

CPTS 111 Introduction to Algorithmic Problem Solving Syllabus, Spring 2017

Washington State University



Link to course site: <https://piazza.com/wsu/spring2017/cpts111/home>

Instructor Information

Instructor: [Gina Sprint](#), PhD

Office: Sloan 320

Email: gsprint@eecs.wsu.edu

Office Hours: MW 12:00pm - 1:00pm, MW 3:00pm - 4:00pm, and by appointment

Teaching Assistant Information

Lead TA: Yunshu Du

Email: yunshu.du@wsu.edu

Lab section TAs: TBA. Please see the [course website](#).

Course Information

CPTS 111 [3 credits]

Introduction to Algorithmic Problem Solving

[UCORE Quantitative Reasoning](#) [QUAN] Category

Prerequisites

Before taking this course, you need to satisfy the prerequisites listed [here](#).

Description

This course introduces the algorithmic constructs used to solve a wide range of problems. An important component of this is developing an understanding of the ways in which information is stored in a computer, both in terms of fundamental units of information such as an integer and in terms of larger collections of information contained with a data structure such as a list. Along the way you will implement computer code to solve several meaningful problems that involve areas such as encryption and graphics. It is assumed you have no prior programming experience. This course uses the Python programming language.

Course Times and Location

- Section 01: M, W 2:10pm - 3:00pm; Abelson 201
- Section 03: M, W 11:10am - 12:00pm; Sloan 7

Lab Times and Location

- Lab Section 01: W 10:10am - 01:00pm; EME 52
- Lab Section 02: W 03:10pm - 06:00pm; EME 52
- Lab Section 03: W 06:10pm - 09:00pm; EME 52
- Lab Section 04: Th 10:45am - 01:35pm; EME 52
- Lab Section 08: F 05:10pm - 08:00pm; EME 52
- Lab Section 09: Tu 10:45am - 01:35pm; EME 52
- Lab Section 10: Tu 01:45pm - 04:35pm; EME 52
- Lab Section 07: Tu 04:45pm - 07:35pm; EME 52

Learning Outcomes

Programming is an art that can take a lifetime to completely master. This class intends to set students on the path to programming enlightenment, and by the end of the term, it is expected that you will be able to:

1. Decompose a problem into a series of algorithmic steps (pseudocode)
2. Convert pseudocode into Python code
3. Know the proper styling guidelines for Python and follow these guidelines in your own code
4. Differentiate between the different Boolean logic operators and use Boolean logic in your programs
5. Write complex conditional statements that contain one or more Boolean operations
6. Identify when a loop is appropriate, when to use both `while` and `for` loops, and write applications that use these looping constructs
7. Understand the importance of lists and dictionaries in programming and identify when it would be beneficial to use either
8. Be able to perform basic string processing and manipulation
9. Leverage your knowledge of programming to answer complex, real-world problems

Additionally, this course targets the following [UCORE QUAN-designated](#) learning goals:

1. Define, analyze, and solve problems
2. Access information effectively and efficiently from multiple sources
3. Convert relevant information into various mathematical forms
4. Understand and apply quantitative principles and methods in the solution of problems
5. Make judgments and draw appropriate conclusions based on quantitative analysis

ABET Outcomes

The following are WSU and ABET [outcomes](#) that will be satisfied in this course. Please refer to <http://school.eecs.wsu.edu/undergraduate/ABET> for more information.

Using ABET terminology, this course targets the following ABET learning goals:

- Setting up and using Python. [i]
- Computers and binary representation of information. [a, i]
- Literals and different data types (int, float, str). [a, b, c, i]
- Variables and identifiers. [a, b, c, i]
- Arithmetic operations and precedence. [a, b, c, i]
- Input and output. [a, b, c, i]
- Assignment and simultaneous assignment. [a, b, c, i]
- Functions, parameter assignment, return values, and scope. [a, b, c, i]
- Introduction to classes and objects. [a, b, c, i]
- Lists and data structures. [a, b, c, i]
- Multidimensional data structures (lists of lists). [a, b, c, i]
- Definite and indefinite loops. [a, b, c, i]
- References. [a, b, c, i]
- Strings and ASCII. [a, b, c, i]
- Boolean variables and Boolean expressions. [a, b, c, i]
- Conditional statements and relational operators. [a, b, c, i]
- Dictionaries (hashes). [a, b, c, i]

