Spring 2024
CSCI-41
Programming Concepts and Methodology II
Syllabus

Instructor: Simon Sultana, Ph.D.
Department: Computer Science
E-mail: simon.sultana@reedleycollege.edu
Phone: (559) 638-0300 x3192
Response time: < 24 hours
Office Hours: MW 10am-12pm, TuTh 2-3pm (PHY-81), or by appt
Prerequisites: CSCI-40 or ENGR-40.

Section Number: 52005
Unit(s): 4
Dates: 1/8/2024 – 5/16/2024
Class Meeting: TTh 12-1:50pm
Location: PHY-82

Advisory: ENGL-125 – Writing Skills for College and ENGL-126 Reading Skills for College

Course Description: This course introduces application of software engineering techniques to the design and development of large programs, data abstraction and structures, and associated algorithms. Topics include linear and non-linear data structures such as lists, stacks, queues, trees, and graphs, algorithms for recursion, searching, sorting, and traversal.

Note: this course is offered in a hybrid format. There will be a recorded lecture available at the beginning of the week. Students are expected to work on zyBooks activities before our second weekly class meeting. The meeting will be used to work on class activities in small teams. The meetings are offered in two-way interactive format, meaning that students have the choice of joining on campus or via Zoom. Please note that live attendance is required each week (in person or online). It is up to each student how they would like to attend each week. Note we will be working on labs in small groups and we will combine students who are in person and online. Note also that students are required to attend the tests (Thursday, February 1; Tuesday, February 27; and Thursday, April 4) and the final exam (Thursday, May 16, 2024) on campus. Please let me know if you have any questions.

Course Goals and Student Learning Outcomes:
Upon completion of this course, students will be able to:
- Develop computer programs to meet given requirements.
- Utilize classic data structures; including linked lists, stacks, queues, trees, and graphs in computer programs.
- Analyze the efficiency of algorithms.
- Write programs using advanced programming concepts; including constructor and operator overloading, inheritance, abstract classes, and templates.

Objectives:
In the process of completing this course, students will:
- Utilize binary search tree data structure.
- Use string, stack and queue classes defined in C++ Standard Libraries.
- Identify and correct syntax and logical errors in computer programs.
- Utilize binary search tree data structure.

- Create proper test cases to test computer programs.
- Define and use dynamic arrays, linked list, stacks and queue data structures.
- Design, implement, and test the implementation of "is-a" relationships among objects using a class hierarchy and inheritance.
- Use string, stack and queue classes defined in C++ Standard Libraries.
- Identify and correct syntax and logical errors in computer programs.
- Write a total of 1000 to 2000 lines of program code.
- Write programs using object-oriented programming and the C++ language.
- Write programs using pointers, recursion and file manipulation techniques.
- Examine ethical issues in software engineering

Student Learning Outcomes are statements about what the discipline faculty hope you will be able to do at the end of the course. This is NOT a guarantee: the ultimate responsibility for whether you will be able to do these things lies with you, the student. In addition, the assessment of Student Learning Outcomes is done by the department in order to evaluate the program as a whole, and not to evaluate individual faculty performance.

Required or Recommended Textbooks and Materials:

Textbooks:

1. **Required**: zyBooks, register and connect via Canvas (cost: $58).
2. **Optional:** *Starting out with C++ From Control Structures through Objects*, 9th Ed, By Tony Gaddis, Pearson

Additional Resources:

Towson University Cybersecurity Modules: https://cis1.towson.edu/~cssecinj/

Learning Management System: CANVAS:

Canvas (https://scccd.instructure.com/) is used to post announcements, course information, programming assignments, and grade. You will submit your programming assignments on Canvas.
To log-in Reedley College CANVAS:

Username: Your 7-digit student ID number.

Password: If you have not previously changed your password, it is:

First name initial (upper case) + *last name initial* (lowercase) + *date of birth* (mmddyy)
Example: John Smith born on July 9th of 1988 Password =Js070988

Computer Lab:

Onsite face-to-face meetings will take place in FEM 3 on campus. Microsoft Visual Studio C++ compiler is used for this class.

