Discuss the following threads as they relate to problem solving: information hiding, abstraction, naming things, and testing.

Low-Level Programming Languages
The student will be able to
•List the operations that computer can perform.
•Discuss the relationship between levels of abstraction and the determination of concrete algorithm steps.
•Describe the important features of the Pep/7l virtual machine.
•Distinguish between immediate mode addressing and direct addressing.
•Convert a simple algorithm into a machine-language program.
•Distinguish between machine language and assembly language.
•Describe the steps in creating and running an assembly-language program.
•Convert a simple algorithm into an assembly-language program.
•Distinguish between instructions to the assembler and instructions to be translated.
•Describe two approaches to testing.
•Design and implement a test plan for a simple assembly-language program.

High-Level Programming Languages
The student will be able to
•Describe the translation process and distinguish between assembly, compilation, interpretation, and execution.
•Name four distinct programming paradigms and name a language characteristic of each.
•Describe the following constructs: stream input and output, selection, looping, and subprograms.
•Construct Boolean expressions and describe how they are used to alter the flow of control of an algorithm.
•Define the concepts of a data type and strong typing.
•Explain the concept of a parameter and distinguish between value and reference parameters.
•Describe two composite data-structuring mechanisms.
•Name, describe, and give examples of the three essential ingredients of an object-oriented language.

Abstract Data Types and Algorithms
The student will be able to
•Define an abstract data type and discuss its role in algorithm development.
•Distinguish between a data type and a data structure.
•Distinguish between an array-based implementation and a linked implementation.
•Distinguish between an array and a list.
•Distinguish between an unsorted list and a sorted list.
•Distinguish between a selection sort and a bubble sort.
•Describe the Quicksort algorithm.
•Apply the selection sort, the bubble sort, and the Quicksort to a list of items by hand.
•Apply the binary search algorithm.
•Distinguish between the behavior of a stack and a queue.

•Draw the binary search tree that is built from inserting a series of items.
•Demonstrate his/her understanding of the algorithms above by hand simulating them with a sequence of items.

<u>Operating Systems</u>
The student will be able to
•Describe the two main responsibilities of an operating system.
•Define memory and process management.
•Explain how timesharing creates the virtual machine illusion.
•Explain the relationship between logical and physical addresses.
•Compare and contrast memory management techniques.
•Distinguish between fixed and dynamic partitions.
•Define and apply partition selection algorithms.
•Explain how demand paging creates the virtual memory illusion.
•Explain the stages and transitions of the process life cycle.
•Explain the process of various CPU scheduling algorithms.

<u>File Systems and Directories</u>
The student will be able to
•Describe the purpose of files, file systems, and directories.
•Distinguish between text and binary files.
•Identify various file types by their extensions.
•Explain how file types improve file usage.
•Define the basic operations on a file.
•Compare and contrast sequential and direct file access.
•Discuss the issues related to file protection.
•Describe a directory tree.
•Create absolute and relative paths for a directory tree.
•Describe several disk-scheduling algorithms.

<u>Information Systems</u>
The student will be able to
•Define the roll of general information systems.
•Explain how spreadsheets are organized.
•Create spreadsheets for basic analysis of data.
•Define appropriate spreadsheet formulas using built-in functions.
•Design spreadsheets to be flexible and extensible.
•Describe the elements of a database management system.
•Describe the organization of a relational database.
•Establish relationships among elements in a database.
•Write basic SQL statements.
•Describe an entity-relationship diagram.

<u>Artificial Intelligence</u>
The student will be able to
•Distinguish between the types of problems that humans do best and those that computers do best.
•Explain the Turing test.
•Define what is meant by knowledge representation and demonstrate how knowledge is represented in a semantic network.
•Develop a search tree for simple scenarios.
•Explain the processing of an expert system.

•Explain the processing of biological and artificial neural networks.
•List the various aspects of natural language processing.
•Explain the types of ambiguities in natural language comprehension.

Simulation and Other Applications
The student will be able to
•Define simulation.
•Give examples of complex systems.
•Distinguish between continuous and discrete event simulation.
•Explain how object-oriented design principles can be used in building models.
•Name and discuss four parts of a queuing system.
•Explain the complexity of weather and seismic models.
•Explain the concept of embedded systems an give examples from his/her own home.
•Distinguish between two-dimensional and three-dimensional CAD systems.

Networks
The student will be able to
•Describe the core issues related to computer networks.
•List various types of networks and their characteristics.
•Explain various topologies of local-area networks.
•Explain why network technologies are best implemented as open systems.
•Compare and contrast various technologies for home Internet connections.
•Explain packet switching.
•Describe the basic roles of various network protocols.
•Explain the role of a firewall.
•Compare and contrast network hostnames and IP addresses.
•Explain the domain name system.

The World Wide Web
The student will be able to
•Compare and contrast the Internet and the World Wide Web.
•Describe general Web processing.
•Write basic HTML documents.
•Describe several specific HTML tags and their purposes.
•Describe the processing of Java applets and Java server pages.
•Compare and contrast HTML and XML.
•Define basic XML documents and their corresponding DTDs.
•Explain how XML documents are viewed.

Limitations of Computing
The student will be able to
•Describe the limits that the hardware places on the solution to computing problems.
•Discuss how the finiteness of the computer impacts the solutions to numeric problems.
•Discuss ways to ensure that errors in data transmission are detected.
•Describe the limits that the software places on the solutions to computing problems.
•Discuss ways to build better software.
•Describe the limits inherent in computable problems themselves.
•Discuss the continuum of problem complexity from problems in Class P to problems that are unsolvable.