

COURSE DESCRIPTION

| | | | |
|----------------|------------------|--------------------|---|
| Dept., Number | CSCI 130L | Course Title | Introduction to Computer Science Lab |
| Semester Hours | 1 | Course Coordinator | Crystal Edge/ Jeannie French |
| | | URL (if any): | |

Current Catalog Description

(Computer Usage) (Coreq: Computer Science 130 and Mathematics 130) Laboratory demonstrates the topics and principles presented in the CSCI 130 lecture. F, S, Su.

Textbook

J. Glenn Brookshear, *Computer Science – An Overview, 9th Ed.*
ISBN 0-321-38701-5

References

Course Goals

The objective of this course, in conjunction with the lecture course, is to provide an introduction to computer science and programming for non-majors or beginning computer science majors. The course will cover basic computer operation and components, how to develop Web pages, and how to write programs to solve given problems using a high-level programming language. Programming concepts including variables, methods, objects, conditional statements, and loops will be introduced and practiced. Also included in the course are several hands-on Action Labs to introduce students to various exciting sub-disciplines of computer science, such as bioinformatics, hardware, graphics, security, and networking.

Learning Outcomes:

After successful completion of this course, the student will be able to

Define algorithm

Describe the main components of a computer

Describe how the Web servers and the Internet work

Create a web site using HTML and CSS

Discuss various security issues in computing

Identify several sub-fields of computer science

Define programming concepts including variables, methods, conditional statements, repetition, objects

Develop algorithms to solve given problems

Write Python programs to solve given problems

Prerequisites by Topic

None

Major Topics Covered in the Course

Computer components and operations
Algorithm design
Web page development
Introductory programming (variables, methods, objects, control structures)
Various sub-fields of computer science (bioinformatics, hardware, security, networking, graphics)

Laboratory projects (specify number of weeks on each)

Problem solving/Algorithm Design – 8 weeks (ongoing)
Introduction to specialty areas in computer science – 3 weeks total
Internet and Web architecture – 1 week
HTML and CSS/Web page development – 5 weeks
Introductory Programming
 Program structure and style – 1 week intro (ongoing)
 Data types – 1 weeks
 Relational, Logical, and Arithmetic operations – 1 week
 Control structures – 2 weeks
 Methods/subroutines – 1 week
 Input/Output – 1 week intro (ongoing)

Estimate Curriculum Category Content (Semester hours)

| Area | Core | Advanced | Area | Core | Advanced |
|-----------------|------|----------|-----------------|------|----------|
| Algorithms | .5 | | Data Structures | | |
| Software Design | | | Prog. Languages | .5 | |
| Comp. Arch. | | | | | |

Oral and Written Communications

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

Social and Ethical Issues

Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

Approximately 5% of class time is spent discussing the significance of a career in computer science and the ethical issues that face computer scientists, including software reliability and intellectual copyright. Students are not tested on these topics in the lab course.

Theoretical Content

Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

Problem Analysis

Please describe the analysis experiences common to all course sections.

For each lab, students are given descriptions of problems that can be solved with computer programs. They are required to analyze the problem descriptions to gather requirements, including input and output specifications and necessary formulas or equations.

Solution Design

Please describe the design experiences common to all course sections.

After analyzing problem descriptions and extracting requirements and specifications for each lab, students design a solution to the problem using a specified programming language. Students must determine what types of variables and data structures to use, design an algorithm to solve the given problem, and implement the algorithm. They also must design a Website using what they have learned of HTML and CSS.