

STANDARDIZED COURSE OUTLINE

SECTION I

SUBJECT AREA AND COURSE NUMBER: CSC 210

COURSE TITLE: C Programming

COURSE CATALOG DESCRIPTION:

Topics in this popular mid-level software development language include advantages of C programming, portability, data representation, storage utilization, registers, input/output macros, looping, decision making, table processing, macros, bit manipulation and structured design. Students, who should have some familiarity with personal computers, will write and execute sophisticated programs with business applications as well as develop proficiency in mini-applications software programs such as a small word processor. *Formerly listed as CIS 207, not open to students who have successfully completed CIS 207.*

LECTURE HOURS PER WEEK: 3

CREDIT HOURS: 3

LAB HOURS PER WEEK (if applicable): n/a

PREREQUISITE(S): CSC*101 with at least one high-level programming language such as FORTRAN, OBOL or Visual BASIC; or permission of the instructor

SECTION II

A. SCOPE:

This course will cover the following topics in detail: Introduction to Programming, variables, strings, conversion characters, input functions, header files, if statements, looping, conditional statements, compound relational operators, arrays, structures, pointers, reading and writing to and from files, and functions.

B. REQUIRED WORK:

Will vary by instructor. Students will be expected to do all required readings, assignments, tests, and quizzes as outlined by their instructor.

C. ATTENDANCE AND PARTICIPATION:

Regular attendance, assignment submission timeliness, promptness and class/lab participation will be expected. Instructors will include specific attendance and participation policies requirements in their class syllabi.

D. METHODS OF INSTRUCTION:

Methods may include any of the following: lecture, lecture/discussion, small group, collaborative learning, experimental/exploration, distance learning, student presentations, computer demonstrations, or use of technologies such as audio-visual materials, and computer laboratory equipment. Emphasis will be on hands-on computer exercises and problems.

E. OBJECTIVES, OUTCOMES, and ASSESSMENT

Students' grades will be based on achievement of learning the objectives and outcomes listed below as measured by the instructor's methods of assessment:

LEARNING OBJECTIVES	LEARNING OUTCOMES	ASSESSMENT METHODS
To demonstrate an understanding of:	Student will:	As measured by:
Introduction to Programming, The Programming Process, printf function, and escape sequences	a) Explain and apply the programming process b) Write a simple program c) Use a compiler d) Debug a sample program	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Variables, strings, and conversion characters	a) Declare variables b) Access values stored in variables c) Display variable values using conversion characters	<ul style="list-style-type: none"> • Same as above
Scanf function, header files, and if statements	a) Prompt a user for input b) Store input in variables c) Use data in variables in IF statements to create a branching program	<ul style="list-style-type: none"> • Same as above
Introduction to looping, while loops, do-while loops, and for loops	a) Explain and apply various types of loops	<ul style="list-style-type: none"> • Same as above
Conditional statements and compound relational operators	a) Use a conditional statement to replace a simple If Else statement b) Use compound relational operators to combine conditions	<ul style="list-style-type: none"> • Same as above
Arrays	a) Implement basic arrays b) Sort data in an array using a variety of different sorting algorithms	<ul style="list-style-type: none"> • Same as above
Structures	a) Implement structures b) Describe the importance and relations of structures to databases	<ul style="list-style-type: none"> • Same as above
Pointers and dynamic memory allocation	a) Explain the relationship between pointers and arrays b) Convert a program written with arrays to a program written with pointers	<ul style="list-style-type: none"> • Same as above
Reading and writing to and from files	a) Use files to store permanent data b) Access stored files to read data from them c) Take data from a file and store it in a variable	<ul style="list-style-type: none"> • Same as above

Functions	<ul style="list-style-type: none"> a) Break up a program into functions b) Observe how different functions in a single program can be written by different people working independently c) Combine multiple functions into one working program 	<ul style="list-style-type: none"> • Same as above
-----------	---	---

F. TEXT(S) AND MATERIALS:

An appropriate C Programming Text, such as:

Text: C++ Game Programming (*current edition*)

Author: Dawson

Publisher: Course Technology

G. INFORMATION TECHNOLOGY:

This course is an information technology course and will require extensive computer lab time both for teaching and performing assignments. Students will require network accounts with access to a basic C compiler as well as file storage space.