CS 162 – Introduction to Computer Science II

Catalog Description: Basic data structures. Computer programming techniques and application of software engineering principles. Introduction to analysis of programs. Lec/lab.

Credits: 4 **Terms Offered:** All

Prerequisites: EECS 161 or CS 161

Courses that require this as a prerequisite: CS 261, CS 275

Structure:
On Campus: Three 50-minute lectures per week, and one 110-minute laboratory per week.

Ecampus: Term totals: This course combines approximately 120 hours of instruction, online activities, and assignments for 4 credits (30 hours of online instruction, 10 hours of online participation, 2 hours of online quizzes, 30 hours of offline reading/study, 15 hours of offline homework/lab assignments, 30 hours of offline programming projects, and 3 hours of proctored exams).

Instructors: Weng-Keen Wong, Jennifer Parham-Mocello, Terry Rooker

Course Content:
- File I/O
- Object-oriented programming principles
- Program design, debugging and testing
- Algorithm analysis
- Recursion
- Sorting and searching
- Linear data structures
- Debugging and testing

Learning Resources: One or more of the following:
- *Absolute C++*, Savitch, Addison-Wesley
- *C/C++ Programmer's Reference*, Schildt, Osborne McGraw Hill
- *Big C++*, Horstmann and Budd, Wiley
- *Programming and Problem Solving with C++* (6th edition), Dale/Weems
- Additional online resources.

Measurable Student Learning Outcomes:
At the completion of the course, students will be able to…

1. **Design** and **implement** programs that require
    a. multiple classes and structures
    b. hierarchies of classes that uses inheritance and polymorphism

c. an understanding of abstraction, modularity and separation of concerns

2. **Construct** and use basic linear structures (arrays, stacks, queues, and various linked lists) in programs, and be able to describe instances appropriate for their use.
3. **Classify** moderately complicated algorithms in these complexity classes: O(1), O(log n), O(n), O(n log n), and $O(n^2)$.
4. **Develop** test-data sets and testing plans for programming projects**.**
5. **Produce** recursive algorithms, and **choose** appropriately between iterative and recursive algorithms.

Evaluation of Student Learning: (Percentages are approximate)

- 30% Programming and other homework assignments
- 20% Labs
- 30% Midterms / Quizzes
- 20% Final

Students with Disabilities:

Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS). Students with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 737-4098.

Link to Statement of Expectations for Student Conduct, i.e., cheating policies
http://oregonstate.edu/studentconduct/http:/%252Foregonstate.edu/studentconduct/code/index.php

Revised: Spring 2014