CityUniversity of Seattle

Syllabus

SCHOOL OF TECHNOLOGY & COMPUTING

CS160 WEB PROGRAMMING - PYTHON

5 Credits
Effective Date: Spring 2020

Access to the Internet is required.
All written assignments must be in Microsoft-Word-compatible formats.
See the library's APA Style Guide tutorial for a list of resources that can help you use APA style.

FACULTY

Faculty Name: Jin Chang, Ph.D.

Contact Information: changkyong@cityu.edu

[INSTRUCTOR MAY INSERT PERSONAL MESSAGE IF DESIRED]

COURSE DESCRIPTION

The purpose of this module is to help students establish basic competency in Python. Python's diversity, adaptability and its elegant, easy-to-master basics set it apart from other programming languages, making it popular for use in web development, machine learning, data science, scientific computation, and cloud infrastructure. Students will study the concepts of Python, practice Python programming skills, and apply Python programming to diverse applications in database, web, networking, data science, and AI.

COURSE RESOURCES

Textbook

- Matthes, E. (2019). Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 2nd Edition. No Starch Press. (9781593279288)
- Sweigart, A. (2015). Automate the Boring Stuff with Python: Practical Programming for Total Beginners. No Starch Press. (ISBN 9781593275990)

Reference

- RealPython. https://realpython.com/python3-object-oriented-programming/
- Python Objects and Class. https://www.programiz.com/python-programming/class

CITYU LEARNING GOALS

This course supports the following City University learning goals:

- Professional competency and professional identity

COURSE OUTCOMES

In this course, learners:

- Analyze and explain the behavior of simple programs in Python involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, classes, and recursion.
- Write programs that use primitive data types in Python.
- Modify and expand short programs in Python that use standard conditional and iterative control structures and functions.
- Design and implement programs in Python that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing.
- Write programs that uses arrays, strings, Lists, Tuples, Dictionaries and Set.
- learn to work with files and error handling
- How to explore data through visual representation.

CORE CONCEPTS, KNOWLEDGE, AND SKILLS

Topics include:

- Variables and Simple Data Types
  - Variables
  - Strings
  - Numbers
  - Comments
- User Input and While Loop
  - How the input () Function Works
  - Introducing while Loops
  - Using a while Loop with Lists and Dictionaries
- if Statements
  - Conditional Tests
  - if Statements
  - Using if Statements with Lists
  - Styling Your if Statements
- Lists and Tuples
  - Changing, Adding, and Removing Elements
  - Organizing a List
  - Looping Through an Entire List
  - Avoiding Indentation Errors
  - Making Numerical Lists
  - Working with Part of a List
  - Tuples
- Dictionaries
  - A Simple Dictionary
  - Working with Dictionaries
  - Looping Through a Dictionary
  - Nesting
- Functions
  - Defining a Function
  - Passing Arguments
  - Return Values
  - Passing a List Passing an Arbitrary Number of Arguments
  - Storing Your Functions in Modules
  - Styling Functions
- Files and Exceptions
  - Reading from a File
  - Writing to a File
  - Exceptions
  - Storing Data
- Classes
  - Creating and Using a Class
  - Working with Classes and Instances
- Object Oriented Programming
  - Inheritance: Single, Multiple, Multilevel
  - Polymorphism
  - Encapsulation: Method Overloading, Method Overriding, Operator Overloading
- Web scrapping & Spreadsheets

- Image manipulation
  - Pillow module
  - Pyautogui
- Data Visualization
  - Matplotlib
  - Line graph
  - Scatter
  - Automatic Calculation
  - Random Walks

OVERVIEW OF COURSE GRADING

The grades earned for the course will be derived using City University of Seattle's decimal grading system, based on the following:

OVERVIEW OF REQUIRED ASSIGNMENTS	% OF FINAL GRADE	POINTS
The Muddiest Point (MP)	5%	50 = 5 points * 10 modules
Hands-On-Practice (HOP)	30%	300 = 30 points * 10 modules
Programming Exercise (PE)	30%	300 = 30 points * 10 modules
Knowledge Check (KC)	10%	100 = 10 points * 10 modules
Team Project (TP)	25%	250 points= Final document (1-2 page) & Python Code with comments
TOTAL	**100%**	**1,000 points**

The following approaches are used for developing this course content:

Assessment

- Summative Assessment. https://en.wikipedia.org/wiki/Summative_assessment
- Formative Assessment. https://en.wikipedia.org/wiki/Formative_assessment

Classroom Assessment Techniques

- The Muddiest Point. https://en.wikipedia.org/wiki/Classroom_Assessment_Techniques

Active Learning. https://en.wikipedia.org/wiki/Active_learning

- Flipped Classroom. https://en.wikipedia.org/wiki/Flipped_classroom
- Just-in-time Teaching (JiTT). https://en.wikipedia.org/wiki/Just-in-time_teaching
- Peer Instruction. https://en.wikipedia.org/wiki/Peer_instruction**)**