Topics:

A. Classes

1. Classes and Members; Friend Functions
2. Constructors and Destructors
3. Using a Namespace, Header file, and Implementation file
4. Operator Overloading

B. Inheritance, Polymorphism, and Virtual Functions

1. Introduction to Inheritance
2. Protected members and Class Access
3. Constructors and Destructors in Base and Derived Classes

4. Redefining Base Class Functions
5. Class Hierarchies
6. Polymorphism and Virtual Member Functions
7. Abstract Base Classes and Pure Virtual Functions

C. Pointers and Dynamic Memory
1. Pointers and arrays as parameters
2. A container class with a dynamic array

D. Linked Lists
1. A fundamental node class for linked lists
2. A linked-list toolkit
3. A container class with a linked list

E. Software Development with Templates, Iterators, and the STL
1. Template functions
2. Template classes
3. Standard template classes and their iterators

F. Stacks
1. Stacks and the STL stack
2. Stack applications

G. Queues
1. Queues and the STL queue
2. Queue applications

H. Recursion
1. Recursive functions
2. Recursive applications

I. Binary Search Trees
1. Binary trees
2. Binary tree representations
3. Binary tree nodes
4. Binary tree traversals
5. Binary search trees

J. Hash Tables
1. Chaining
2. Linear & quadratic programming
3. Hashing

J. Graphs
1. Graph definition
2. Graph implementation
3. Graph traversal
4. Path algorithm

K. Further topics
1. Repositories

2. Programming style
3. Secure programming
4. Engineering ethics
5. Heaps

Tentative Schedule:

	Assignment Points	Date
Week 0: CSCI-40 Topics Review	**25**	**1/8/2024**
Class 12pm - Intro to Class		1/9/2024
STEM Connect/Class 12pm		1/11/2024
In-Class Activity	10	1/15/2024
zyBooks Participation Activities (optional for review)	-	1/14/2024
Project #0 Proposal	10	1/15/2024
Schedule One-on-One	5	1/14/2024
Week 1: Inheritance & Abstract Classes	**52**	1/15/2024
Class 12pm		1/16/2024
Mini Quiz	5	1/16/2024
zyBooks Participation Activities	7	1/18/2024
Class 12pm		1/18/2024
In-Class Activity	10	1/21/2024
zyBooks Week 1-2 Challenge Activities	-	1/21/2024
Project #1 - User Interface & Dynamic Array	30	1/29/2024
Week 2: Templates, Style Guidelines, & Intro to GitHub	**19**	1/22/2024
Class 12pm		1/23/2024
zyBooks Participation Activities	1	1/25/2024
Mini Quiz	4	1/25/2024
Class 12pm		1/25/2024
In-Class Activity	10	1/28/2024
zyBooks Week 1-2 Challenge Activities	4	1/21/2024
Week 3: Test #1 / Stand-Up Meeting	**45**	1/29/2024
Class 12pm - Stand Up Meeting	10	1/30/2024
Class 12pm - Test #1 (covers weeks 0 - 3) ON CAMPUS ONLY – PHY 82	35	2/1/2024
Week 4: Intro to Data Structures & Algorithms	**74**	2/5/2024
Class 12pm		2/6/2024
zyBooks Participation Activities	6	2/8/2024
Mini Quiz	4	2/8/2024
Class 12pm		2/8/2024
In-Class Activity	10	2/11/2024
zyBooks Weeks 4-6 Challenge Activities	14	2/25/2024
Project #2 - Array of Structures	40	2/19/2024
Week 5: Searching & Algorithm Analysis	**23**	2/12/2024
Class 12pm		2/13/2024
zyBooks Participation Activities	8	2/15/2024
Mini Quiz	5	2/15/2024
Class 12pm		2/15/2024
In-Class Activity	10	2/18/2024

