

CNMT 100

Section 03

Fall 2022

Course Information

Course Meeting Times: MWF 9:00 – 9:50 pm B228 SCI

Course Description: Explore the foundations of modern computing to include the creation of computational artifacts, the Internet, big data, digital privacy and security, algorithms, databases, programming, business Intelligence and the societal impacts of computing.

This section was created for students who have majors within the School of Mathematics, Computing, Physics and Astronomy and the School of Biology, Biochemistry, and Chemistry. This section will focus more on computational thinking, and specifically programming with Python compared to the other sections of the course.

Credits: 3

Prerequisite: none

GEP: none

Textbook & Course Materials

Required Text: Morley, D. (2015). Understanding computers in a changing society. 6th Edition, Stamford, CT: Cengage Learning. ISBN 9781285767710. This is the official text. We will not use it.

Handouts: Distributed during class and posted on Canvas

Instructor Information

Brad Hinaus
Office: B107 Science
Phone: 715-254-5141 (cell)
Zoom Address:
Email: bhinaus@uwsp.edu

Office Hours

You can stop by my office at the times listed above or make an appointment. I warn you that I wander during office hours. Feel free to send me a text if I am not there. If you know you are coming, let me know. If you would like to meet via Zoom, send me a text and we can meet at our class Zoom Address for our meeting.

Instructor Office Hours; B107 SCI

M 2-3

W 2-3

Th 11-12

F 10-11

By Zoom during above hours or by appointment

1. Send me a text on cell phone @7152545141

2. Click to link for meeting. <https://wisconsin-edu.zoom.us/j/8496836849>

My Teaching Philosophy

I think the college classroom should reflect basketball practice or music lessons. Mentally picture what basketball practice looks like or what individual music lessons look like. What do you see? Its active, people (the learners) are moving around and doing things. Players or instrumentalists don't spend 100% of their time watching their coach or teacher draw diagrams on a whiteboard and talk continuously. They spend a good portion of their time working on the skills with each. That is what I want us to do, work on our coding skills during class *with each other*. Will we eliminate the lecture? No, but I hope to reduce the amount of time in that mode so we can practice and ask questions. Most often, we will introduce a topic a lecture, then spend the next 1 or 2 lectures using it in coding.

Here is my philosophy on answering questions you have while working on code. It is similar to the statement, "Give a person a fish, feed them for a day, teach them how to fish and feed them for a life." When a person asks, "Why doesn't my code work?", I will try to ask the person a question about their code so **they can reason** why it doesn't work. This will teach them to be a better thinker, understand their own thinking, and understand the coding process better. Is it efficient? Not in the time scale of class. It is quicker to tell you the answer. But it is more efficient in the time scale of completing a major. One of the best skills you can learn as a programmer is to determine why your code does not work. It usually comes down to one of a few things: a.) correct syntax b.) logical reasoning or c.) an untrue assumption. Sometimes the best way out of jam it to generously use the "print" statement at numerous lines in the code to see if the output from the code matches what you expect it to be.

Inclusivity Statement

It is my intention that students from all backgrounds are well served in this course. Backgrounds can include gender, race, orientation, age, disability, religion, culture, and other ways a person identifies. Other backgrounds that give students various perspectives of this course are their current mathematical abilities, their developing problem-solving abilities, past courses, life experiences growing up, classroom environments they have experienced, preferred learning style and more. In this course, it is expected that each other's thoughts and comments be respectfully listened to and/or responded to

during class, lab, and discussion. There are numerous ways to have a computer solve a problem. It is also expected during the course times, that students work to assist each other in the learning process.

To help you with difficulties of this class, I am available in scheduled office hours and meeting by appointment. In past semester's I have had standing 1-hour meetings with individual students each week. I am welcome to those if you would like to schedule on with me. If you would like to hear a different perspective, the STEM Drop-in tutoring is available in CBB 190 (see below). If you have suggestions for me on how to make this class more inclusive between instructor/student and student/student interactions or activities, please let me know.