Learning Theory

- Learning-by-doing. https://en.wikipedia.org/wiki/Learning-by-doing
- Project-Based Learning (PBL). https://en.wikipedia.org/wiki/Project-based_learning
- Social Learning. https://en.wikipedia.org/wiki/Social_learning_(social_pedagogy)

Evidence-Based Practice (EBP). https://en.wikipedia.org/wiki/Evidence-based_practice

- Pair Programming. https://en.wikipedia.org/wiki/Pair_programming
- Stand-up Meeting. https://en.wikipedia.org/wiki/Stand-up_meeting
- Agile Software Development. https://en.wikipedia.org/wiki/Agile_software_development

SPECIFICS OF COURSE ASSIGNMENTS

The instructor will provide grading rubrics that will provide more detail as to how this assignment will be graded.

The Muddiest Point (MP)

Before class, students are required to finish the muddiest point activity. This activity is designed to stimulate student engagement in class. Also, the instructor uses feedbacks of the Muddiest Point in preparation for the classroom lecture to implement Just-in-Time Teaching (JiTT). This activity consists of writing a brief reflective essay (<= 50 words) in which students identify the most confusing part (i.e. the muddiest point) of the content covered in the upcoming module. If you do not have MP, you can explain the most interesting point. In addition, students will answer one multiple choice question from the required reading to determine students' grasp of core concepts.

Components	***% of Grade***
Quality Participation: Meets requirements in a timely manner	60%
Writing: Is clear, concise, and grammatically correct.	20%
Accuracy: Answers quizzes correctly	20%
TOTAL	**100%**

Concept Test (CT)

In class, students may be required to answer questions called Concept Tests, which allows peer to teach other, i.e. Peer Instruction. 1) Instructor poses question based on students' responses to their pre-class reading. 2) Students reflect on the question. 3) Students commit to an individual answer. 4) Instructor reviews student responses without giving the correct answer to the students. 5) Students discuss their thinking and answers with their peers. 6) Students then commit again to an individual answer. 7) The instructor again reviews responses and decides whether more explanation is needed before moving on to the next concept. Any participating students will earn their 100% grade.

Components	***% of Grade***
Quality Participation: Meets requirements in a timely manner	100%
TOTAL	**100%**

Discussion Board (DB)

All classes are required to use the Discussion Board. Participation through DB is an integral part of this course and is defined as active engagement in a discussion or other activity. Instructors will determine the type of activities and their due dates; moreover, different DB activities will have different guidelines with regard to substance and length. The instructor will provide specific instructions to students.

A discussion question or topic from the instructor appears weekly in the Discussion Board. Students are to post their answers as well as responses to two other students' responses in the DB by the end of each module. The DB is to help promote student to student engagement. The instructor may not respond to each posting. Questions or comments that are specifically for the instructor, should be emailed directly to the instructor or posted in the Question and Answer Forum. Students who want to talk with other students about issues unrelated to the discussion forums should use the Coffee Talk Forum.

Although the tone of your DB postings can be informal, your instructor will expect the content to be on a professional level. In other words, your comments and questions for discussion should be clear and thoughtful, with correct grammar, spelling, and punctuation. As with written assignments, the quality of your discussion postings will be graded on both content and presentation.

Components	*% of Grade*
Quality Participation: Meets requirements in a timely manner	80%
Writing: Is clear, concise, and grammatically correct.	20%
TOTAL	**100%**

Hands-on Practice (HOP)
The instructor may assign hands-on practice exercises to a pair of students in class or individually in online. Students will learn and practice either specific tools or languages pertinent to their course.

Components	***% of Grade***
Quality Participation: Meets requirements in a timely manner	100%
TOTAL	**100%**

Programming Exercise (PE)
The students must individually perform the programming exercise which is based on the topics and Hands-on Practice. No code sharing or copying form other sources are allowed. Non-executable programs will be not graded. The programs in poor coding styles will be asked to be resubmitted. Please note that copying a segment of code from the Internet and submitting as your work is considered as plagiarism.

