OUTCOMES BASED LEARNING MATRIX

Course: CTIM 372 – Advanced Programming in C++ 3 credits/4 contact hours
Department: Computer Technology and Information Management

Description:
This course is a continuation of Programming in C++. The course covers Object Oriented Programming concepts: classes, member functions, stream I/O, inheritance, pointers, arrays, linked lists. Hands-on programming assignments will be completed using the college's workstation computers and Visual Studio C++ compiler.

Lecture: 2 Hours Laboratory: 2 Hours

Prerequisite: Programming in C++ or permission of Department.

While completing the table below, remember that the individual outcomes you list in the first column should answer this question: **What must the learner know and be able to do at the end of the course?** Items in the third column should answer the question: **How do we know?** The second column is where teachers can be most creative; it's for pedagogy. Each rectangle in column one should contain just one outcome; the corresponding rectangles in columns two and three, however, may contain more than one item. Using the code at the end of the matrix, indicate the core competencies being strengthened by the outcomes activities and the assessment tools.

Approved by CTIM Department: September 2003

COURSE OUTCOMES	OUTCOME ACTIVITIES	ASSESSMENT TOOLS
At the end of this course, the student will: 1. Understand advanced use of arrays in C++ programming.	1. a) Sort arrays of different data types using a variety of sorting algorithms. (R,CCT) b) Code a linear and a binary search of arrays. (R,CCT) c) Process multidimensional arrays and implement Matrix Algebra. (R,CCT) d) Use nested `for` loops to process multidimensional arrays. (R,CCT) e) Use arrays to process strings. (R,CCT)	Exercises from the end of the chapter. (R,W,QS,IG,CCT,QT) Design & develop computer programs given the problem definitions. (R,W,QS,IG,CCT,QT) Course Objective test. (R,W,QS,IG,CCT,QT)

COURSE OUTCOMES	OUTCOME ACTIVITIES	ASSESSMENT TOOLS

2. Understand functions in C++ programming.	2. a) Use string and character functions to process strings. (R,CCT) b) Use prototypes, arguments, and returned values to design functions. (R,CCT) c) Use local and global variables. (R,CCT) d) Pass by value and by reference. (R,CCT) e) Pass an array to a function. (R,CCT)	Referenced Above

COURSE OUTCOMES	OUTCOME ACTIVITIES	ASSESSMENT TOOLS

3. Understand the concept of pointers in C++ programming.	3. a) Declare a pointer and use the indirect operator. (R,CCT) b) Use pointers for indirect addressing. (R) c) Do pointer arithmetic. (R,CCT) b) Use an array name as a constant pointer. (R,CCT) c) Use pointers as function arguments and return value. (R,CCT) d) Use Dynamic Memory Allocation. (R,CCT)	Referenced Above

COURSE OUTCOMES	OUTCOME ACTIVITIES	ASSESSMENT TOOLS

4. Understand structured variables.	4. a) Declare a structure and a structure variable. (R,CCT) b) Use the dot access operator to access structure variable members. (R,CCT) c) Use an array of structures to define a table. (R,CCT) d) Access structure members using pointers . (R,CCT) e) Code a `switch` statement to select 1/N cases. (R,CCT) f) Use the break statement to prematurely exit from a loop. (R,CCT) g) Use the continue statement to end the current iteration of a loop. (R,CCT)	Referenced Above

COURSE OUTCOMES	OUTCOME ACTIVITIES	ASSESSMENT TOOLS
5. Understand classes and objects	5. a) Define a class and instantiate objects. (R,CCT) b) Code class member functions, accessors, mutators, constructors, and destructors. (R,CCT) c) Overload constructors. (R,CCT) c) Use `for` loops to process arrays. (R,CCT)	Referenced Above

COURSE OUTCOMES	OUTCOME ACTIVITIES	ASSESSMENT TOOLS

6. Manipulate objects.	6. a) Overloading operators. (R,CCT) b) Using inheritance and polymorphism. (R,CCT) c) Designing a linked list. (R,CCT) d) File I/O using the ios class. (R,CCT)	Referenced Above

*Try to express an outcome as an infinitive phrase that concludes this sentence: **At the end of the course, the students should be able to . . ..** Finding the line between too general and too specific can be difficult. In an English Composition course, for instance, it is probably too general to say, "The student should be able to write effective essays." It is probably too specific to say, "The student should be able to write an introductory paragraph of at least 50 words, containing an attention-getting device, an announcement of the narrowed topic, and an explicit thesis sentence." Just right might read, "The student will write introductions that gather attention and focus the essay."

**Indicate the Core Competencies that apply to the outcomes activities and assessment tools: Critical Thinking (CT); technology skills (TS); oral communications (OC); quantitative skills (QS); reading (R); writing (w). Approved by the CTIM Department – September, 2003

**Indicate the Core Competencies that apply to the outcomes activities and assessment tools: Critical and Creative Thinking (CCT); Integrative Learning (IG); oral communications (OC); quantitative skills (QS); reading (R); writing (w). Approved by the CTIM Department – September, 2015