University of Wisconsin-Milwaukee

CS 250: Introductory Computer Programming

Spring 2026

Lecture Meeting Times:
9:10-10:02 Dates: Jan 28, 2026 - June 16, 2026

Instructor:
Elizabeth Bart
Office: By Appointment
Phone: 262-332-0532
E-mail: ebart@uwm.edu

Location:
Riverside University High School
1615 E. Locust Street
Milwaukee, WI 53211
(414) 906-4900

Course Objectives:
CS 250 aims to teach its students the basics of programming using a structured high-level language. Specifically, a successful student will have the ability to:

- Perform basic input and output.
- Declare and use variables of the primitive types.
- Use control statements including conditionals (if, if-else), and iteration (while, for, do-while).
- Call and write functions using parameters.
- Declare and use arrays.

Other activities include the SkillUSA Drone Competition, which involves four-member teams on professional piloting, technical maintenance, and data analysis skills for careers in the drone industry.

- Focus on safe, efficient flight operations (manual and autonomous)
- payload management
- troubleshooting
- and adhering to FAA regulations.

Key Policies and Student Responsibilities
Workload: This course includes homework assignments, quizzes, exam preparation, and the opportunity to prepare for and take the PCEP Certification exam.

The following is a breakdown of the expected course workload (in hours):

Activity hours spent: 192

- Lecture meetings 30
- Preparation for lectures 30
- Exercise/Lab meetings 30
- Writing daily and weekly programs 60
- Studying for Unit quizzes 20
- Studying for midterm exams 14
- Studying for the final exam 8

Late Policy:
Homework assignments are accepted late only if an extension is granted to the entire class. Students are strongly encouraged to begin assignments early. Technical issues, such as computer system outages, may occur and should be anticipated. Planning for these and other unexpected events is essential.

Homework assignments will not be accepted after the posted due dates without prior approval.

Make-up Policy:
Everyone is expected to complete assignments and complete exams. Failure to attend an exam without a substantial and verifiable reason will not qualify for a make-up exam. Exam dates are listed below, and students are expected to plan accordingly. To be considered for a make-up exam, students must contact the instructor as soon as possible.

Accommodations and Religious Obligations:
Please see the official UWM policy statement for complete descriptions.
http://www.uwm.edu/Dept/SecU/SyllabusLinks.pdf

COVID Considerations: Please see the official UWM policy statement for complete descriptions.
https://uwm.edu/cetl/covid-19-syllabus-statements/

Prerequisite

- Referred by math instructors or
- Math Placement Code of 40 or Math 116(P) or Math 211(P)

To be ready to take CS 250, a student must be comfortable with college algebra (as taught in UWM's Math 116) because basic programming requires writing formulas that are very similar to algebra problems. If you have any questions about this prerequisite, please feel free to talk to your instructor.

Course Materials:
Access to Chromebook or computer
Wi-Fi
Codehs.com
Carnegie Mellon University - CS

Grading:

- Programming Assignments = 10% of the course grade. The lowest single assignment score will be dropped.
- Unit Quizzes = 25% of course grade. The lowest single quiz score will be dropped.

- In-Class Examinations:
    - Midterm I = 15% of the course grade.
    - Midterm II = 15% of the course grade.
    - Final = 25% of the course grade.
- Course letter grades will be assigned using the following scale, unless we decide that this scale is too severe, in which case we will adjust the scale downward.

Letter Grade:	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F
Minimum Score:	92	88	84	80	77	74	70	67	64	60	57	0

Programming Assignments
Programming assignments in this course are completed through CodeHS using the PCEP Certification Unit, which aligns with the Python Institute entry-level Python certification standards. Assignments focus on building foundational programming skills, including variables, data types, input/output, conditionals, loops, and basic problem-solving using Python.

Students will complete interactive coding exercises, practice problems, and assessments designed to reinforce core concepts and prepare them for the PCEP Certification exam daily.

Assignments are scaffolded to support learning progression and emphasize accuracy, logical thinking, and proper coding practices. Regular completion of programming assignments is essential for success in the course and certification readiness.

You are allowed to work together (collaborate) on the programming assignments; you alone are responsible for making sure that you understand not just the specifics of a particular assignment solution, but the general use of the programming constructs used in that solution, which shall be tested on quizzes and examinations.

