SOUTHEAST MISSOURI STATE UNIVERSITY

Department(s): Computer Science **Course No.** CS 125 **(100 level)**

Revision:____ New: 2/2011___

Title of Course: Computer Programming Logic

I. Description and Credit Hours of Course:

A. Course Description for Undergraduate Bulletin:

Logic concepts in computer programming and how they can be used in several application environments. Two hours lecture, two hours lab. (3)

More Extensive Course Description for University Studies Handbook:

This course provides the mathematical and logical tools for understanding core programming concepts. After beginning with a review of elementary algebraic expressions, Boolean algebra (including logical operations such as “and” and “or”, and relational operators such as “less than” and “greater than”) are then covered. These concepts are then used in programming constructs such as selection and iteration (“looping”). Initially, these mathematical and logical concepts are discussed using language-independent computer algorithms; they are then applied to a variety of software applications, such as a simple high-level programming language, a spreadsheet (such as Microsoft Excel) and HTML (the language used for web programming).

II. Interdisciplinary Nature of the Course:

N/A (100-level course)

III. Prerequisite: MA 102 with a grade of CR or MA 095 with a grade of ‘C’ or higher or ACT Math score of 18-20 with MA 095 placement score of 14 or higher or ACT Math score of 21 or higher.

IV. Course Objectives:

Upon successful completion of this course, the student should be able to (corresponding University Studies objective numbers in brackets):

A. Understand the fundamental concepts of Boolean logic. [2]
B. Understand selection (if-then-else) and iteration (looping) programming constructs. [2]
C. Read and comprehend computer algorithms using the core programming constructs (selection, iteration, input/output and assignment). [3]

D. Be able to develop and document computer-based solutions to small problems in an introductory programming language using the core programming constructs, for use in an spreadsheet application using some of the core programming constructs and for simple web programming in hypertext markup language (HTML) using some of the core programming constructs. [1,2,3,6]

V. Expectations of Students:

Students are expected to:

1. Attend classroom lectures and course laboratory sessions.
2. Complete readings, exams, quizzes, laboratory exercises and homework assignments.
3. Demonstrate a working knowledge of course objectives through satisfactory performance on quizzes, exams, laboratory exercises and homework assignments.
4. Properly document solutions as specified in laboratory exercises and homework assignments.

VI. Course Outline:

(University Studies objectives shown in brackets)

Topic	**Lecture Hours**	**Lab Hours**
A. Boolean Logic and Algebraic Expressions [2]	3	2
B. Selection Constructs [2]	2	3
C. Looping Constructs [2]	3	4
D. Assignment Construct [2]	1	1
E. Input/Output Constructs [2]	1	2
F. Reading Computer Algorithms [3]	2	1
G. Applying Computer Algorithms in Non-Formal Areas [6]	2	2
H. Developing Solutions in a Programming Language [1]	2	2
I. Documenting Solutions in that Language [3]	1	2
J. Applying Programming Solutions in Non-Formal Areas [6]	1	2
K. Developing Solutions in a Spreadsheet Application [1]	1	1
L. Documenting Solutions for that Application [3]	1	1
M. Using Spreadsheet Programming in Non-Formal Areas [6]	1	1
N. HTML and Applying Logic in it [6]	2	2
O. Developing Solutions in HTML [1]	3	2
P. Documenting HTML Solutions [3]	1	1
Q. Applying HTML Solutions in Non-Formal Areas [6]	1	1
R. Exams (not including two hours for final)	2	0
Total	**30**	**30**

VII. Textbook(s) and Course Materials:

The textbook will be custom-built from multiple sources by Pearson Education, as follows:

Prelude to Programming: Concepts and Design, 5th edition by Stewart Venit and Elizabeth Drake, Addison-Wesley, 2010, Chapters 0-5 and Appendix D.

The Practice of Computing using Python by William F. Punch and Richard Embody, Addison-Wesley, 2010, Chapters 1 and 2.

