Model Syllabus for TCSS 480
Version: April 2012
(Approved May 11, 2012)

Catalog Description

Study and comparison of at least three programming languages, including at least one imperative, one logic, and one functional language, with regard to data structures, operations, notation, and control flow. Examines programming paradigms, implementation strategies, programming environments, and programming styles. Prerequisite: a minimum grade of 2.0 in TCSS 342.

Preconditions

- Translate problem descriptions into mathematical formalisms.
- Develop and implement programs involving the fundamental programming constructs (variables, types, expressions, assignment, simple I/O, conditional and iterative control structures, functions and parameter passing, structured decomposition).
- Apply object-oriented design concepts such as inheritance, composition, encapsulation, abstraction, method overloading, method overriding, exception handling, and scope.
- Correctly employ programming language features by reading and interpreting the associated published API documentation.
- Define what a data abstraction is.
- Compare and contrast time/space characteristics of different implementations of the same data abstraction.
- Select data abstractions, structures, and implementations that a developer would use in solving larger problems and defend the appropriateness of these choices.

Course Student Learning Goals (to be added to syllabus handed out to students)

- Design, implement, document and debug programs in at least three programming languages, including at least one imperative and one declarative language.
- Identify the differences among declarative, imperative, and structured programming language paradigms.
- Identify the paradigm (or paradigms) to which a programming language belongs, given sufficient documentation about the language syntax and constructs.
- Select appropriate programming paradigms in which to implement programs that solve specific problems, and defend the appropriateness of these choices.

CSS Degree Student Learning Outcomes that this course contributes to (to be added to syllabus handed out to students). Note that the use of the term *outcome* here instead of *goal* is simply for purposes of integration with ABET and has no other semantic import.

a. an ability to apply knowledge of computing and mathematics appropriate to the discipline;

b. an ability to analyze a problem, identify and define the computing requirements appropriate to its solution;

c. an ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;

h. recognition of the need for, and an ability to engage in, continuing professional development;

i. an ability to use current techniques, skills, and tools necessary for computing practice.

UWT Student Learning Goals that this course contributes to (to be added to syllabus handed out to students)

[Below contains the complete list. Remove those that do not apply to this course]

Inquiry and Critical Thinking
Students will acquire skills and familiarity with modes of inquiry and examination from diverse disciplinary perspectives, enabling them to access, interpret, analyze, quantitatively reason, and synthesize information critically.

Communication/Self-Expression
Students will gain experience with oral, written, symbolic and artistic forms of communication and the ability to communicate with diverse audiences. They will also have the opportunity to increase their understanding of communication through collaboration with others to solve problems or advance knowledge.