ITM 312 SYLLABUS

ITM 312 Introduction to Systems Software Programming (C++ Programming)

Hours: 3 credit hours / 60 contact hours; 30 hours lecture, 30 hours lab

Instructor: TBD

Textbook, title, author, and year: *Starting Out With C++, 8th Edition*, Tony Gaddis, 2013

Specific course information

a. **Catalog description:** Introduces basic concepts of systems programming. Students learn to apply basic programming concepts toward solving problems, create source files and implement header files, work with and effectively use basic data types, abstract data types, control structures, code modularization and arrays. Students will be introduced to object paradigm including, classes, inheritance, and polymorphism applications.

b. **Prerequisites:** None

Specific goals for the course

a. Program Educational Objectives:
   2. Perform requirements analysis, design and administration of computer and network-based systems conforming to policy and best practices, and monitor and support continuing development of relevant policy and best practices as appropriate.

b. Course Outcomes:
   - Introduces basic concepts of systems programming.
   - Students learn to write computer programs in C++ that solve problems; solving problems with programs is the focus.
   - Students learn to create source files and implement header files, work with and effectively use basic data types, compile source code into binary executable files, and understand the use of project management.

c. Course student outcomes:
   Upon completion of this course the student should be able to do the following:
   - Design an algorithm to solve a problem.
   - Translate that algorithm into a C++ program.
   - Debug, test, and verify that the program solves the original problem.
   - Apply Object-Oriented-Programming methods in developing a program.
   - Use C++ standard libraries and classes.
   - Analyze programs for efficiency; change techniques and data structures to improve efficiency.
   - Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions **(ABET Computing Criterion 3.1)**
   - Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline **(ABET Computing Criterion 3.2)**
   - Identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems **(ABET IT Criterion 3.6)**

Topics to be covered

a. Using Introduction to Computers and Programming, Introduction to C++ Expressions and Interactivity, Making Decisions
b. Loops, Making Decisions
c. Loops
d. Functions
e. Arrays
f. Arrays Part 2
g. Introduction to Classes
h. Continuation of Classes
i. Pointers
j. Inheritance, Polymorphism, and Virtual Functions
k. Recursion
l. Exceptions, Templates, and the Standard Template Library (STL)