zyBooks Weeks 4-6 Challenge Activities	-	2/25/2024
Week 6: Sorting Algorithms, Secure Programming #1	**76**	2/19/2024
Class 12pm		2/20/2024
zyBooks Participation Activities	12	2/22/2024
Mini Quiz	4	2/22/2024
Class 12pm		2/22/2024
In-Class Activity	10	2/25/2024
zyBooks Weeks 4-6 Challenge Activities	-	2/25/2024
Project #3 - Searching & Sorting	50	3/4/2024
Week 7: Test #2 / Linked Lists 1	**60**	2/26/2024
Class 12pm Test #2 (on weeks 4 - 6) ON CAMPUS ONLY – PHY 82	34	2/27/2024
zyBooks Participation Activities	6	2/29/2024
Class 12pm		2/29/2024
zyBooks weeks 7-10 Challenge Activities	20	4/1/2024
Week 8: Linked Lists 2	**74**	3/4/2024
Class 12pm		3/5/2024
zyBooks Participation Activities	9	3/7/2024
Mini Quiz	5	3/7/2024
Class 12pm		3/7/2024
In-Class Activity	10	3/10/2024
zyBooks weeks 7-10 Challenge Activities	-	4/1/2024
Project #4 - Linked List	50	3/25/2024
Week 9: Stacks & Queues	**25**	3/11/2024
Class 12pm		3/12/2024
zyBooks Participation Activities	10	3/14/2024
Mini Quiz	5	3/14/2024
Class 12pm		3/14/2024
In-Class Activity	10	3/17/2024
zyBooks weeks 7-10 Challenge Activities	-	4/1/2024
Week 10: Hash Tables	**26**	3/18/2024
Class 12pm		3/19/2024
zyBooks Participation Activities	11	3/21/2024
Mini Quiz	5	3/21/2024
Class 12pm		3/21/2024
In-Class Activity	10	3/24/2024
zyBooks Challenge Activities	-	4/1/2024
Week 11: Test #3 / Software Engineering Ethics	**121**	4/1/2024
Class 12pm - Stand Up Meeting	10	4/2/2024
Discussion Forum - Software Engineering Ethics	15	4/5/2024
Class 12pm Test #3 (on weeks 7-10) ON CAMPUS ONLY – PHY 82	36	4/4/2024
In-Class Activity	10	4/7/2024
Project #5 - Stack/Queue	50	4/15/2024
Week 12: Trees	**41**	4/8/2024
Class 12pm		4/9/2024
Mini Quiz	4	4/11/2024
Class 12pm		4/11/2024

zyBooks Participation Activities	12	4/11/2024
In-Class Activity	10	4/14/2024
zyBooks weeks 12-16 Challenge Activities	15	5/10/2024
Week 13: Balanced Trees & Heaps, Secure Programming #2	**76**	4/15/2024
Class 12pm		4/16/2024
Mini Quiz	4	4/18/2024
Class 12pm		4/18/2024
zyBooks Participation Activities	12	4/18/2024
In-Class Activity - Secure Programming	10	4/21/2024
Project #6 - Hash Table	50	4/29/2024
Week 14: Code Review	**130**	4/22/2024
Class 12pm		4/23/2024
Class 12pm		4/25/2024
Code Review	25	4/25/2024
Project #7 - Final	100	5/9/2024
Project Peer Evaluation	5	5/9/2024
Week 15: Graphs	**25**	4/29/2024
Class 12pm		4/30/2024
zyBooks Participation Activities	11	5/2/2024
Mini Quiz	4	5/2/2024
Class 12pm		5/2/2024
In-Class Activity	10	5/5/2024
Week 16: More Algorithms with Graphs	**8**	5/6/2024
Class 12pm		5/7/2024
zyBooks Participation Activities	8	5/10/2024
Class 12pm		5/9/2024
In-Class Activity - Spanning Tree, Dijkstra's Algorithm, & More Security	-	5/10/2024
Week 17: Finals Week	**100**	5/13/2024
Final Exam **12pm ON CAMPUS ONLY - PHY 82**	100	5/16/2024
TOTAL	1000	

Subject to Change:
This syllabus and schedule are subject to change. If you are absent from class meeting, it is your responsibility to check on any changes made while you were absent.

Evaluation:
Students will be evaluated on the basis of their performance on various assignments according to the following scale. The instructor reserves the right to adjust scores as it may be required throughout the semester.

Points in the course total 1000 and are distributed as follows:

zyBooks Participation Activities (13)	113
zyBooks Challenge Activities (3)	53
In-Class Activities (13 of 14)	130
Programming Projects (8)	385
Mini-Quizzes (11), Tests (3), & Exam (1)	254
Software Engineering Activities/Forums/Code Review	60

Final grade is assigned using following scale:

895+ points	A
795 – 894 points	B
695 – 794 points	C
595 – 694 points	D
< 595 points	F

Attendance

Attendance for onsite and synchronous online meetings will be taken at beginning of each class. Students who leave before the end of class will be marked tardy. For synchronous online meetings you are expected to have your camera turned on and will have the best experience if you use earbuds/headphones. Please make sure to stay muted until you have a question or something to add so as to cut down on background noise.

Students will be dropped from the class if they fail to attend the first class session of the semester. During the semester up to final drop date, any student who missed two weeks of class meetings (cumulative) will be dropped from this class (i.e. 4 classes).