Microsoft Office Excel 2010 Comprehensive by Robert T. Grauer, Keith Mulberry and Mary Anne Poatsy, Pearson Education, 2010, Chapters 1, 2, 7 and 11.

Programming the World Wide Web, 6^{th} edition by Robert W. Sebesta, Addison-Wesley, 2010, Chapters 1 and 9.

No additional course materials will be required.

VIII. Basis for Student Evaluation:

Grade percentage breakdown:

Two One-Hour Exams	25%
Final Exam	25%
Laboratory Exercises (at least 15)	20%
Homework Assignments (at least 7)	20%
Quizzes (at least 10)	10%

- There will be two one-hour exams plus a final. Most exam problems will be in a problem-solving format (e.g. writing a short code fragment given the description of its function).

- The quiz format will be instructor-specific and is intended to supplement understanding of the individual course lectures. One possible format would be to have the students receive the quiz at the beginning of a lecture, work on it during the lecture, and turn it in at the end of the period.

- The laboratory exercises should be completed during laboratory time and are intended for solving small problems, such as modifying source code provided to them and observing what occurs. There will be at least 15 lab assignments; the precise number will in part depend on whether the lab meets for one or two hours at a time.

- Homework assignments are intended to be done outside of class or lab. Seven assignments taking an average of 4-5 hours per assignment would be a typical format. The type of assignment would vary on the section of the course being covered. For instance, a homework assignment might be to write a program in the high-level programming language being used.

Course Grade Division:

90-100	A
80- 89	B
70- 79	C
60- 69	D
0- 59	F

IX. Justification for Inclusion in the University Studies Program:

A. Justification by University Studies Objectives:

Objective 1: Demonstrate the ability to locate and gather information
Emphasis: Significant Emphasis
Content: In order to apply logic using a high-level programming language, spreadsheet application and HTML, the student will need to be able to use their textbook and other information resources (e.g. online help tools) in order to reference the syntax details needed to program in each of these areas.
Teaching Strategies: Students will be taught in classroom lectures on the basic information resources required to locate and gather information in order to apply logic in programming using a high-level language, a spreadsheet application and HTML.
Student Assignments: There will be homework assignments that require programming in the high-level programming language, spreadsheet application and HTML, and using resources to gather and use information which will aid the student in doing the required programming. For instance, a homework assignment might require the writing of a program in the high-level programming language being used, and information resources would help the student in finding the appropriate syntax to use.
Student Evaluation: Correctness of the constructs used in the software artifact developed in the homework assignments. This will be determined as part of the rubrics developed for assessing student performance in each of these assignments.

Objective 2: Demonstrate capabilities for critical thinking, reasoning and analyzing.
Emphasis: Significant Emphasis
Content: Throughout the course, the student will be required to apply logic to Boolean algebra and core computer statements in order to create computer programming algorithms. Typically, a student needs to be able to take a problem that needs to be solved and be required to analyze it and use that analysis to design and implement an appropriate algorithmic solution.
Teaching Strategies: Several class lectures will be devoted to the various core programming components (Boolean logic, selection constructs, looping constructs, the assignment construct, and input/output constructs) and their use in computer algorithms.
Student Assignments: There will be laboratory exercises that focus on the analysis of each of the individual core computer statements. Also, synthesis of these components will be performed in at least two homework assignments in the form of algorithms to write. There will also be exam problems related to both the analysis and synthesis of algorithms using Boolean algebra and core computer statements.

Student Evaluation: Proper analysis of the use of each of the core components in the lab exercises, correctness of the logic of software artifact developed in the homework assignments, and accurate answers to problems posed on exams involving these components. This will be determined as part of the rubrics developed for assessing student performance for each related lab exercise, homework assignment and exam question.