Course Materials

Course Website

<http://piazza.com/wsu/spring2017/cpts111/home>

Course Schedule

<http://www.eecs.wsu.edu/~gsprint/cpts111/schedule.html>

Communication

We will use [Piazza](#) and [Blackboard](#) in this course. Piazza will be used to ask questions and discuss topics with other students in the class, TAs, and the instructor. **Piazza will not be used to share code solutions.** You can however post high level code explanations and/or snippets of pseudocode. I will also email important information to you through Piazza.

[Blackboard](#) will be used to submit assignments and view your grades. The website URL is: <https://learn.wsu.edu/> (use your WSU network ID and password to login).

Textbooks

Required Textbook

Programming in Python 3 (a [zyBooks](#) book). This is an online interactive textbook. Follow these instructions to gain access to the book:

1. Sign up at zybooks.zyante.com
2. Enter zyBook code WSUCptS111SprintSpring2017
3. Click Subscribe

The cost to subscribe is \$48; any applicable returning student discounts will be applied automatically. The student subscriptions will be valid through 05/25/17.

Supplementary Textbooks

[How to Think Like a Computer Scientist: Interactive Edition](#) by Downey, Elkner and Meyers.

This is a free, interactive textbook available online under the Creative Commons license. You can access the book [here](#). Recommended reading for this book is included in the course schedule. I highly recommend you read/interact with this book, it is a great resource for learning Python.

Also, there are several other Python books, online practice problems, and resources listed on the [course website](#) under Resources.

Required Hardware

A laptop adhering to EECS requirements. Please refer to <https://school.eecs.wsu.edu/?s=laptop>.

Required Software

[Anaconda3](#) Python3 distribution. This is a free distribution of Python version 3 available for Windows, OS X, and Linux. You can download Anaconda3 [here](#) and view the installation instructions [here](#).

Course Grading

This is an *active learning class*. You are expected to actively attend and participate in lectures and laboratories regularly, and to participate regularly in the Piazza discussions. Active participation may positively affect your grade at the end of the semester.

Assignment Weights

- Micro assignments (10%)
- Programming assignments (30%)
- Labs (20%)
- Two midterm exams (20%, 10% each)
- One final exam (10%)
- zyBooks participation activities (5%)
- zyBooks challenge activities (5%)

Assignment Categories

Micro Assignments

Micro assignments (MAs) ensure that you are attending lecture, staying on top of the textbook reading assignments, and understanding the material. MAs may be graded for participation and/or correctness. MAs can take on the following forms:

1. Short assignments completed in lecture.

- *Nearly every lecture will have an MA.* You can think of in-class MAs as a form of lecture attendance.
2. Short assignments completed outside of lecture/lab.
 - CptS 111 is a UCORE course so there will be writing (non-programming) MAs.

Note: **I will drop your 4 lowest MA scores.** This means that you are given 4 MA "freebies" that excuse your failure to turn in a MA *for any reason*.

Programming Assignments

You will be given 6 programming assignments (PAs) to complete. All Python code written in assignments must adhere to the recommended [Python Style and Coding Standards](#); your TA will let you know if you are not adhering to these standards.

I want to emphasize that **PAs should be completed independently**. It is easy to detect code sharing, don't do it.

Please upload assignments as attached .zip files to the corresponding assignment in Blackboard. Recall the site is <https://learn.wsu.edu/>. Please refer to the [schedule](#) for assignments and dates. All programming assignments must be submitted by midnight of the due date.

Labs

You will be given 14 labs to complete. Each lab will give you hands-on experience with the Python programming language to solve multiple problems. These lab assignments also give you the opportunity to put into practice the techniques and concepts covered in the lessons with the help of your peers and a knowledgeable teaching assistant (TA). You are encouraged to share ideas with your peers in lab! Take advantage of learning from each other. You will receive full credit for a given lab if both:

1. You show up and actively participate in the entire lab by making a sincere effort to complete all of the assigned problems.
2. You make a sincere effort to assist other students with the lab in the event that you finish before them.

