

COURSE SYLLABUS

Course Title:	Object-Oriented Programming with C++	Date submitted:	Spring 2014 (AAC: 14-28)	
Department:	Business and Technology			
Curriculum:	Computer Information Systems			
Course Descriptors: Make certain that the course descriptors are consistent with college and Board of Trustees policies, and the current course numbering system.	Course Code: (eg. ACC 101)	CSC* 215	Prerequisites: C- or better in Programming Logic and Design with Visual Basic (CSC*126)	
	Course Type:	X		
	A: Clinical B: Lab D: Distance Learning I: Individual/Independent L: Lecture N: Internship M: Seminar P: Practicum U: Studio X: Combined Lecture/Lab Y: Combined Lecture/ Clinical/Lab Z: Combined Lecture/Studio	Elective Type:	G	Corequisites: None
	AH: Art History E: English FA: Fine Arts G: General HI: History HU: Humanities LA: Liberal Arts FL: Foreign Language M: Math S: Science SS: Social Science	Credit Hours:	4	
	Developmental: (yes/no)	No		
	Lecture:	3		
	Clinical:	0		
	Lab:	1		
	Studio	0		
	Other:	0		
Contact Hours:	TOTAL:	4		
Class Maximum:	24	Other Requirements: None		
Semesters Offered:	F/Sp			
Ability Based Education (ABE) Statement:	At Tunxis Community College students are assessed on the knowledge and skills they have learned. The faculty identified the General Education Abilities critical to students' success in their professional and personal lives. In every class, students are assessed on course abilities, sometimes program abilities, and, in most classes, at least one General Education Ability. Students will receive an evaluation of the degree to which they have demonstrated or not demonstrated that General Education Ability.			
Catalog Course Description:	Introduces students to the C++ programming language. Techniques for solving problems with both numerical and non-numerical applications will be explored, incorporating rules of syntax, expressions and operators. Sequential and direct-access file processing are discussed. Concepts and examples of data types, recursive & virtual functions, arrays, pointers, vectors, strings, namespaces, data abstraction with classes, objects, overloading, inheritance, and data structures are presented.			
Topical Outline: List course content in outline format.	<ol style="list-style-type: none"> 1. Introduction to C++ 2. Input/output 3. Functions 4. Parameters and Overloading 5. Structures and Classes 6. String, Pointers and Dynamic Arrays 			

	<ol style="list-style-type: none"> 7. Recursion 8. Inheritance 9. Polymorphism 10. Data Structures
<p>Outcomes: Describe measurable skills or knowledge that students should be able to demonstrate as evidence that they have mastered the course content.</p>	<p>Upon successful completion of this course, the student will be able to do the following:</p> <p>COURSE:</p> <ol style="list-style-type: none"> 1. demonstrate an understanding of basic object-oriented programming concepts such as classes, inheritance, and polymorphism 2. demonstrate an understanding of basic data structures used to solve programming problems 3. design, write and execute modular programs on a personal computer using object-oriented programming techniques 4. design, write and execute modular programs on a personal computer using data structures <p>PROGRAM: <i>(Numbering reflects Program Outcomes as they appear in the college catalog)</i></p> <p>PROGRAMMING</p> <ol style="list-style-type: none"> 1. solve computer-related problems 2. apply the use of the Program Development Life Cycle 3. practical knowledge of a high-level programming language such as Java, C++ or Visual Basic <p>PROGRAM: CIS: COMPUTER PROGRAMMING OPTION</p> <ol style="list-style-type: none"> 3. apply object-oriented programming techniques in a variety of programming languages 6. apply programming skills and constructs to develop large-scale programs and applications <p>GENERAL EDUCATION: <i>(Numbering reflects General Education Outcomes as they appear in the college catalog)</i></p> <ol style="list-style-type: none"> 2. Critical Analysis/ Logical Thinking - Students will be able to organize, interpret, and evaluate evidence and ideas within and across disciplines; draw reasoned inferences and defensible conclusions; and solve problems and make decisions based on analytical processes. <p>Demonstrates: Identifies the issue(s); formulates an argument; explains and analyzes relationships clearly; draws reasonable inferences and conclusions that are logical and defensible; provides support by evaluating credible sources of evidence necessary to justify conclusions.</p> <p>Does Not Demonstrate: Identifies few or no issues; formulates an argument without significant focus; provides an unclear explanation of analysis and relationships; drawing few reasonable inferences and conclusions that are illogical and indefensible; provides little to no support using credible sources of evidence necessary to justify conclusions.</p>
<p>Evaluation: List how the above outcomes will be assessed.</p>	<p>Assessment will be based on the following criteria:</p> <ol style="list-style-type: none"> 1. Programming assignments and projects 2. Quizzes 3. At least one programming project will be uploaded to the student's ePortfolio

<p>Instructional Resources:</p> <p>List library (e.g. books, journals, on-line resources), technological (e.g. Smartboard, software), and other resources (e.g. equipment, supplies, facilities) required and desired to teach this course.</p>	<p>Required: Computer Lab</p> <p>Desired:</p>
<p>Textbook(s)</p>	<p>Refer to current academic year printout.</p>