UNIVERSITY OF NEBRASKA AT OMAHA
COURSE SYLLABUS

Department and Course Number	CSCI 1620
Course Title	Introduction to Computer Science II
Course Coordinator	Patrick Cavanaugh
Total Credits	3
Repeat for Credit?	No
Date of Last Revision	Jan 17, 2017

1.0 Course Description Information

1.1 Catalog description:

Advanced topics in programming; topics in data representation and manipulation, data structures, problem solving and algorithm design. This course has a required laboratory component; students must register for a laboratory section when enrolling in lecture.

1.2 Prerequisites of the course:

CIST 1400 and MATH 1930 or MATH 1950 (with a grade of "C-" or better)

1.3 Overview of content and purpose of the course:

This is the follow up course to CIST 1400 Intro to Computer Programming. This course will focus more on the object oriented aspect of programming using the Java programming language. The basics of algorithms, including basic sorting techniques will be discussed, as well as introduction to data structures and file I/O and GUI programming. In the laboratory sections, students will gain additional weekly hands-on practice with the programming skills and techniques described in the corresponding lecture.

1.4 Unusual circumstances of the course.

None.

2.0 Course Justification Information

2.1 Anticipated audience / demand:

This course is intended for freshman/sophomore students in computer science and related disciplines as an intro level course.

2.2 Indicate how often this course will be offered and the anticipated enrollment:

Three to four sections of this course will be offered every semester, with an additional distance section. Anticipated enrollment per section is approximately 30 students.

2.3 If it is a significant change to an existing course, please explain why it is needed:

This change incorporates a mandatory lab component to this course. Student surveys indicate that the bulk of students enrolled in CSCI1620 wish there were a dedicated lab associated with the course, and they request more opportunities to practice programming outside of the large homework projects typically associated with CSCI1620. Data from OIE indicates a potential skills gap that arises in 1620. Requiring lab in CSCI1620 for all students ensures there is adequate time for hands-on practice, and that all students going into CSCI3320 do so on a more even skill level. As with the new format of CIST1400, the new lab will not carry a separate course number and will appear as a separate lab section with the CSCI1620 course number.

3.0 List of performance objectives stated in learning outcomes in a student's perspective:

In completing this course students will be expected:

- To fluently create and use Java classes
- Understand and utilize basic sorting techniques
- Develop algorithms and determine their efficiency
- Create and use basic data structures
- Perform file I/O in Java

4.0 Content and Organization Information

4.1 List the major topics central to this course:

- Data Representation (2 hours)
    - Basic Data Types
    - Computer Storage of Basic Data Types
- Classes & Objects (8 hours)
    - Abstract Data Types
    - Object Implementation
    - Object Instantiation
    - Inheritance
    - Polymorphism
    - Packages
- Recursion (2 hours)
    - Space, Time and Computability Considerations
    - Base Case vs. General Case
    - Method Stack
    - Recursion vs. Iteration
    - Recursive Backtracking
- Basic sorting techniques (5 hours)
    - Definition of sorting
    - Basic algorithms for sorting
        - Insertion Sort
        - Selection Sort
    - Recursion and sorting
- Exceptions (3 hours)
    - Stack Trace
    - Catching Exceptions
    - Exception Hierarchy
    - Errors vs. Exceptions
    - Check Exceptions and Unchecked Exceptions
    - Throwing Exceptions
    - Creating Exceptions

        - Assertions
    - Array List (2 hour)
        - Data Structures
        - Array List implementation
    - Generics (2 hour)
        - Type Parameters
        - Primitive Wrappers
        - Comparable Types
        - Creating Generic Methods and Classes
    - Stacks & Queues (2 hours)
        - Implementation and use
        - Priority Queues
        - Double Ended Queues
        - Circular Queues
    - Linked Lists (4 hours)
        - Singly Linked
        - Circular
        - Doubly Linked
    - Files (4 hours)
        - Data Hierarchy
        - Streams
        - I/O classes
        - Writing to Files
        - Reading from Files
        - Object Serialization
    - GUI Applications (4 hours)
        - Components creation
        - Components usage
        - Event Handling

5.0 Teaching Methodology Information

5.1 Methods:

The main method of instruction for this course is through instructor lectures. These consist of discussion of topics as well as in class examples and exercises. Additionally students will be required to demonstrate their understanding through programming projects and exams. Weekly laboratory sessions provide students with directed practice on the skills of programming relevant to the current lecture content through hands-on problem solving and discussions, which are informed by formative assessments.

5.2 Student role:

The student’s role in the course is to attend the lectures, take notes, and participate in discussion as well as complete programming assignments and exams. Students are expected to take an active role in laboratory sessions to clarify their understanding of course content. Students will complete both homework assignments and weekly labs to deepen their understanding.

6.0 Evaluation Information

6.1 Describe the typical types of student projects that will be the basis for evaluating student performance:

Exams: Students' mastery of course concepts will be assessed through a midterm exam and a cumulative final exam.

Labs: Weekly labs will be completed in the laboratory sections. Approximately 15 labs will be completed by students to assist their mastery of programming skills. Labs focus on programming in the small; that is, solving multiple-related problems in order to gain practice with specific skills each week. Students must have their labs checked-off by course staff no later than the start of the next lab period.

Homework: Approximately 5-8 integrative homework assignments will be completed during the semester. These assignments provide larger problems for students to solve that allow assessment of cross-cutting skills and synthesis of multiple course concepts. The typical project will be focused on the current topic of discussion, requiring the student to demonstrate their understanding as well as apply this knowledge to new situations. All projects will require the student to write a program given strict criteria in to ensure that the goals of the assignment are met.