Hence, you are awarded credit in lab for participation and effort. You may make up some labs if you have an excuse that is acceptable as determined by the instructor. You are responsible for ensuring you receive appropriate credit from your TA for these special circumstances.

Exams

We will have two midterm exams and one final exam in this course. The final exam will be held on:

- *Section 01:* Monday, May 1st; 8:00 – 10:00am
- *Section 03:* Tuesday, May 2nd; 3:10 – 5:10pm

Exams may be rescheduled for students that have valid excuses. To increase your chances of your excuse being determined "valid", notify the instructor no less than two days in advance if you are going to miss an exam

Note: As per [university policy](#), final exams may not be taken early. You must take your final exam at the time listed above for the course section you are enrolled in.

zyBooks Textbook Activities

You will be graded for completing participation and challenge activities in the zyBooks textbook. Completing at least 90% of the activities constitutes full credit for the assigned zyBooks chapters. See the [course schedule](#) for the schedule of assigned zyBooks chapters and due dates.

Grading Scale

Percentage	Grade
94-100%	A
90-93.99%	A-
86-89.99%	B+
82-85.99%	B
78-81.99%	B-
74-77.99%	C+
70-73.99%	C
66-69.99%	C-
62-65.99%	D+
58-61.99%	D
0-57.99%	F

Contesting a Grade

If you believe that a mistake has been made with grading an assignment or exam, please *speak with the instructor or TA within one week of the assignment or exam being returned*. Do not wait until the end of the semester to discuss any grade changes. You need to constantly be aware of how you are performing in the class. Thus, there will not be any surprises at the end of the semester when grades are to be formally submitted. You should be able to view your grades via Blackboard. These will be updated weekly.

Note: the grades in Blackboard may just be raw scores and not be weighted according to the ones described here. Thus, be sure that you keep track of these weights so that you are not surprised by a change in your overall grade at the end of the semester.

Course Policies

Late Work

All assignments are due by the established due dates and times (see [schedule](#)).

Specifically for programming assignments, you may turn in a PA up to two days late (the weekend counts as one day), at a penalty of 10% per 24 hours late. Forty-eight hours after the assignment is due, you may no longer hand in the PA for credit.

If an emergency occurs, the instructor will accommodate the student as much as possible. Make-up exams will not be possible unless the student speaks with the instructor at least two days in advance. Emergencies do occur and rescheduling of exams because of these is to be determined by the instructor.

EECS Computing Policies

You may read about EECS's school computing policies at the following site <https://school.eecs.wsu.edu/documents/2016/08/eecs-computing-policies.pdf>.

Academic Integrity Policy

University Policy

Academic integrity is the cornerstone of higher education. As such, all members of the university community share responsibility for maintaining and promoting the principles of integrity in all activities, including academic integrity and honest scholarship. Academic integrity will be strongly enforced in this course. Students who violate WSU's Academic Integrity Policy (identified in Washington Administrative Code (WAC) 504-26-010(3) and -404) will fail the course, will not have the option to withdraw from the course pending an appeal, and will be reported to the Office of Student Conduct. Cheating includes, but is not limited to, plagiarism and unauthorized collaboration as defined in the Standards of Conduct for Students, WAC 504-26-010(3). You need to read and understand all of the definitions of cheating: <http://app.leg.wa.gov/WAC/default.aspx?cite=504-26-010>.

You are expected to know and understand Washington State University Academic Integrity Policies. Copying and plagiarism of other sources will result in an automatic F on the assignment. For a second offense, an automatic F in the course will be awarded without the option to withdraw. University authorities will be notified and the proper procedures will be followed. Possible outcomes resulting from violating these policies include denial of certification into your program and expulsion.

If you have any questions about what is and is not allowed in this course, you should ask course instructors before proceeding. If you wish to appeal a faculty member's decision relating to academic integrity, please use the form available at conduct.wsu.edu.