Make-up tests are limited to students who have made arrangements with the instructor **prior** to the announced testing date or those students who have been excused by High School Attendance Office. Exam material is constructed from class discussions, assigned readings, guest lectures, video presentations, and special assignments. **Unless the student receives prior approval from the instructor, no make-up tests will be allowed.**

Grading Policy

zyBooks Participation Activities:

ZyBooks participation activities are to be completed before announced due date (typically at start of the second weekly class meeting for full credit for augmented credit as shown below, exception in the first two weeks). These activities allow you to actively engage each week's content. Points will be attributed according to the number of responses required and the percentage of the activities a student completes before class.

Percentage Completed	Before Due Date Points Awarded	By End of Week Points Awarded
90%+	100% of points	90% of points
80%+	90% of points	80% of points
70%+	80% of points	70% of points
60%+	70% of points	60% of points
50%+	60% of points	50% of points
25%+	40% of points	30% of points
13%+	20% of points	10% of points

In-Class Activities:

In-class activities will be completed in groups of 2-3 during class meeting time with some outside of class time, as necessary. Only one assignment needs to be submitted per group. Be sure to include names of all students in the group or credit will not be given. The lowest score on the activities of the following weeks will be dropped: 0, 1, 3, 15, 16.

Programming Assignments:

Seven programming mini-projects are assigned in this course. Students will work individually (Project 0 pre-work, Project 2, Project 5, and Project 6) and in pairs if desired. Students should NOT share code with other individuals or groups unless working in a pair on a group project. Projects are suggested to be turned in for grading by the due date indicated on the course calendar. These are to be done However, these can be turned in up to two weeks later at no penalty (with the exception of Project #7 as all assignments are due before the week before the final exam) **as long as the student submits a stand-up video on the original due date (see late policy) and the following week, if needing a second extra week.** Stand-ups are required from each student, even in group projects if not meeting the original due date. Please be careful as it is not a good idea to get too far behind and the work will build up and you will likely have a difficult time succeeding in the class. Consideration is given to the following criteria: functionality, error-proofing/exception handling, efficient use of course concepts, documentation, and readability in addition to test case results. All programming exercises must include your name at the top of all programs submitted, or you will not receive credit.

Stand-Up Meetings and Code Review:
A stand-up meeting will be held during class in the week of Test #1. Students are to present 2-3 minutes (either in person or online with camera on) on their progress on the current project, what is planned to be accomplished in the next week, and most importantly, their biggest hurdle. The goal is for students to share best practices and get assistance as needed. For the Code Review, students are to present a piece of code they have written and discuss their design and coding choices. These must be done in person or online with a camera on. Students should share their code as they discuss it. Students are also responsible for entertaining questions at the end of their presentation and are responsible for asking questions or making meaningful suggestions to at least two other students. See Canvas for details.

Mini-Quizzes, Tests, and Exam:
There are (mostly) weekly mini-quizzes, three tests, and one final exam. All are timed. See course calendar for timing, content covered, and points allotted. Mini-quizzes review content from lecture videos. Tests will consist of multiple-choice questions on syntax, program flow, etc. There will also be small programming problems. A more difficult late test can only be arranged if you have an excuse verified by an impartial party (i.e., a doctor or a court note). Final exam is cumulative.

Late Policy
Points will be minimally deducted for late homework problem sets/zyBooks challenge activities at 0.5 – 1 point for assignments late. Problem sets are to be submitted electronically in Canvas and all work must be shown for partial credit. Programming assignments can be turned in up to **TWO WEEKS** late after the original announced due date, and never later than the announced deadline of project #7. **If you will turn in a project in late, you must submit a stand-up video by the original due date in each week it is late (for each week late)** and you must address the following questions in the video:

a) What you have accomplished so far (show/share code) focused on the project theme
b) What you plan to accomplish in the next week
c) Your biggest hurdle

The video should be 2-4 minutes and can be linked in the comments section of the assignment or shared in the submission area. The focus of the video should be on the specific topic of the project (e.g. linked list, stack or queue, etc.). Failure to meet this deadline will result in a zero on the project.

POGIL exercises will not be accepted late as these are group endeavors.

College Policies:
Cheating & Plagiarism, see Cheating and Plagiarism under Campus Policies:

Cheating and plagiarism is prohibited in the class. Incidents of cheating and plagiarism will result a failing grade on the particular assignment in question. Please see Student Conduct Standards at https://www.reedleycollege.edu/about/about-us/policies-and-procedures/student%20conduct%20standards.html for more information about academic integrity.

