

Computer Science 160 - Computer Science I

Instructor – Tom Stokke

Office – Upson II 366C

Email – thomas.stokke@und.edu

Office Hours – TR 1:30 – 2:30, or by appointment

Phone - 777-3337 or 777-4107 (CSci Office)

Required texts

zyBooks Programming with Python 3

To access the book:

- Sign in or create an account at learn.zybooks.com
- Enter zyBook code: UNDCSCI160StokkeFall2018
- Subscribe

Websites of interest

Python - <https://www.python.org> - make sure you download Python 3

Blackboard – <http://blackboard.und.edu>

Processing – <http://processing.org>

GTAs

Manohar Kumar Reddy Dasannagari

Smruti Nanda

Email – manoharkumarreddy.da@und.edu

Email – smruti.nanda@und.edu

Office – Starcher 238C

Office – Starcher 238C

Purpose

This course surveys computer science and includes topics in algorithms, machine architecture, data representation, operating systems, and software. Emphasis will be placed on algorithm development, structured programming, and object-oriented design using the Processing programming language. After completion students will be familiar with the core areas in computer science and will be able to design, write, test, and debug moderately complex programs.

Learning Outcomes:

At the end of this class, the students will be able to:

Implement psuedocode algorithms using a structured programming language.

Demonstrate problem solving strategies using creative and critical thinking.

Demonstrate the appropriate use of data types.

Use the basic commands for decisions making and iteration.

Learn debugging techniques for the detection and correction of runtime and logical errors.

Demonstrate the basic techniques for modularization of programs using top-down and/or bottom-up design.

Understand and demonstrate simple use of arrays.

Use documentation to enhance readability and understanding.

Course Requirements

Lecture:	Two semester tests – 100 points each	200 points
	Final	150 points
	In-class/homework assignments	50 points
Lab:	10 lab assignments – 10 points each	100 points
	10 in-lab assignments – 10 points each	100 points
	3 lab tests – 50 points each	150 points

There are 750 points available for the semester. Letter grades will be determined using 90, 80, 70, and 60 percent criteria. The instructor reserves the right to alter the percentage, but only to the advantage of the student.

Material covered

Chapter 1

Basic Processing graphic primitives

Chapter 2 – Variables and Expressions

Chapter 3 – Types

Chapter 4 – Branching

Chapter 5 – Loops

First Test – October 5 (tentative) – Lab Test 1 10/2 or 10/4

Chapter 6 – Functions

Chapter 7 - Strings

Chapter 7 – Lists

Second test – November 5 (tentative) – Lab Test 2 11/6 or 11/8

Chapter 7 – Dictionaries

Lab Test 3 12/4 or 12/6

Final – 9:00 section Monday, December 10, 8:00 AM

10:00 section Wednesday, December 12, 10:15 AM

General

- 1) Exams will only be given at the scheduled times unless prior arrangements have been made. Test dates are **TENTATIVE** and will be confirmed at least one week in advance in class.
- 2) Regular attendance is expected and students are responsible for any/all material covered in class.
- 3) All in-class assignments are due on the assigned dates (by close of business). No late assignments are accepted without prior approval.
- 4) If you have any special needs or concerns, which could include but are not limited to note taking or audio or visual difficulties, you must talk to your instructors. We will try to accommodate your needs, and if needed, will work with you and Disability Support Services to create a productive learning environment.

Essential Studies Contribution

This course has been approved for Essential Studies in the **Math, Science, and Technology** category. The course is specifically focused on helping students learn to use **Thinking and Reasoning** skills effectively with a particular focus on critical thinking. Students will work on developing their critical thinking capabilities when designing, implementing, and testing programs. When creating the programs, they will learn to analyze problems, break them down into different components, and then work out a course of action that will satisfy the problem. The desired solution will then be implemented using a programming language and, finally, tested for validity.

Computer science students learn logical reasoning, algorithmic thinking, and problem solving - concepts and skills that are valuable in more than just a computer science class. Computer science students learn to program, but only as a means to an end. Computer science teaches problem solving through the use of a programming language.

Plagiarism

Students are allowed and even encouraged to discuss ideas. Students can “help” other students with algorithms and small sections of code. Students are also expected to do their own work. Students handing in assignments or programs that are not their own work may be given the opportunity to “share” the points. For blatant first time cases or any second offense all students involved will receive an F for the course and may be reported for academic fraud.