CS1410 Course Syllabus

For additional course information, including prerequisites, corequisites, and course fees, please refer to the Catalog: https://catalog.uvu.edu/

Semester: Spring
Year: 2025
Course Prefix: CS
Course and Section #: 1410-002
Course Title: Object-Oriented Programming
Credits: 3

Course Description

Teaches proper program structure using the core concepts of object-oriented programming: **classes, objects, encapsulation, inheritance**, and **polymorphism.** Presents problems of **increasing size and complexity** requiring OOP techniques, standard libraries, and other appropriate language constructs. Presents methods to identify, define, and implement solutions to naturally recursive problems. May be delivered online.

This is a **required course** for CS majors.

You must earn a **C+** or higher in order to take the next course, **CS 2420** (Data Structures and Algorithms).

Course Attributes

This course has the following attributes:
☐ General Education Requirements
☐ Global/Intercultural Graduation Requirements
☐ Writing Enriched Graduation Requirements
☒ Discipline Core Requirements in Program
☐ Elective Core Requirements in Program
☐ Open Elective
Other: *Click here to enter text.*

Instructor Information

Instructor Name: Imtiaz Parvez

Student Learning Outcomes

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

1. Implement object-oriented solutions to problems of increasing size and complexity, which exhibit use of appropriate classes, objects, encapsulation, inheritance, and polymorphism
2. Test programs to assure that solutions are correct and complete
3. Design readable, maintainable code, using a good, consistent programming style
4. Use algorithms and data structures from standard libraries to solve problems

Mastery Rubric

This mastery rubric should help you gauge where you are and how you're doing in mastering the skills this course is intended to teach you mapped to a grade range. For example "C" represents "C+", "C", and "C-".

Course Outcomes	A (Mastered)	B (Near-Mastery)	C (Approaching Mastery)	D (Partially Mastered)	E (Not Yet Mastered)
Implement object-oriented solutions to problems of increasing size and complexity, which exhibit use of appropriate classes, objects, encapsulation, inheritance, and polymorphism	Consistently implements robust object-oriented solutions. Utilizes appropriate classes, objects, encapsulation, inheritance, and polymorphism. Handles complexity exceptionally well.	Implementations generally show understanding of object-oriented principles, but there may be occasional inconsistencies or errors. Handles complexity well, but struggles with some intricate problems.	Occasionally uses object-oriented principles correctly, but there are frequent inconsistencies or errors. Struggles with problems of moderate complexity.	Rarely uses object-oriented principles correctly. Demonstrates poor or inconsistent use of classes, objects, encapsulation, inheritance, and polymorphism. Difficulty handling simple problems.	Fails to correctly use object-oriented principles. Cannot handle even simple problems effectively.
Test programs to assure that solutions are correct and complete	Develops comprehensive test cases that ensure program correctness and completeness. Regularly employs testing frameworks and methodologies.	Develops good test cases, but they may not be entirely comprehensive. Uses testing frameworks and methodologies inconsistently.	Develops some useful test cases, but they may be limited in scope. Rarely uses testing frameworks or methodologies.	Struggles with developing adequate test cases. Testing may be sporadic or ineffective, and fails to ensure program correctness and completeness.	Fails to develop meaningful test cases. Does not employ testing frameworks or methodologies.
Design readable, maintainable code, using a good, consistent programming style	Code is consistently well-organized, readable, and maintainable. Adheres to best practices for programming style, including appropriate	Code is generally well-organized and readable, but may have occasional lapses in maintainability or style. May occasionally	Code is sometimes readable and maintainable, but often deviates from best practices for	Code is occasionally readable and maintainable, but generally lacks organization and adherence	Code is rarely readable or maintainable. Programming style is inconsistent, lacks appropriate documentation, or fails to

	naming conventions, documentation, and code structure.	deviate from best practices.	programming style.	to best practices.	adhere to best practices.
Use algorithms and data structures from standard libraries to solve problems	Consistently uses appropriate and efficient algorithms and data structures from standard libraries to solve problems. Demonstrates a deep understanding of algorithm and data structure principles.	Generally uses appropriate algorithms and data structures, but may occasionally choose inefficient or unnecessary options. Demonstrates good understanding of algorithm and data structure principles, but may struggle with more complex problems.	Occasionally uses appropriate algorithms and data structures, but often chooses inefficient or unnecessary options. Struggles with problems of moderate complexity.	Rarely uses appropriate or efficient algorithms and data structures. Demonstrates poor understanding of algorithm and data structure principles.	Fails to use appropriate or efficient algorithms and data structures. Does not understand algorithm and data structure principles.