Components	***% of Grade***
Quality Participation: Meets requirements in a timely manner	80%
Comments in the programming	20%
TOTAL	**100%**

Python Team Project (TP)
The students are required to write a Tic-Tac-Toe (3x3) program. For this project, the student is going to build the classic game tic-tac-toe in Python. In this game, one player is X by a user and the other player is O. You can imagine that the game will be played by two individuals. The person who will start should be assigned randomly provided the program prior to start. Each run, the program should display the board (3x3) as shown below. Placement is made based on the positional number. Your program should check if there is a winner, and print the winner then exits. One other requirement is to store the move of each X and O to be saved in a file called tictactoe.txt (X:5 O:2 X:1 O:9 etc...). Please note that copying a segment of code from the Internet and submitting as your work is considered as plagiarism.

```
  1   |   2   |   3
------+-------+------
  4   |   5   |   6
------+-------+------
  7   |   8   |   9
```
(Example of Tic-Tac-Toe board and its position)

Components	***% of Grade***
Meets requirements in a timely manner	70%
Documents (comments in the programming and 1-2 pages of word document of the programming logic)	30%
TOTAL	**100%**

Team Project (TP)

Each student can select his or her own team. Each team consists of three students. A team of less than three students requires instructor's approval. Each team will use an instructor approved topic relevant to the course. For effective project management, an agile software development process, Scrum, is used.

The paper must be no less than 6-7 pages. We required you to use the paper template from EDSIG/CONISAR, which is the international conference standard. *The best papers in this course may be submitted to conferences with your team's approval and the instructor's recommendation and revisions. A paper submission is optional and has nothing to do with your course grade*.

As in any scholarly writing, students should not merely copy information from another author, but use evidence to support the contentions they have drawn from their findings and critically analyze related literature - each paper needs to be an analytical paper, not a summary of readings.

Three submissions are required according to the following schedule:

- Proposal (1 page; 30 points) - Starting (Module 1) & Ending (Module 3)
- Progress (3-4 pages; 70 points; graded only after the proposal has been submitted) - Starting (Module 4) & Ending (Module 7)
- Final (6-7 pages; 100 points; graded only after the progress has been submitted) - Starting (Module 8) & Ending (Module 10)

Students are expected and encouraged to use the assigned readings, videos, and other materials used throughout the quarter on this project. Students will need to utilize additional sources that were not assigned by the professor. While stylized after an industry report; nonetheless, students are expected to employ APA formatting of citations, footnotes, and bibliography. Students must cite the sources of all ideas, facts, and information used that are not their own, even if they have put the information into their own words. Failure to do so is plagiarism, even if the oversight is unintentional.

Components	***% of Grade***
Structure: Consists of the required report elements.	5%
Content: Demonstrates critical analysis and synthesis of concepts.	30%
Reference: Is pertinent to the topic and cited properly.	5%
Writing: Is clear, concise, and grammatically correct.	10%
Visual Presentation: Is well designed, legible, and persuasive.	30%
Team Collaboration: Is based on peer review	20%
TOTAL	100%

Knowledge Check (KC)
Students will complete weekly quizzes that are from the course content to reflect on what they have learned in the course. Completing all KCs will help ensure that you successfully master the concepts in this course. The best way for you to gain a thorough understanding of the underlying concepts is to apply those concepts to solve the quizzes. You should focus on the underlying principles, rather than just memorizing information.

Components	***% of Grade***
Accuracy: Answers questions correctly.	100%
TOTAL	**100%**

COURSE POLICIES

Late Assignments

20% deduction from graded score per submission.

Participation

PARTICIPATION

Professional Writing
Assignments require error-free writing that uses Standard English conventions and logical flow of organization to address topics clearly, completely, and concisely. CityU requires the use of APA style.

UNIVERSITY POLICIES
You are responsible for understanding and adhering to all of City University of Seattle's academic policies. The most current versions of these policies can be found in the University Catalog that is linked from the CityU Web site.

Scholastic Honesty
Scholastic honesty in students requires the pursuit of scholarly activity that is free from fraud, deception and unauthorized collaboration with other individuals. You are responsible for understanding CityU's policy on scholastic honesty and adhering to its standards in meeting all course requirements. A complete copy of this policy can be found in the University Catalog in the section titled *Scholastic Honesty* under *Student Rights & Responsibilities*.

Attendance
Students taking courses in any format at the University are expected to be diligent in their studies and to attend class regularly.

Regular class attendance is important in achieving learning outcomes in the course and may be a valid consideration in determining the final grade. For classes where a physical presence is required, a student has attended if s/he is present at any time during the class session. For online classes, a student has attended if s/he has posted or submitted an assignment. A complete copy of this policy can be found in the University Catalog in the section titled *Attendance Policy for Mixed Mode, Online and Correspondence Courses*.

SUPPORT SERVICES

Disability Resources
If you are a student with a disability and you require an accommodation, please contact the Disability Resource Office as soon as possible. For additional information, please see the section in the University Catalog titled *Students with Special Needs* under *Student Rights & Responsibilities*.

Library Services
CityU librarians are available to help you find the resources and information you need to succeed in this course. Contact a CityU librarian through the *Ask a Librarian* service, or access library resources and services online, 24 hours a day, seven days a week.

Smarthinking
As a CityU student, you have access to 10 free hours of online tutoring offered through Smarthinking, including writing support, from certified tutors 24 hours a day, seven days a week. Contact CityU's Student Support Center at help@cityu.edu to request your user name and password.