You should strive to be able to write your programs with minimal to no collaboration with others. Program descriptions, their due dates, and their grading rubrics are posted on Google Classroom.

Unit Exercises
Work is reviewed and stepped through during class time, with units lasting between 2 and 3 weeks. Each unit will have a unit exam.

Quizzes
Quizzes cover each exercise level and will be graded accordingly. cover material from recent lectures and homework assignments. Each quiz will be graded on a 10-point scale.

Examinations
You will take seven (7) unit exams and a final examination, which will be based on the material covered by CodeHS lectures, quizzes, exercises, and practice exams.

The unit examinations will be held at your normal classroom time in room 35. The examinations are closed notes, with tabs locked to the exam on Chromebooks. This is also the same for the final examination.

Academic Misconduct
You can find extensive information on UWM's Academic Misconduct Policy at this URL:
http://www.uwm.edu/CHS/administrationinfo/acadmisc.html

Course Schedule:

Module 1: Basic Python and Console Interaction (3 weeks/15 hours)
Students learn the basics of programming by writing programs that interact with users through the keyboard.

Browse the full content of this unit at https://codehs.com/course/26697/explore/module/38061
Topics Covered
- Printing
- Variables
- Types
- User Input
- Converting Input Types
- Arithmetic Expressions
- String Operators
- Comments
- Fundamental Terms and Definitions

Example Assignments
- Printing
  - Print messages to the console
- Variables
  - Create variables of different types and print them to the console.
- Types
  - Investigate the types of variables
  - Convert between types
- Arithmetic Expressions & Converting Input Types
  - Age in One Year - Ask the user how old they are, and tell them how old they will be in one year
  - Rectangle, part 1 - Make variables for length and width and compute area and perimeter
  - Rectangle, part 2 - Ask the user for length and width, and compute area and perimeter

Module 2: Conditionals (2 weeks/10 hours)
Students teach their programs to make decisions based on the information it receives.
Browse the full content of this unit at https://codehs.com/course/26697/explore/module/38062
Objectives / Topics
Covered
- If Statements
- Boolean Values
- Logical Operators

● Comparison Operators
● Floating Point Numbers and “Equality.”
Example Assignments
/ Labs
● If statements and Boolean values
○ Is it raining? - Write a program that uses a Boolean variable to determine whether or not it is raining
● Boolean operators and expressions
○ Boolean variable - Take a variable and use it in an if statement
○ Legally allowed to vote - User reports age, and the program tells them whether or not they can vote in the US
○ Transaction - The user reports the balance and deposit/withdrawal, and the program prints a new balance or error
○ Recipe - Ask the user for ingredients, amounts per serving, and number of servings, and report the total amount of each ingredient needed

Module 3: Looping (2 weeks/10 hours)
Students learn how to write more efficient code by using loops as shortcuts.
Browse the full content of this unit at https://codehs.com/course/26697/explore/module/38063

Objectives / Topics Covered
● While Loops
● For Loops
● Break and Continue
● Nested Control Structures
● While
● Else Clauses
● Pass Keyword
Example Assignments
/ Labs
● While Loops
○ Divisibility - Ask the user to enter a numerator and denominator, and re-prompt until the denominator is non-zero
● For Loops
○ Average test score - Compute the average of several test scores
● Break and Continue
○ Higher/ Lower - Ask the user to guess a particular number between 1 and 100. If the user’s guess was too high or too low, they should be notified
● Nested Control Structures
○ Rolling Dice - Print out all combinations that can be made when 2 dice are rolled

Module 4: Functions and Exceptions (2 weeks/10 hours)
Students learn how to decompose problems into smaller pieces that work together to solve a problem.
Browse the full content of this unit at https://codehs.com/course/26697/explore/module/38064

Objectives / Topics Covered
● Functions
● Namespaces

● Parameters
● Return Values
● Exceptions
● Exception Hierarchy
● Exception Delegation and Propagation
● Recursion
● Global Keyword
● Shadowing