Course Materials and Texts

Required materials, fees and technology:

- *CS: Objects in Python Links to an external site.*- Introduction to Objects in Python with native Codio content promoting engagement and active learning with fully auto-graded assessments and minimal text.

- Access to Codio for textbook, assignments and exams.
    - paid software (see steps below)
    - Codio Statement on Privacy Compliance and other Legal StuffLinks to an external site.
- Links to an external site.

  To access Codio:

  1. Click on **Modules** in Navigation at the left.
  2. Scroll down to Module 1 or any linked assignment or reading, and click on it. This will take you to a screen to access Codio.

3. Students who took CS 1400 at UVU within a year before CS 1410 will have previously used Codio and will have purchased a license that is good for one year. The same account should be active for CS 1410.
4. If #3 does not apply to you, you will be asked to purchase access and set up an account. Use a credit card to purchase access. Use the same email that you use for UVU login credentials. Make sure that you have **popups** enabled in your browser.

Optional Materials:

- Access to **Python** 3.10 or later on your own local machine.
  - free software
  - Python Software Foundation Privacy StatementLinks to an external site.
- Discord OnStudy server for CS 1410 discussion groups across all sections
  - free software
  - join link is bit.ly/onstudyLinks to an external site..
  - Privacy PolicyLinks to an external site.

Course Requirements

Course Assignments, Assessments, and Grading Policy

Grading

You earn a grade for completing assignments, you don't lose grades for not completing them. You start with and E and work your way up to a grade you want---you don't start with an A and lose it. *It is your job to demonstrate that you have mastered the material. It is not your instructor's or your IA's job to demonstrate that you have not.*

50% is the minimum score a student can receive on any assignment or project, keeping in mind that 50% is an E.

Grading Scale:

The following grading standards will be used in this class, correlate roughly to the mastery rubric above using the latter grades:

Percentages for Letter Grade

Grade	Minimum Percent
A	93

A-	90
B+	87
B	83
B-	80
C+	77
C	73
C-	70
D+	67
D	63
D-	60
E	0

Assignment Categories

Activity	Weight
Codio Readings	not graded
Watching Videos and other Supplements	not graded

Assignment Categories

Activity	Weight
Codio Practice Labs	not graded
Codio Weekly Exercises	17%
Codio Programming Projects	50%
Codio Midterm Exam	12%
Codio Final Exam	19%
Accountability	2%
Extra Credit	2%

Academic Dishonesty: Copying someone else's code and submitting it as your own is asking for a grade you did not earn and claiming mastery of skills that you have not demonstrated. We will use a plagiarism tool on your code in this class. This includes studying in groups and writing code together--two students MAY NOT submit the same code that they wrote together, nor code "given to you" by an IA in the lab.

You are not alone, though we do expect you to learn to solve problems. If you find yourself stuck on something, contact the instructor or a classmate, your course I.A. or the CS Tutor Lab for help. Class time, Office Hours, Discord and Teams are there for your support; please use them. If you need to, set up an appointment for help. These projects should be rewarding and instructional, not frustrating and demoralizing—but I don't know when or how to help unless you ask.

Late Work Statement: All work is due on the date assigned before the stoke of midnight: If an assignment is due on Sep 8, it is late at 12:00am on Sep 9. There is no grade penalty for late work, and assignments may be resubmitted with corrections for regrading. If you find yourself routinely submitting work 3-5 business days late, this will likely negatively affect your ability to learn later material and you may struggle to do well in the course.

Assignment and Assessment Descriptions

Assignments

There are two types of assignments for this course: **Reflective and Evaluative.**

Reflective Assignments

Reflective assignments give you an opportunity to reflect on your own learning--in this case the most important questions revolve around whether your program design and problem-solving skills are improving over time.

- **Pretest:** In week 1, you have Image Processing exercise as an assessment of general readiness for CS 1410. If you can complete it comfortably on your own within 1 hour, then you are likely OK. You don't have to pass this exercise to stay in the class, but if you cannot do it, you should have a serious talk with your instructor.
- **Self-test:** The level to which you feel comfortable experimenting with non-graded snippets given in reading material in Codio and how easy it is to complete non-graded Practice Labs gives you a low-stakes way of testing yourself and possibly stretching yourself in the class.
- **Practice Labs:** low stakes or additional coding practice within Codio. These labs are scored by Codio, but not included in your final grade.
- **Spiral Learning with Projects:** As in CS 1400, the curriculum in CS 1410 is not linear, but instead is a spiral. We introduce new ideas and revisit them over and over during the semester. The projects intentionally require you remember and use concepts covered earlier in the course. Most students realize that things which were difficult in week 3 are now easy in week 12 because that which they persist in doing often becomes easier.