STEM Drop-In Tutoring Center Schedule

Tutors for STEM courses are available on a drop-in basis - no appointment needed! The STEM Drop-In Tutoring Center is located in CBB 190, right next to Starbucks. Click on the subject below to expand the section to view tutoring days and times. Tutors are students who have done well, A- or better in classes, so they are knowledgeable about the subject. Drop-in Tutoring is FREE! The schedule will be posted at the following link shortly after the semester begins. <https://www3.uwsp.edu/tlc/Pages/dropInTutoring.aspx>

Three pieces of advice:

1. Find a tutor who thinking meshes with your own.
2. Be sure the tutor is helping you think about it how to solve the problem you are having. The tutor should not "just give you the answer". For new tutors, sometimes they are excited to solve a problem and just solve it. (When I was in graduate school, I was that kind of tutor until someone pointed it out). If you do not understand what the tutor did/explained to you, you have permission ask them to explain it again, or ask them to dream up another question where you need to use the same type of thinking so you can practice it on your own.
3. Your goals for going to the tutor is to understand the material that the assignment is making you practice NOT to finish the assignment without understanding.

Grading Policies

Graded Course Activities

Homework and Projects – 60%

You will have regular homework assignments and projects during this course. These activities are designed to help you practice, learn, and apply the course content. You can expect weekly homework in this course. You will also have several larger projects in which you will design and code the program to solve the problem.

Exams – 40%

There will be two exams in this course. Exams will be given in class . You will be allowed to bring in cheat sheet. Details will be given before the exam. Exams will be designed to test your knowledge and understanding of core course concepts. You will only receive one attempt on exams

Late Work Policy

Be sure to pay close attention to deadlines—you will be able to turn in an assignment late up until the time the instructor grades it.

Any assignment that is turned in after the instructor grades it will receive a reduction of up to 30% of the score at the discretion of the instructor. If students are having difficulties completing the assignments due to factors outside of class (health, family, work, other classes, etc.), they should communicate this to the instructor (email) before the due date to avoid the full deduction in points. If difficulties are coming because of class material, it is suggested that the **student advocate for themselves** and make an appointment with the instructor for further assistance.

Exams must be taken during the scheduled day in class.

Letter Grade Assignment

Letter Grade	Percentage
A	93-100%
A-	90-92.99%
B+	87-89.99%
B	83-86.99%
B-	80-82.99%
C+	77-79.99%
C	73-76.99%
C-	70-72.99%
D+	67-69.99%
D	60-66.99%
F	0-59.99%

Technology

Course Technology Requirements

- View this website to see [minimum recommended computer and internet configurations for Canvas](#).
- If you will be using your personal computer, you will need to install the Python programming language on it. In this class, we will use the Anaconda Navigator to install an editor and the Python 3 programming language. Students are free to install a different editor and suite if they like.

-

UWSP Technology Support

- Visit with a [Student Technology Tutor](#)
- Seek assistance from the [IT Service Desk](#) (Formerly HELP Desk)
 - o IT Service Desk Phone: 715-346-4357 (HELP)
 - o IT Service Desk Email: techhelp@uwsp.edu

University Course Policies

Understand When You May Drop or Withdraw from This Course

It is the student's responsibility to understand when they need to consider unenrolling from a course.

- Dropping the Course: You may drop the course on or **before Sept 15, 2022, with no grade on your official university transcript.** The date of the drop will appear. This is called a "clear drop"
- Withdrawing from the Course: You may withdraw from this class on or **before Nov. 11, 2022** . You will receive a grade of "W" on your transcript. After the first two semesters of enrollment at UWSP, students are allowed a total of only four (4) "W" drops during the balance of their undergraduate career, including summer sessions.
- These are posted on the UWSP [Academic Calendar](#). After this period, a serious and compelling reason is required to drop from the course.
- Serious and compelling reasons includes: (1) documented and significant change in work hours, leaving student unable to attend class, or (2) documented and severe physical/mental illness/injury to the student or student's family.
- [More Info from this University Link](#)

Incomplete Policy

Under emergency/special circumstances, students may petition for an incomplete grade. [Link - University Policy on Incompletes](#)

Inform Your Instructor of Any Accommodations Needed

If you have a documented disability and verification from the [Disability Resource Center \(DRC\)](#) and wish to discuss academic accommodations, please contact your instructor as soon as possible. It is the student's responsibility to provide documentation of disability to the DRC and meet with their counselor to request special accommodation *before* classes start. [Three Steps to Apply for Accommodations](#).