Objective 3: Demonstrate effective communication skills.
Emphasis: Significant Emphasis
Content: Students will learn how to document all the types of software artifacts covered in this course (algorithms, programs in the high-level programming language, the use of core computer logic components in the spreadsheet application, and HTML programs). Such documentation is essential so that other people besides the original creator of a programming artifact can understand its workings in order to do such things as program maintenance or the upgrading of functionality.
Teaching Strategies: For each of the four types of software artifacts covered in the course (algorithms, programs in the high-level programming language, the use of core computer logic components in the spreadsheet application, and HTML programs) there will be at least one class lecture devoted to teaching how to document that type of artifact. This will include where to place such documentation, how to determine the right amount of documentation to use, and self-documenting by such techniques as using meaningful variable names.
Student Assignments: For each type of artifact, there will be at least one lab exercise devoted to having the students document that artifact. Also, there will least one homework assignment involving the documentation (as well as development from a problem description) of each type of artifact. There will also be exam problems testing the ability to document such artifacts.
Student Evaluation: Quality and proper quantity of the documentation performed on the artifacts. This will be determined as part of the rubrics developed for assessing student performance for each related lab exercise, homework assignment and exam question.

Objective 4: Demonstrate an understanding of human experiences and the ability to relate them to the present.
Emphasis: Not Emphasized
Content: N/A
Teaching Strategies: N/A
Student Assignments: N/A
Student Evaluation: N/A

Objective 5: Demonstrate an understanding of various cultures and their interrelationships.
Emphasis: Not Emphasized
Content: N/A
Teaching Strategies: N/A
Student Assignments: N/A
Student Evaluation: N/A

Objective 6: Demonstrate the ability to integrate the breadth and diversity of knowledge and experience.
Emphasis: Some Emphasis
Content: After learning how use core computer logic constructs to apply Boolean algebra to the creation of computer algorithms, the student will apply them to providing formal solutions to non-formal descriptions taken from various disciplines. The problems will be posed in the form of *customer requirements*, the computer science version of "word problems". For instance, suppose a person has three flash drives of different sizes and wishes to put as much music as possible on each of these three drives. Often, a person would just try to estimate this without any formal computations. However, with the logic constructs taught in this course, a person could write a small program in a high-level language which when used could provide more accurate estimates.
Teaching Strategies: There will be at least one lecture involving the formalization of non-formal problem statements from other disciplines for basic algorithms and for each of the three application areas: the high-level programming language, spreadsheets and HTML (5 lectures total).
Student Assignments: There will be at least one related laboratory exercise for algorithms and each of these three types of applications (6 lab sessions in all). For both algorithms and each of the application areas, there will be at least one homework assignment which involves the development of a solution from a non-formal problem description using the core computer logic constructs initially learned for the development of computer algorithms (Objective 2). (These assignments will also be documented, as discussed in Objective 3.) There will also be exam problems that test this knowledge.
Student Evaluation: The accuracy of translating a non-formal description to a formally-defined problem solution will be assessed. This will be determined as part of the rubrics developed for assessing student performance for each related lab exercise, homework assignment and exam question.

Objective 7: Demonstrate the ability to make informed, intelligent value decisions.
Emphasis: Not Emphasized
Content: N/A
Teaching Strategies: N/A
Student Assignments: N/A
Student Evaluation: N/A

Objective 8: Demonstrate the ability to make informed, sensitive aesthetic responses.
Emphasis: Not Emphasized
Content: N/A
Teaching Strategies: N/A
Student Assignments: N/A
Student Evaluation: N/A

Objective 9: Demonstrate the ability to function responsibly in one's natural, social and political environment.
Emphasis: Not Emphasized
Content: N/A

Teaching Strategies: N/A
Student Assignments: N/A
Student Evaluation: N/A

X. Background:

The instructor should have a general background in computer science and sufficient teaching knowledge and ability for the instruction of programming logic, a high-level programming language, a spreadsheet application and HTML programming.

XI. Class Size:

The maximum optimal class size is 25 students, although facilities are currently available to teach this course to up to 40 students per section.