MASTER COURSE OUTLINE

Prepared By: Arthur Wanner/Tom Willingham | Date: September 2017

COURSE TITLE

Advanced Programming with C++

GENERAL COURSE INFORMATION

Dept.: CS	Course Num: 132	(Formerly:)
CIP Code: 11.0201	Intent Code: 11	Program Code: 515
Credits: 5		
Total Contact Hrs Per Qtr.: 88		
Lecture Hrs: 22	Lab Hrs: 66	Other Hrs:
Distribution Designation: General Elective (GE)		

COURSE DESCRIPTION (as it will appear in the catalog)

This course expands on the fundamentals covered in CS& 131. Students learn to develop procedural and object-oriented C++ programs of increasing complexity using advanced algorithm design and data abstraction techniques. Language concepts explored include pointers, dynamic memory allocation, classes, inheritance, polymorphism, operator overloading, templates, and exception handling.

PREREQUISITES

CS& 131

TEXTBOOK GUIDELINES

Textbook and materials to be determined by CS Faculty

COURSE LEARNING OUTCOMES

Upon successful completion of the course, students should be able to demonstrate the following knowledge or skills:

1. Create C++ programs of intermediate to advanced complexity using procedural and object-oriented design techniques, structured problem solving and data abstraction.
2. Utilize recursion as a problem solving technique and define simple recursive functions.
3. Use pointers and indirection to dynamically allocate memory, arrays and objects.
4. Solve problems involving 2 dimensional arrays or vectors.
5. Sort an array by implementing the Bubble Sort and Insertion Sort algorithms.
6. Implement the binary search algorithm to efficiently search a sorted array.
7. Compose object-oriented programs using concepts such as classes, objects, composition, inheritance, and polymorphism.
8. Explore advanced techniques such as operator overloading and templates.
9. Create custom exception classes and use exception handling to throw and catch common errors.
10. Use pointers to explore and implement the linked list data structure

INSTITUTIONAL OUTCOMES

IO1 **Communication**: Students will be able to communicate clearly and effectively within a workplace context

IO2 **Quantitative Reasoning**: Analyze and solve computational problems using a modern program language

COURSE CONTENT OUTLINE

1. 2D Arrays / 2D Vectors
2. Recursion
3. Pointers & Dynamic Memory Allocation
4. Classes & Objects
5. Inheritance & Polymorphism
6. Operator Overloading & Templates
7. Exception Handling
8. Linked Lists

DEPARTMENTAL GUIDELINES *(optional)*

_____ _____

DIVISION CHAIR APPROVAL **DATE**