Each student is expected to assist in the overall environment of the classroom making it conducive to learning.

Code created by external sources can be utilized with permission from the instructor and 1) must be properly documented so as not to plagiarize the work of others and 2) must not make up more than 10% of the actual code (exclusive of documentation) of the project.

It is expected each student will do his/her own work unless otherwise instructed. This course involves both individual work and collaborative work. It is your responsibility to understand the guidelines that apply to each kind of work, and to be clear about which assignments are individual assignments and which are collaborative.

Activities not designated as cooperative assignments in this course are to be done individually. Individual work is to be carried out entirely and solely by an individual. Programs will be checked for plagiarism using both computer and human similarity checkers. Take extreme precautions that your individual work is not viewed by other students. This includes deleting all your computer files from public workstations when you are finished, retaining private permissions on your files, destroying printouts of source code, and not letting other students use your personal computer where you store your coursework.

In addition, the work you submit must be entirely your original creation. Using solutions from any other source is forbidden; in particular, using solutions (either instructors' or other students') from previous offerings of this or other courses is not allowed. Using solutions found on the Internet or getting help from online forums is not allowed.

Assignments which appear to be the result of a "group effort", or appear to have been copied from another student, will be considered plagiarized. Violations of this policy will be reported as violation of academic integrity per college policy.

In-class activities are designated as pair or team activities. Students will be assigned a partner or will self-assign to complete these activities. Be sure to use these collaborative activities as a chance to master the skills, as there will be tests in which each person must perform the skills individually.

A note on the use of AI

It is crucial to address the potential perils associated with the misuse of Artificial Intelligence (AI) in academic settings. Using AI to cheat not only undermines the learning experience but also violates the principles of academic integrity.

Perils of AI Cheating:

- Ethical Breach: Utilizing AI to cheat goes against the core values of honesty, integrity, and personal responsibility in academic pursuits.
- Short-Term Gains, Long-Term Consequences: While AI might provide a temporary advantage, the long-term consequences include a lack of genuine understanding, hindering your overall academic and professional development.

- Violation of Academic Policies: Most educational institutions have strict policies against cheating. Using AI to gain an unfair advantage can lead to severe academic penalties and damage your academic record.
- Missed Learning Opportunities: The primary purpose of education is to foster genuine understanding and skill development. Cheating with AI denies you the opportunity to acquire essential knowledge and skills.

My Expectations:

I, as your instructor, emphasize the importance of upholding academic integrity. I trust you will approach your studies with honesty, diligence, and a commitment to genuine learning.

Remember the true value of your education lies not just in grades but in the skills, knowledge, and ethical foundation you acquire. Resist the temptation to misuse AI for academic dishonesty, and instead, channel your efforts towards becoming proficient, ethical computer scientists.

Let's create an environment where academic excellence is synonymous with integrity.

Accommodations for Students with Disabilities
If you have a verified need for an academic accommodation or materials in alternate media (i.e., Braille, large print, electronic text, etc.) per the Americans with Disabilities Act (ADA) or Section 504 of the Rehabilitation Act, please contact the Reedley College Disabled Students Programs & Services (DSP&S) Department at (559) 638-0332. You can find more information at https://www.reedleycollege.edu/student-services/disabled-student-programs-and-services/index.html.

Reedley College is committed to creating accessible learning environments consistent with federal and state law. To obtain academic adjustments or auxiliary aids, students must be registered with the DSP&S office on campus. DSP&S can be reached at (559) 638-3332. If you are already registered with the DSP&S office, please provide your Notice of Accommodation form as soon as possible.

Important College Dates Spring 2024

Class begins	Monday	01/08/2024
Last day to drop a full-term class for a full refund	Friday	01/19/2024
Last day to register	Friday	01/26/2024
Last day to drop this class to avoid a “W” in person	Friday	01/26/2024
Last day to drop this class to avoid a “W” online	Sunday	01/28/2024
Last date to drop this class	Friday	03/08/2024
Spring recess (no classes held)	Mon-Fri	03/25-29/2024
No classes, campus is closed		
Martin Luther King, Jr. Day observed	Monday	01/15/2024
Lincoln Day observance	Friday	02/16/2024
Washington Day observance	Monday	02/19/2024
Good Friday	Friday	03/29/2024
Final Exam	Thursday	05/14/2024