Evaluative Assignments: Exercises and Projects

Coding Exercises (Module 1 - Module 6): Practice coding skills in Codio. These assignments are scored in Codio and count toward your final grade.

Projects (Module 7 - end): Projects include both single-session tasks and multi-week assignments where students revisit and enhance the same code using new skills. This spiral learning approach reinforces and builds upon prior coding knowledge.

Key projects include:

- **Dessert Shop Project:** Comprising 12 small parts, each designed to be manageable and progressively build coding skills.
- **GUI Project using Kivy**

Since each part of the Dessert Shop Project is small, it is manageable to complete two parts in a week.

*Evaluation***:** Project code will be manually graded. Please *do not* change the names of any provided files, functions or classes within the code--doing so makes submissions harder to grade.

All other exercises are autograded by Codio.

Discussions

There is a Discord link for CS 1410 to join via bit.ly/onstudyLinks to an external site.. Once you're in, go to the #course-list channel and click the box next to cs1410. However, keep in mind I am not using Discord for comminution. If you want to ask for help from me, please **email** me or message me via **Teams.**

Discord and Teams: Do not post spoilers nor executable code.

Communication within the Course

Mechanisms for getting help, communicating with others:

- Message the instructor. Make sure you know what the preferred contact method is.
- Talk to the Instructional Assistant assigned to your class section. Make sure you know who that is.
- Go to the CS Tutoring Lab in CS 726
- Initiate discussions in MS Teams for help or topics outside of class time. This is for your convenience and enrichment but will not be graded.

Assessments

- A midterm exam
- A final project exam

All exams are given in Codio but linked in Canvas in the Exams module on the Module Page**.** Each exam has instructions for taking it.

Required or Recommended Reading Assignments
Complete the required readings under the **"View and Watch"** pages for each module.

General Description of the Subject Matter of Each Lecture or Discussion
Module 1: Image Processing
Module 2: Introduction to Classes and Objects
Module 3: Object-Oriented Design & Programming - Inheritance
Module 4: Object-Oriented Design & Programming - Encapsulation
Module 5: Object-Oriented Design & Programming - Polymorphism
Module 6: Object-Oriented Design & Programming - Advanced Topics
Module 7: Abstraction
Module 8: Raising Exceptions
Module 9: Debugging and Testing
Module 10: Decorators and Generators
Module 11: Advanced Object-Oriented Programming Concepts II
Module 12: Graphical User Interfaces
Module 13: Final Project

How This Course Works

Course Mode: Face to Face

Description of How this Course Works:

This course is organized into 13 modules by topic, each 1 week long. Students will progress through the course week by week. A typical week includes readings, videos, practice labs, exercises, and programming projects. There are also a midterm and a final exam. The final exam is a programming project.

For this **three (3) credit** course students should expect to spend up to **3 hours/week** reading materials, watching videos or interacting with the instructor and other students online, and up to **6 hours/week** completing assignments outside of class for a total of 9 hours of academic engagement each week. Practice Labs, Exercises, Projects and Exams are *typically* done outside "class" time.

"Expect" when completing assignments means generally "on average". Some students may finish things faster, and some students may take longer. If you are unhappy with your progress, ask for help, and do not cheat. There are no shortcuts when it comes to learning to program and problem solve.

All grades are in Canvas. Any assignments graded in Codio are automatically transferred from Codio to Canvas. This is the same as it was for CS 1400.

Note about Codio: Major assignments such as labs, exercises, and projects are in Codio and accessed through Canvas. **If you attempt to access Codio without going through Canvas, many items will not be correctly accessible and grades will not sync to Canvas correctly.**

How to Be Successful in this Course

Start Early--Think Ahead!

Student Responsibilities:

- Start class the first week of the term.
- **Be accountable by setting aside regular time each week to complete course activities and assignments on time as noted per the due dates**.
- Learn how to use Canvas for assignments, grades and course materials.
- Learn how to use Microsoft Teams to hold video/voice meetings, post chats, and retrieve files. If you have technology-related problems contact the Service Desk.
- Learn to use Discord to: communicate with your instructor and other students, ask questions when you need help sooner rather than later, be part of a community rather than going it alone.
- Abide by ethical standards. Your work must be your own. Do not claim mastery of skills you haven't mastered, nor ask for a grade you have not earned by submitting someone else's work, nor by allowing someone else to submit your work as their own.
- Contact your instructor as early as possible if an emergency arises. Do NOT wait until the last minute to ask for an extension.

