CS161 – Introduction to Computer Science I
CRN 40069
Spring 2022

General Class Information

Instructor: Norah Wang

Email: wangn@linnbenton.edu (please allow 24 hours response time)

Class Dates	Time
Mon/Wed/Fri (lecture + demo)	12 PM to 1:50 PM

Class Zoom Link (opens in a new window)

- Alternatively, you can enter the Zoom ID:
- Passcode: CS161

Office Hours (please schedule an appointment by email):
- Tuesday   10 AM to 12 PM
- Thursday   1 PM to 2 PM

Office Hours Zoom Link (opens in a new window)

- Office Hours are a way for me to answer course questions and get to know you.
- Please **schedule an appointment with me beforehand by email** so that you do not have to wait when you come to office hours.
    o Friendly reminder - include the course name and number in the email subject line (e.g. CS161 Questions on Assignment 1).
- I'm here to help you learn, and I encourage you to ask questions early and often when the material or assignments are unclear.
- If you the above times do not work for you, feel free to email me and we can meet during other times.

Class Discord Server: (Professor Joseph Jess' server)

Zoom Room Conduct:

1) Participate from a quiet space where you can listen and speak.
2) It is highly recommended to turn the video on, but it is not mandatory.
3) If you know that you will be disturbed during the class time, make sure that you are muted, and your video is off while you are dealing with the issue.
4) Please respect the learning environment of others and keep distractions to a minimum.

Guidelines for Communication:

The best way to reach the instructor is by email. While they need not be strictly formal, your emails should be concise, list necessary details (course name/number), and written in a professional manner. Please allow 24 hours for response (although I usually reply promptly) and send me a friendly reminder if your questions are not addressed within this timeline. See Anatomy of an Email for reference.

Course description and pre-requisite:

Introduces the principles of computer programming using an object-oriented language. Includes topics related to problem-solving concepts, verification and validation, representation of data, sources of error, debugging techniques, conditional statements, loops, and arrays. Introduces both command line applications and those with graphical user interfaces.

As the name implies, CS 161 is the computer science department's introductory course. While it is the first course in the programming sequence for majors it is appropriate for non-majors and it does NOT assume that you have programming experience.
This course has two primary goals:

- First, to introduce the general field of computer science. We hope that you will leave this course with a sense of what computer science is and what computer scientists do.
- Second, to introduce the concept of programming. Programming is the way that computer scientists express their ideas and implement solutions to problems. Even if you never "program for a living", you will need to know how to program in order to appreciate the ideas you learn and to work in the industry.

Prerequisite:

- o **MTH 095** Intermediate Algebra or higher and **CS 160** Orientation to Computer Science, both with a grade of "C" or better.

Course Objectives:

While a major goal of this course is to provide a good start to the development of programming skills, the course is not solely about programming. Upon successful completion of the course students should have gained the following skills and proficiencies:

- Demonstrate an understanding of the difference between primitive data types and objects and their representation,
- Demonstrate the use of good program development, debugging techniques and documentation.
- Write object-oriented code that includes control statements, while loops, for loops, output to the screen and input from the keyboard and from a file.
- Write and run simple graphical user interface (GUI) applications.
- Write simple, user-designed classes that demonstrate an understanding of encapsulation.

Additionally, you should develop skills and understanding that will ultimately allow you to analyze complex problems and apply your knowledge and experience to developing good solutions to them. Programming is a creative process. However, to exercise that creativity, one must learn basic tools and principles. That is the purpose of this course.

Course Materials:

1) Recommended but not required: *The Practice of Computing Using Python, 2nd or 3rd edition*, by William Punch and Richard Enbody. ISBN-13: 978-0132805575.
2) Python Interpreter (version 3.6 or better; https://www.python.org/)
3) A text editor. Recommend Visual Studio Code.
4) Stable Internet connection.
5) A LBCC student Gmail account.

Other Learning Resources:

1) A Byte of Python (free and openly accessible)

Grading Table:

Assignments/Labs/Exams	Weight
Assignments – lowest score dropped.	30%
Labs	30%
Midterm	15%
Final Project (70%) + Final Exam (30%)	25%
➢ Late midterm/finals will **not** be accepted unless **permission has been obtained from the instructor in advance**. ➢ Assignments and labs will **not** be accepted **more than 3 days after the due date**. ➢ However, if something happens and you need some extra time for the assignments/labs, please **communicate with me beforehand** and we can create a doable plan for you to submit the work. ➢ Communication is the key!	
TOTAL	100%

Grades:

A: 90 – 100%
B: 80 – 89%
C: 70 – 79%
D: 60 -69%
F: < 60%

IMPORTANT: a grade of "C" (>= 70%) or higher is considered passing

Course Outline (tentative):

Week	Activities
1	**No class on Friday, April 1 (Campus closed)** o Welcome, introductions, syllabus, Moodle, scope & set clear expectations o Topics: What is CS?, Why Python?, Operations, functions (quick start) o Reading Assignment: 0.1-0.5, 1.1-1.5, 1.6-1.10
2	o Topics: Conditional statements, loops o Reading Assignment: 2.1,2.2.1-2.2.9
3	o Topics: more on functions, docstrings, strings o Reading Assignment: 2.2.10-2.3, 3.1-3.5
4	o Topics: strings: working with text, iterating over strings o Reading Assignment: 4.1-4.3, 4.4-4.8
5	o Topics: files and exceptions, functional abstraction o Reading Assignment: 5.1–5.7, 6.1–6.4
6	o Topics: lists and mutable data, tuples (and more lists) o Reading Assignment: 7.1-7.3, 7.4-7.6 o **Midterm**
7	o Topics: dictionaries, sets o Reading Assignment: 9.1-9.3,9.4-9.5
8	o Topics: recursion o Reading Assignment: 16.1-16.5
9	o Topics: intro to classes o Reading Assignment: 11.1-11.4, 12.1-12.6
10	**No class on Monday, May 30 – Memorial Day** o Work on final projects, review and reflection
11	o **Final Exam Due in Week 10** o **Final Project Due on Tuesday of the Finals Week.**

Important Dates – see [Academic Calendar](#) (opens in a new window).

Project/labs components are generally graded based on:

A. **Program Design (20%)**
Rating Criteria
20   Solution well thought out.
10   Solution partially planned out.
0-5   ad hoc solution; program was "designed at the keyboard" or no design submitted.

B. **Program Execution (20%)**
Rating Criteria
20   Program runs very well under a variety of conditions, as submitted.
10   Program runs much of the time, may be missing required files or instructions for libraries used.
0-5   Program runs very poorly, not at all, or requires several modifications or files before it runs.

C. **Specification Satisfaction (20%)**
Rating Criteria
20   Program satisfies specification completely and correctly.
10   Important parts of the specification not implemented.
0-5   Program poorly satisfies specification, or not at all.

D. **Coding Style (20%)**
Rating Criteria
20   Well-formatted, understandable code and appropriate use of language capabilities.
10   Code difficult to follow in one reading or poor use of language capabilities.
0-5   Incomprehensible code, poor use of language capabilities, or a need to scroll up and down repeatedly.

E. **Comments (20%)**
Rating Criteria
20   Concise, meaningful, and well-formatted comments and docstrings.
10   Partial, poorly written, or poorly formatted comments.
0-5   Wordy, unnecessary, incorrect, badly written or formatted, or none or nearly no comments.

Note: careful design, systematic testing, consistent style, and readability of code are important software quality factors (all of which are subject to interpretation but graded by the instructor based on the spirit and letter of the requirements, so be sure to explain your decisions).

Academic Honesty:

Academic integrity is the principle of engaging in scholarly activity with honesty and fairness and participating ethically in the pursuit of learning. Academic integrity is expected of all learners at LBCC. Behavior that violates academic integrity policies at LBCC includes cheating, plagiarism, unauthorized assistance or supporting others in engaging in academic dishonesty, knowingly furnishing false information, or changing or misusing college documents, among others. LBCC students are responsible for understanding and abiding by the College's academic integrity policy.

LBCC Center for Accessibility Resources:

LBCC is committed to inclusiveness and equal access to higher education. If you have approved accommodations through the [Center for Accessibility Resources (CFAR)](#) and would like to use your accommodations in this class, please contact your instructor as soon as possible to discuss your needs. If you think you may be eligible for accommodations but are not yet registered with CFAR, please visit the CFAR Website for steps on how to apply for services. Online course accommodations may be different than those for on-campus courses, so it is important that you contact CFAR as soon as possible.

LBCC Comprehensive Statement of Nondiscrimination:

LBCC prohibits unlawful discrimination based on race, color, religion, ethnicity, use of native language, national origin, sex, sexual orientation, marital status, disability, veteran status, age, or any other status protected under applicable federal, state, or local laws.

LBCC Statement of Inclusion:

The LBCC community is enriched by diversity. Each individual has worth and makes contributions to create that diversity at the college. Everyone has the right to think, learn, and work together in an environment of respect, tolerance, and goodwill (related to Board Policy #1015).