

# CS162 – Introduction to Computer Science II

## CRN 41720

## Spring 2022

### General Class Information

**Instructor:** Norah Wang

**Email:** [wangn@linnbenton.edu](mailto:wangn@linnbenton.edu) (please allow 24 hours response time)

Class Dates	Time
Mon/Wed/Fri (lecture + demo)	10 AM to 11:50 AM

[Class Zoom Link](#) (opens in a new window)

- Alternatively, you can enter the Zoom ID:
- Passcode: CS162

**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.
  - 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.

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

## 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:

Covers software engineering principles, basic data structures, and abstract data types including arrays, strings, lists, and others. Introduces the analysis of algorithms run-time complexity as well as algorithm testing and verification. Explores sorting and searching techniques. Expands the areas of graphical user interfaces event-driven programming introduced in CS 161. Covers polymorphism, inheritance, recursion, and exception handling.

### Prerequisite:

- **CS 161** Orientation to Computer Science I, with a grade of "C" or better.

## Course Objectives:

Upon successful completion of this course, students will be able to:

- Write and debug object-oriented code using class dependencies of aggregation and inheritance.
- Write object-oriented code that includes the use of two-dimensional collections of data and objects.
- Demonstrate the use of various layout managers and listeners in an event-driven programming paradigm.
- Demonstrate the use of polymorphism through inheritance and draw Unified Modeling Language (UML) diagrams of class hierarchies.
- Write code to sort and search one-dimensional collections.
- Write code that detects and handles exceptions.
- Demonstrate an understanding of recursion, contrast it to iteration, and select the best approach for a given problem.

## Course Materials:

- 1) Recommended but not required: *The Practice of Computing Using Python, 2<sup>nd</sup> or 3<sup>rd</sup> edition*, by William Punch and Richard Enbody. ISBN-13: 978-0132805575.
- 2) [Python Interpreter](https://www.python.org/) (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 (not required):**

- 1) [A Byte of Python](#) (free and openly accessible)
- 2) Python Programming: An Introduction to Computer Science ([Amazon link](#))  
 Author: John Zelle  
 ISBN: 978-1-59028-275-5
- 3) Practical Programming ([Amazon link](#))  
 Authors: Paul Gries, Jennifer Campbell, Jason Montojo  
 ISBN: 978-1-68050-268-8

**Grading Table:**

Assignments/Labs/Exams	Weight
<b>Weekly Assignments</b>	70%
<ul style="list-style-type: none"> <li>➤ There are a number of projects to be completed for this class, designed to challenge and solidify design, coding, and testing skills.</li> <li>➤ Project components are generally graded based on:               <ul style="list-style-type: none"> <li>✓ Completeness (does it work)</li> <li>✓ Correctness (does it meet the listed requirements)</li> <li>✓ Quality and explanation of design (features in advance, organized)</li> <li>✓ Quality and explanation of tests (test each feature for expected successes and failures)</li> <li>✓ Quality of implementation (consistent and readable style, runs well, is easy to learn and use)</li> </ul> </li> </ul>	
<b>Final Project</b>	30%
<ul style="list-style-type: none"> <li>➤ Late final project will <b>not</b> be accepted unless <b>permission has been obtained from the instructor in advance.</b></li> <li>➤ Weekly will <b>not</b> be accepted <b>more than 3 days after the due date.</b></li> <li>➤ However, if something happens and you need some extra time for the assignments/labs, please <b>communicate with me beforehand</b> and we can create a doable plan for you to submit the work.</li> <li>➤ Communication is the key!</li> </ul>	
<b>TOTAL</b>	100%

**Grades:**

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

**IMPORTANT:** a grade of “C” ( $\geq 70\%$ ) or higher is considered passing

**Course Outline (tentative):**

Week	Activities
1	<p><b>**No class on Friday, April 1 (Campus closed)**</b></p> <ul style="list-style-type: none"> <li>○ Session 1 (variables, conditionals, loops, functions, input, scope, comments, docstrings)</li> <li>○ A bit more time: functions, input, scope, comments, docstrings</li> </ul>
2	<ul style="list-style-type: none"> <li>○ Session 2 (Classes, methods, objects, file I/O, basic testing ideas, version control systems)</li> <li>○ Classes, methods, objects</li> <li>○ pydocstyle (PEP documentation style checker), pycodestyle (PEP code style checker)</li> <li>○ Version control systems: Git tutorial?</li> </ul>
3	<p><b>**No class on Friday, April 15 (instructor training)**</b></p> <ul style="list-style-type: none"> <li>○ Session 3 (recursion, variables + functions = classes, basic GUIs, PyTest, Type hint?)</li> </ul>
4	<ul style="list-style-type: none"> <li>○ Session 4 (interfaces, more GUIs, events, MVC, command line arguments)</li> </ul>
5	<ul style="list-style-type: none"> <li>○ Session 5 (yet more GUIs, wrappers, event listeners, search algorithms, more recursion)</li> </ul>
6	<ul style="list-style-type: none"> <li>○ Session 6 (sorting algorithms, more on testing, yet more recursion)</li> </ul>
7	<ul style="list-style-type: none"> <li>○ Session 7 (inheritance, UML diagrams, exceptions, polymorphism, custom GUI widgets)</li> <li>○</li> </ul>
8	<p><b>**No class on Friday, May 20 (instructor training)**</b></p> <ul style="list-style-type: none"> <li>○ Session 8 (discuss final projects, more work on inheritance and exceptions, start an in-class project)</li> <li>○ More examples of inheritance and some more on exception classes.</li> <li>○ Exceptions (exception, exception class, exception inheritance, exception hierarchy, precedence of exceptions)</li> </ul>
9	<ul style="list-style-type: none"> <li>○ Session 9 (continue in-class project, and work on final projects)</li> </ul>

10	<p><b>**No class on Monday, May 30 - Memorial Day**</b></p> <ul style="list-style-type: none"> <li>○ Work on final projects, review and reflection</li> </ul>
11	<ul style="list-style-type: none"> <li>○ <b>Final Project Due on Tuesday of the Finals Week.</b></li> </ul>

## Projects 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).