The Disability and Assistive Technology Center is located in 108 Collins Classroom Center and can be contacted by phone at 715-346-3365 (Voice) (715) 346-3362 (TDD only) or via email at datctr@uwsp.edu

Statement of Policy

UW-Stevens Point will modify academic program requirements as necessary to ensure that they do not discriminate against qualified applicants or students with disabilities. The modifications should not affect the substance of educational programs or compromise academic standards; nor should they intrude upon academic freedom. Examinations or other procedures used for evaluating students' academic achievements may be adapted. The results of such evaluation must demonstrate the student's achievement in the academic activity, rather than describe his/her disability.

If modifications are required due to a disability, please inform the instructor and contact the Disability and Assistive Technology Center in 609 ALB, or (715) 346-3365.

UWSP Academic Honesty Policy & Procedures

Student Academic Disciplinary Procedures

UWSP 14.01 Statement of principles

The board of regents, administrators, faculty, academic staff and students of the university of Wisconsin system believe that academic honesty and integrity are fundamental to the mission of higher education and of the university of Wisconsin system. The university has a responsibility to promote academic honesty and integrity and to develop procedures to deal effectively with instances of academic dishonesty. Students are responsible for the honest completion and representation of their work, for the appropriate citation of sources, and for respect of others' academic endeavors. Students who violate these standards must be confronted and must accept the consequences of their actions.

UWSP 14.03 Academic misconduct subject to disciplinary action.

(1) Academic misconduct is an act in which a student:

- (a) **Seeks to claim credit for the work or efforts of another without authorization or citation;**
- (b) Uses unauthorized materials or fabricated data in any academic exercise;
- (c) Forges or falsifies academic documents or records;

- (d) Intentionally impedes or damages the academic work of others;
- (e) Engages in conduct aimed at making false representation of a student's academic performance; or
- (f) Assists other students in any of these acts.

(2) Examples of academic misconduct include, but are not limited to: cheating on an examination; collaborating with others in work to be presented, contrary to the stated rules of the course; **submitting a paper (i.e. code) or assignment as one's own work when a part or all of the paper or assignment is the work of another**; submitting a paper or assignment that contains ideas or research of others without appropriately identifying the sources of those ideas; stealing examinations or course materials; submitting, if contrary to the rules of a course, work previously presented in another course; tampering with the laboratory experiment or computer program of another student; knowingly and intentionally assisting another student in any of the above, including assistance in an arrangement whereby any work, classroom performance, examination or other activity is submitted or performed by a person other than the student under whose name the work is submitted or performed.

Religious Beliefs

Relief from any academic requirement due to religious beliefs will be accommodated according to UWS 22.03, with notification within the first three weeks of class.

Schedule

Week	Monday	Wednesday	Friday
1	No Scheduled Class	Syllabus	Monte Hall Problem
2	Variables, Print, Input	Install Python w/Anaconda	Lab Exercise- Hello World, Strings, Simple Math
3	Strings	Lists	Lab- A "Listed" Phone Number
4	While Loops	While Loops	Lab- Averages with Loops
5	Lab- Calculate Pi	Lab- Calculate Pi	Modules – Random, Math
6	Module – Matplotlib	Lab- Graphing with Matplotlib	Intro to SIR Model of Infectious Disease
7	SIR Model Hand Calc (Graphing and Solve Differential Equations).	Lab- Program SIR	Lab- Graph SIR
8	Test #1	For Loop	For Loops, Lists, Items, and Indices
9	Lab – Sierpinski Gasket (Lab – Sierpinski Gasket	IF Statements
10	Lab – A Roll of the Dice (Loops, if statements, probability)	User Defined Functions	Lab- Rolling Dice Again - Passing Variables and Returning Data
11	Lab – Game with Lame Name	Lab- Game with Lame Name	File Input and Output
12	Lab – Benford’s Law (Using File Input/Output)	Cushion	Thanksgiving Break
13	Vote Counting Methods	Monte Carlo Methods	Lab- Shall We Play a Game – Chutes and Ladders
14	Lab- Shall We Play a Game- Chutes and Ladders	Cushion	Final Project
15	Final Project	Final Project (last day of class)	No Scheduled Class

Test #2 During Final Exam Period. Wed Dec 21 2:45 – 4:45 B228 SCI