Example Assignments / Labs
● Example exercises:
○ Functions
■ Raining cats and dogs - Write functions to print text art of a cat and a dog
■ Temperature converter - write functions to convert from Fahrenheit to Celsius and vice versa
○ Exceptions
■ Temperature converter, part 2 - Add exception handling to your temperature conversion program
○ Putting it all together
■ Enter a positive number - Make a function to repeatedly ask the user to enter a number until they enter a positive number

Module 5: Strings (1-2 weeks/5-8 hours)
Students learn more sophisticated strategies for manipulating text in their programs.
Browse the full content of this unit at https://codehs.com/course/26697/explore/module/38066

Objectives / Topics Covered
● Indexing and Slicing
● Math Operators on Strings
● For Loops Over a String
● String Methods

Example Assignments / Labs
● Example exercises:
○ Indexing
■ First character - write a function that takes a string and returns the first character
■ All but the first character - write a function that takes a string and returns everything but the first character
○ Math operators and strings
■ Full name - write a function that takes two strings (a first name and a last name) and returns a full name as a single string
■ Replace a letter - write a function that takes a string and returns a copy with the character at a particular index replaced with a dash
○ For loops on strings
■ Count occurrences - write a function that takes two strings and returns the number of times the second string appears in the first string
○ String methods
■ Add enthusiasm - write a function that takes a string and returns that string in all upper case

■ Remove all from string - write a function that takes two strings and returns a string that consists of the first string with all instances of the second string removed

Module 6: Creating and Altering Data Structures (2 weeks/10 hours)
Students learn how tuples and lists are formed and the various methods that can alter them.
Browse the full content of this unit at https://codehs.com/course/26697/explore/module/38067

Objectives / Topics Covered
● Tuples
● Lists
● For Loops and Lists
● List Methods

Example Assignments / Labs
● Example exercises:
○ Tuples
■ Cookout Orders - Given a tuple of food orders, add up the number of burgers and hotdogs and print the total sums.
○ Lists
■ Listed Greeting - Ask a user to enter their name, age, and favorite sport, then split their response into list elements and use index values to greet them by name and respond that you enjoy that sport as well!
■ Exclamat!on Po!nts - Ask the user for a string and then print the same string with every lowercase i replaced with an exclamation point.
■ Librarian - Ask the user for the last names of the authors of the five books they are returning. Print a list of those namesin sorted order.

Module 7: Extending Data Structures (2 weeks/10 hours)
Students learn to build more complex programs that make use of grids and dictionaries.
Browse the full content of this unit at https://codehs.com/course/26697/explore/module/38068

Objectives / Topics Covered
● Dictionaries
● 2d and 3d lists
● List comprehensions
● Packing and unpacking
● Mutable vs. immutable

Example Assignments / Labs
● Example exercises:
○ Dictionaries
■ Phone book - user repeatedly enters their name, and the program either asks for the person's phone number or reports the phone number already provided
○ 2d lists
■ Checkerboard - write a program that prints the initial setup of a checkerboard, with a 1 where a piece would be and a 0 where a blank square would be

Module 8: Review Problems Students practice answering questions for the PCEP Certification Exam.
Browse the full content of this unit at https://codehs.com/course/26697/explore/module/38132

Objectives / Topics Covered
- Programming and Python Fundamentals
- Control Flow
- Data Collections
- Functions and Exceptions

Drone Prep & Competition:

I - Feb 23- Feb 27: Safety, Regulations, FAA TRUST - Safety Documentation

II - Feb 2 - Mar 6: Drone Build, Anatomy, Maintenance - Technical Knowledge

III - Mar 9 - Mar 15: Manual Flight Proficiencies - Piloting Skills

IV - Mar 16 - Mar 29: Autonomous Coding & Data Mission - Mission Planning

V - Mar 30 - Apr 6: Mock Competitions & Refinement-Final Portfolio Prep

Apr 7 & 8 Madison, WI SkillsUSA Drone Competition

Dates to Remember
(Consult the official UWM/Riverside University High School Calendar for full listings and details.)
* Records day on 1/26/26
* 1/29 - Multicultural Show (Repeat Period 7 - Schedule will be forthcoming). Dress rehearsal in the morning.
* 1/30 - Winter Graduation
* 2/25 - Black History Program Matinee (elementaries)
* 2/27 - Black History Program Day & Evening
* 3/3 - Staff Meeting
* 3/4 - PD Day (Partial)
* 3/4 - Hold Drill (with ICE information)
* 3/11 - ACT Day
* 3/11 - RUHS hosts district Family Financial Literacy Night
* 3/20 - PT Conf Day
* 3/24 - Staff Meeting
* 3/24 - Open House 5:30-7:30 (THIS IS AN UPDATED DATE)
* 3/30 - 4/3 - Spring Break
* 4/7 - Staff Meeting
* 4/9 - Pre ACT Day
* 4/15 - PD Day (Partial)
* 4/17, 4/18 - Evening Musical
* 4/21 - Staff Meeting
* 4/21 -Matinee Musical (elementaries)
* 4/24, 4/25 - Evening Musical

* 5/1 - Prom: Italian Community Center
* 5/12 - Staff Meeting
* 5/19 - Staff Meeting
* 6/4 - Senior Picnic
* 6/8 - Exam Review & Practice Exam
* 6/9 - Certification Exam/Last day of Regular Class
* 6/10-12 - Final Exams
    - Final Exam Room 035 - 6/10 beginning at 9:35 AM: 1.5 hours, Google Form, Closed Tab, Closed Notes
* 6/24 - Graduation (9:00 AM Panther Arena)

Statement on the use of Artificial Intelligence Engines in completing course assignments
This policy should be the same as the policy used for sections of this course when taught by UWM instructors on campus. Include an explicit statement that explains what AI use is and is not allowed in this class. If AI is allowed, include very specific details about how it can be used and how AI use should be cited. See the CETL page on AI, which includes information, sample syllabus statements, and resources. It is also important to talk with students about this policy as you discuss assignment expectations.

Student Workload and Time Expectations
According to UW System policy, students are expected to invest a minimum of 48 hours per semester per credit hour of the course. A 3-credit course requires a minimum of 144 (3 x 48) hours of student work time, and a 4-credit course requires a minimum of 192 hours of student work and a 5-credit course requires at least 240 hours. For this class, students will spend 144 hours in the classroom; students should expect to spend at least 3-4 hours per week outside of class for reading, writing, and review.

UWM Student Resources and Policies

Dual enrollment students have access to some, but not all, of the student resources on campus. Here are some of the resources available:

Student Success Center
- virtual appointments available
- includes course-specific tutoring and one-on-one academic skills sessions

The Writing Center
- virtual appointments available

UWM Libraries
- Need to activate UWM account to access all resources

Complete description of UWM Policies can be found here. Contact UWM's Office of Dual Enrollment for questions about University policies.

Dropping or Withdrawing from the Class

Students will receive an official grade on their UWM transcript for this class. To drop or withdraw from this class, they need to email dual-enrollment@uwm.edu and give UWM consent to unenroll them from the UWM class.

Important Dates for Spring 2026:

- Deadline to Drop without a W on transcript: February 22
- Deadline to Withdraw with a W on transcript: April 12

Academic Misconduct

Because students are enrolled in a UWM class, any academic misconduct will need to be handled both by the high school and by UWM. Students who engage in cheating of any sort may be required to repeat the assignment, get a failing grade for the assignment or course, or be removed from the class. Details here on UWM's policies related to academic misconduct.

Student Accommodations for Dual Enrollment Students

Reasonable accommodations for students with disabilities are a shared faculty and student responsibility. When a student is accepted into UWM's dual enrollment program, the student must contact the Accessibility Resource Center (ARC) to request services and provide the necessary documentation. ARC will not provide modifications if the modifications would alter the essential elements of a course. It is possible that accommodations available to a student in the high school setting will not be allowed in college classes.

Students must work with UWM's Accessibility Resource Center (ARC) to establish an Accommodation Plan for any dual enrollment class. Students, or their high school teacher or counselor, should start by completing the form here. Once the form and disability documentation have been submitted, a UWM ARC Access Specialist will create an Accommodation Plan for the student. The instructor will be notified, via email, of the accommodations for the student in the course. Instructors will work either directly with students or in coordination with the UWM ARC to provide reasonable academic accommodations.

For students who have already established an Accommodation Plan with UWM and are now enrolling in a second class, contact Jon Broskowski, UWM ARC Manager, at bros@uwm.edu to request that the accommodations plan be shared with new teachers.