Instructor Responsibilities:

- Respond to messages within ONE business day. If multiple messages are received regarding the same question or concern, they may be responded to with an announcement to the entire class.
- Provide timely, meaningful and constructive feedback on assignments.
- Facilitate an effective learning experience.
- Refer students to appropriate services for issues that are non-course content specific. For instance, technical issues, writing labs, accessibility services, etc.
- Mentor students through the course.

Required Course Syllabus Statements

Generative AI

Statement on using ChatGPT (and similar tools) for this class: Use ChatGPT as a Learning Assistant, not an Oracle. ChatGPT can significantly streamline the learning process for students in a programming class, but it's crucial to ensure that it doesn't inadvertently promote dependency or shortcut genuine understanding. The tool can provide immediate help on complex topics and code troubleshooting. Many companies are expecting students to graduate knowing how to use it to generate code, so using it can no longer be considered cheating or plagiarism. Nevertheless students are encouraged to independently problem solve and think critically. These most important skills in programming are best developed through a balance of self-led exploration and learning with an assistant.

Using Remote Testing Software

☒ This course does not use remote testing software.

☐ This course uses remote testing software. Remote test-takers may choose their remote testing locations. Please note, however, that the testing software used for this may conduct a brief scan of remote test-takers' immediate surroundings, may require use of a webcam while taking an exam, may require the microphone be on while taking an exam, or may require other practices to confirm academic honesty. Test-takers therefore shall have no expectation of privacy in their test-taking location during, or immediately preceding, remote testing. If a student strongly objects to using test-taking software, the student should contact the instructor at the beginning of the semester to determine whether alternative testing arrangements are feasible. Alternatives are not guaranteed.

Required University Syllabus Statements

Accommodations/Students with Disabilities

Students needing accommodations due to a permanent or temporary disability, pregnancy or pregnancy-related conditions may contact UVU Accessibility Services at accessibilityservices@uvu.edu or 801-863-8747.

Accessibility Services is located on the Orem Campus in BA 110.

Deaf/Hard of Hearing students requesting ASL interpreters or transcribers can contact Accessibility Services to set up accommodations. Deaf/Hard of Hearing services can be contacted at DHHservices@uvu.edu

DHH is located on the Orem Campus in BA 112.

Academic Integrity
At Utah Valley University, faculty and students operate in an atmosphere of mutual trust. Maintaining an atmosphere of academic integrity allows for free exchange of ideas and enables all members of the community to achieve their highest potential. Our goal is to foster an intellectual atmosphere that produces scholars of integrity and imaginative thought. In all academic work, the ideas and contributions of others must be appropriately acknowledged and UVU students are expected to produce their own original academic work.

Faculty and students share the responsibility of ensuring the honesty and fairness of the intellectual environment at UVU. Students have a responsibility to promote academic integrity at the university by not participating in or facilitating others' participation in any act of academic dishonesty. As members of the academic community, students must become familiar with their rights and responsibilities. In each course, they are responsible for knowing the requirements and restrictions regarding research and writing, assessments, collaborative work, the use of study aids, the appropriateness of assistance, and other issues. Likewise, instructors are responsible to clearly state expectations and model best practices.

Further information on what constitutes academic dishonesty is detailed in UVU Policy 541: *Student Code of Conduct*.

Equity and Title IX
Utah Valley University does not discriminate on the basis of race, color, religion, national origin, sex, sexual orientation, gender identity, gender expression, age (40 and over), disability, veteran status, pregnancy, childbirth, or pregnancy-related conditions, citizenship, genetic information, or other basis protected by applicable law, including Title IX and 34 C.F.R. Part 106, in employment, treatment, admission, access to educational programs and activities, or other University benefits or services. Inquiries about nondiscrimination at UVU may be directed to the U.S. Department of Education's Office for Civil Rights or UVU's Title IX Coordinator at 801-863-7999 – TitleIX@uvu.edu – 800 W University Pkwy, Orem, 84058, Suite BA 203.

Religious Accommodation
UVU values and acknowledges the array of worldviews, faiths, and religions represented in our student body, and as such provides supportive accommodations for students. Religious belief or conscience broadly includes religious, non-religious, theistic, or non-theistic moral or ethical beliefs as well as participation in religious holidays, observances, or activities. Accommodations may include scheduling or due-date modifications or make-up assignments for missed class work.
To seek a religious accommodation, a student must provide written notice to the instructor and the Director of Accessibility Services at accessibilityservices@uvu.edu. If the accommodation relates to a scheduling conflict, the notice should include the date, time, and brief description of the difficulty posed by the conflict. Such requests should be made as soon as the student is aware of the prospective scheduling conflict.

While religious expression is welcome throughout campus, UVU also has a specially dedicated space for meditation, prayer, reflection, or other forms of religious expression.