6.2 Describe the typical basis for determining the final grade (e.g. weighting of various student projects):

15%	Weekly Labs
60%	Homework assignments
25%	Exams (midterm and final)

6.3 Grading type:

Points	Grade
97 – 100%	A+
93 – 96%	A
90 – 92%	A-
87 – 89%	B+
83 – 86%	B
80 – 82%	B-
77 – 79%	C+
73 – 76%	C
70 – 72%	C-
67 – 69%	D+
63 – 66%	D
60 – 62%	D-
0 – 59%	F

7.0 Resource Material Information

7.1 Textbooks and/or other required readings used in course:

Deitel, P. & Deitel, H. (2011). *Java How to Program, Ninth Edition.*. Boston, MA: Prentice Hall.

7.2 Other student suggested reading materials:

This will be left up to the individual instructors of the course.

7.3 Current bibliography and other resources:

Deitel, P. & Deitel, H. (2011). *Java How to Program, Ninth Edition.*. Boston, MA: Prentice Hall.

Niemeyer, P. & Knudsen, J. *Learning Java.* Sebastopol, CA: O'Reilly Media.

Darwin, I. *Java Cookbook, Second Edition.* Sebastopol, CA: O'Reilly Media.

Horstmann, C. & Cornell, G. *Core Java, Volume I.* Boston, MA: Prentice Hall.

Flanagan, D. *Java Examples in a Nutshell, 3rd Edition.* Sebastopol, CA: O'Reilly Media.

Oracle Corporation. "Java 2 Platform SE 5.0." Retrieved from http://download.oracle.com/javase/1.5.0/docs/api/

Oracle Corporation. "Lesson: Classes and Objects." Retrieved from http://download.oracle.com/javase/tutorial/java/javaOO/

Oracle Corporation. "Lesson: Basic I/O." Retrieved from http://download.oracle.com/javase/tutorial/essential/io/

Oracle Corporation. "Lesson: Generics." Retrieved from http://download.oracle.com/javase/tutorial/extra/generics/index.html

Thiebaut, D. "Sorting Algorithms." Retrieved from http://maven.smith.edu/~thiebaut/java/sort/demo.html

8.0 Other Information:

8.1 Accommodations statement:

8.2 Other:

8.3 Author(s):

Patrick Cavanaugh

9.0 Computer Science Accreditation Board (CSAB) Category Content (class time in hours):

CSAB Category	*Core*	*Advanced*
Data structures	15	0
Computer organization and architecture	0	0
Algorithms and software design	10	0
Concepts of programming languages	0	20

10.0 Oral and Written Communications:

Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) to typically 0 pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

11.0 Social and Ethical Issues:

The importance of testing algorithms is made clear to students. Examples of algorithm failure and the impact of these failures are presented.

12.0 Theoretical content:

This course introduces algorithmic problem-solving in the context of a modern programming language. Such topics as problem solving strategies, basic data structures, data and procedural abstraction, and algorithmic complexity are treated.

Contact Hours	
Data Representation	2.0
Classes and Objects	8.0
Recursion	2.0
Sorting Techniques	5.0
Exceptions	3.0
Array List	3.0
Generics	2.0
Stacks & Queues	2.0
Linked Lists	2.0
Trees	4.0
Files	4.0
GUI Applications	4.0

13.0 Problem analysis:

Basic problem analysis is a focal point of this course. While learning the techniques of abstraction and encapsulation for basic data structures, the programming exercises provide opportunities to use the analysis techniques discussed in class.

14.0 Solution design:

This course amplifies the object-oriented programming techniques introduced in CIST 1400. Students learn fundamental techniques for organizing data using a variety of classic data structures. Treatment of memory management and recursion complement the discussion of these data structures. The significance of choosing the appropriate data structure for a solution is illustrated throughout the design of solutions to the programming exercises

CHANGE HISTORY

Date	***Change***	***By whom***	***Comments***
02/20/1998	Original Version	Clark	
06/13/2003	Cleanup, ABET-specific material	Wileman	
12/2/2008	Cleanup, ABET-specific material	Clark	
02/15/2011	Coordinator, Prereq topics, Objectives, Content & Organization, Text, Theoretical Content	Cavanaugh	
04/26/2011	Revised for new format	Cavanaugh	
02/22/2014	Syllabus Reviewed, no changes	Cavanaugh	
1/17/2017	Updated to add required lab	Dorn	Initial output of CSEd Working group during Fall 2016
1/18/2017	Updated content, adjusted grade weights	Cavanaugh	Tweaking further from working group

S – Strong relationship
X – Contributing relationship

Course objective	**(a) knowledge of discipline**	**(b) analyze problem, define requirements**	**(c) design and implement solution**	**(d) function on a team**	**(e) ethical issues**	**(f) communicate effectively**	**(g) analyze impact of computing**	**(h) continued professioanl development**	**(i) Current techniques and tools**	**(j) apply foundations**	**(k) apply design and development principles**
Data Representation	**X**									**X**	
Classes and Objects	**X**	**X**	**X**							**X**	
Recursion			**X**								**X**
Sorting		**X**							**X**		**X**
Exceptions		**X**									**X**
Data Structures	**X**	**X**	**X**						**X**	**X**	**X**
Files			**X**							**X**	
GUI Applications	**X**		**X**						**X**		**X**