

# CS 161 – Introduction to Computer Science I

**Catalog Description:** Overview of the fundamental concepts of computer science. Introduction to problem solving, software engineering, and object-oriented programming. Includes algorithm design and program development.

**Credits:** 4                      **Terms Offered:** All

**Prerequisites: Enforced:** MTH 112\* or (MPT score  $\geq 33$ ) or (MPAL  $\geq 61$ )

**Co-requisites: Enforced:** MTH 112 (can be taken as a prereq)

**Courses that require this as a prerequisite:** CS 162, CS 271, CS 352

## **Structure:**

On-Campus: Three 50-minute lectures and one 110-minute lab 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:** Jennifer Parham-Mocello, Joseph Jess

## **Course Content:**

- Identifiers and primitive data types
- Assignment, arithmetic, logical, and relational operators
- Expressions and statements
- Flow of control: selection, repetition, recursion
- Functions/parameter-passing including call-by-value and call-by-reference
- 1- and 2-dimensional arrays, strings, and other structured data types
- Pointers
- Error Handling
- Debugging

**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++* (5<sup>th</sup> 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. various control statements involving selection and repetition
  - b. expressions with variables, constants, function calls, pointers, and arithmetic/relational operators with mixed data
  - c. arrays, strings, and other data structures
  - d. library functions and programmer-defined functions with parameter-passing by value and by reference
  - e. define and use classes and objects
2. **Debug** programming syntax and run-time errors.
3. **Produce** recursive algorithms
4. **Describe** and **apply** basic software engineering design principles and software quality factors.

### **Evaluation of Student Learning:** (Percentages are approximate)

- 40% Programming and other homework assignments
- 10% 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: Winter 2014