

TCSS 305 Master Syllabus

Version: April 2011

(Approved: 27 May 2011)

Catalog Description

Provides a practicum in program design and development. Programming practice on a medium-scale, object-oriented application, consolidating prior programming principles and expanding knowledge of application design. Prerequisite: a minimum grade of 2.0 in either TCSS 143 or CSE 143.

Preconditions

- 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).
- Develop and implement programs that use each of the following: arrays, objects, strings.
- Implement a low-complexity program (3 or more interacting classes) that includes the use of interfaces and/or abstract classes, and polymorphism given some design guidance.
- Apply object-oriented design concepts such as inheritance, composition, encapsulation, abstraction, method overloading, method overriding, exception handling, and scope appropriately to the implementation of a program.
- Provide formal documentation for a program using Javadoc comments.
- Use single and multidimensional arrays and linked lists as well as basic data structures (Lists, Stacks, Queues) in the implementation of a program.
- Use a provided class, given only its API.

Student Learning Outcomes (to be added to syllabus handed out to students)

1. Design, implement, document and debug a medium complexity program with an object hierarchy that includes interfaces and/or abstract classes and a graphical user interface, given some guidance on the design.
2. Design and implement unit tests for a medium complexity program with an object hierarchy that includes interfaces and/or abstract classes.
3. Utilize modern software engineering tools (e.g., IDEs, static checkers, unit testing frameworks, revision control systems) during the implementation of a medium complexity program.
4. Correctly employ programming language features by reading and interpreting the associated published API documentation.

CSS Degree Student Learning Outcomes that this course contributes to (to be added to syllabus handed out to students)

- 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)

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.

Topics covered

- Tools and Tool Usage
 - Basic IDE Usage (Eclipse)
 - Basic Version Control (Subversion in Eclipse)
 - Basic Debugging (Eclipse Debugger)
 - Basic Automated Testing (JUnit, EclEmma)
 - Basic Style Checking (CheckStyle)
 - Basic Static Checking (FindBugs, PMD)
 - Documentation (Javadoc)
- Programming and Design Concepts
 - Classes and Objects
 - Inheritance and Interfaces
 - Collections (Lists, Sets, Maps)
 - Java 1.5+ Features (Enhanced For Loops, Enumerated Types, Generics)
 - Graphical User Interface Programming (Events, 2D Graphics, Animation with Timers and Threads)
 - Model-View-Controller Design Pattern
 - Observer Design Pattern
 - Basics of Design by Contract

Additional information

For the last several years, TCSS 305 has been taught without a required textbook and students have been encouraged to use a Java reference book (or the Internet) for their language reference needs. The main reasons for this are the wide variety of topics in the course and the continual evolution of the modern software engineering tools used in the course; no appropriate textbook exists at present.