Please also review the following website for more information about the EECS academic integrity policy (if any discrepancies are found within the EECS and WSU academic integrity policies, always follow the WSU policy): <http://www.eecs.wsu.edu/~schneidj/Misc/academic-integrity.html>

CPTS 111 Policy

For this course both group and individual work will be required. Group work includes:

1. Labs
2. zyBooks activities
3. Some MAs

Individual work includes:

1. PAs
2. Exams/quizzes
3. Some MAs

All individual work must be completed alone. **Do NOT work with any team members on individual assignments.** You may discuss ideas with team members about problems related to individual assignments, but do not discuss implementation details. Discussing implementation details includes (but is not limited to):

1. Copying/taking a picture of another student's code/work
2. Letting another student copy/take a picture of your code/work
3. Sending your code/work to another student (i.e. digitally or in print)
4. Receiving another's student code/work (i.e. digitally or in print)

Note: If you use content from sources other than the ones provided by the instructor (e.g. textbook, notes, etc.), cite the source in your code.

Here are few example scenarios (adapted from Aaron Crandall's CptS 223 Syllabus):

1. You and a few friends are working together on an individual assignment. You just finished the assignment but have to head to class. One of your friends asks you for a copy of your work so that he/she can use it as a guide. Wanting to be a good friend, you email him/her a copy of your assignment on the promise that he/she won't copy.
 - Is this cheating? YES! You are violating implementation detail #3 (above) and your friend is violating implementation detail #4.
2. You and a few friends are working together on an individual assignment. You just finished the assignment but have to head to class. One of your friends asks to meet up with you later for some help. You meet up with him/her after class and answer his/her questions with high-level descriptions on a white board (e.g. "here is an example of how a for loop works") and pictures (e.g. a drawing).
 - Is this cheating? No, this kind of working together is encouraged.
3. Your friend is almost done with his individual assignment but is stuck debugging. You take a look at his/her program and quickly realize that his/her loop is off by one. You explain how to fix the error to your friend.
 - Is this cheating? No, this kind of working together is encouraged.
4. On Piazza, a student has asked for code help on an individual assignment. You respond with a short code snippet.
 - Is this cheating? No, this kind of working together is encouraged.
5. You are stuck on an individual assignment, say coding a Hangman game, and turn to Google to help you out. You find code that completely implements the game of Hangman! You download the code, modify it to meet the assignment requirements, and turn it in.
 - Is this cheating? YES! This is plagiarism.

If are unsure of whether a situation might be considered cheating, be cautious and don't do it. If help is required please ask the instructor or TA for guidance. We are always more than willing to help!!

Classroom Safety Statement

Classroom and campus safety are of paramount importance at Washington State University, and are the shared responsibility of the entire campus population. WSU urges students to follow the "Alert, Assess, Act," protocol for all types of emergencies and the "[Run, Hide, Fight](#)" response for an active shooter incident. Remain ALERT (through direct observation or emergency notification), ASSESS your specific situation, and ACT in the most appropriate way to assure your own safety (and the safety of others if you are able).

Please sign up for emergency alerts on your account at MyWSU. For more information on this subject, campus safety, and relate topics, please view the [FBI's Run, Hide, Fight video](#) and visit the [WSU safety portal](#).

WSU's AWARE Network

As a student you have many responsibilities and obligations. One of the most important obligations that you may have is to your friends and peers at WSU. If you feel like one of your friends or peers is struggling with academics because of physical or mental health please inform me and/or appropriate university personnel. For more information refer to <http://aware.wsu.edu/>

Documented Disabilities

Washington State University has facilities and resources available for accommodating students with disabilities. Please notify the instructor during the first week of class of any accommodations needed for the course.

Pullman Campus Syllabus Statement

Students with Disabilities: Reasonable accommodations are available for students with a documented disability. If you have a disability and need accommodations to fully participate in class, please either visit or call the [Access Center](#) (Washington Building 217; 509-335-3417) to schedule an appointment with an Access Advisor. All accommodations MUST be approved through the Access Center. You may also contact the Access Center via email: Access.Center@wsu.edu.

Final Notes

Free Tutoring

- The Voiland College of Engineering and Architecture offers free tutoring for students in CptS 111. For more information and the tutor schedule, see <https://vcea.wsu.edu/tutors/>.
- The WSU ACM offers free tutoring for students in CptS 111. For more information, see the [WSU ACM website](#).

Advice

The fastest (and easiest) way to get better at programming is to program! This course moves fast and builds on itself. If you don't stay on top of the assignments, labs, and reading it will be difficult to catch up. You will do great in the course if you put the time in to learn the material on schedule and practice programming :)