DICE 4000: Everyday Coding

Fall 2017

Course information
Credits: 5
Location: Online

Instructor information
Instructor: Dr. Sarah Berry
Email: berrysa@seattleu.edu
Office hours: by appointment

Course Description:
In this course, students learn to apply computational thinking and key concepts to programming in Python. Topics include variables, data structures, loops, conditionals, logical flow, and object-oriented programming. Students reflect upon the application of programming strategies for a range of purposes (web development, data visualization, and game development) and produce a fully-functioning game with scoring. Students will collaborate online to troubleshoot shared problems and are encouraged to work synchronously on weekly exercises for mutual aid.

Pre-requisites:
Key Concepts in Computing and Foundations of Digital Rhetoric

Required textbook (available online – buy the 2nd edition, it's cheaper than the new one)

Eric Matthes, *Python Crash Course: A Hands-on, Project-based Introduction to Programming*. No Starch Press 2016.

Required video tutorials (these are embedded in our Canvas modules)

Python 3.4 Programming Tutorials
thenewboston (Bucky Roberts) via YouTube standard license. Most of these are high-definition, so make sure to select that setting for playback, or the code is really hard to read.

Course Student Learning Outcomes:
Upon the successful completion of this course, you should be able to

1. Apply computational thinking to Python programming exercises
2. Create programs that demonstrate a working knowledge of coding best practices
3. Analyze the ways programming conventions reflect modes of production and the need to communicate clearly with other programmers
4. Reflect on the "Zen of Python" as a set of programming values and community-driven ethics
5. Develop intellectual resilience in the process of researching and debugging programming challenges.

E-Portfolios: A final game developed in Python will be suitable for inclusion in the DICE program portfolio.

Week 1: Introducing Python

Topics explored in this week's presentation

- Installing and running Python

- Setting up Sublime Text or PyCharm
- Review: program types and variables
- Numbers and strings in Python

What to read/watch this week

- Chapters 1 and 2 of your textbook: “Getting Started” and “Variables and Simple Data Types

Course startup to-do list

- Review the course syllabus and assignment descriptions.
- Check your SU email for invitations to our course Slack channels.
- Watch the video demo in Module 1 on how to join and use Slack and Appear.in.
- Email me (berrysa@seattleu.edu) with three possible bi-weekly (every other week) video chat days/times. These will take 15-20 minutes and require an internet connection.
- Block out time on your calendar to complete weekly exercises. Share your schedule on Slack and look for others you can work synchronously with, using Slack or Appear.in to help each other (using the 20 minute rule and hints rather than copying code. See the Weekly Exercise description below for rule details).

Exercises to post in Canvas:

2-1: Simple Message
2-2: Simple Messages
2-3: Personal Message
2-4: Name Cases
2-5: Famous Quote
2-6: Famous Quote 2
2-7: Stripping Names
2-8: Number Eight
2-9: Favorite Number
2-10: Adding Comments
2-11: Zen of Python

Week 2: Python data structures: Lists

Topics explored in this week's presentation

- What is a list?
- Modifying and organizing lists
- Looping through lists
- Review: functions
- Making numerical lists with the range() function
- Slicing and duplicating lists
- Using Tuples
- The importance of indentation and code style

What to read/watch this week

- Chapters 3 and 4, “Introducing Lists” and “Working With Lists.”

Exercises to post in Canvas:

3-1 Names
3-2 Greetings
3-3 Your Own List
3-4 Guest List
3-5 Changing Guest List
3-6 More Guests
3-7 Shrinking Guest List
3-8 Seeing the World
3-9 Dinner Guests
3-10 Every Function
4-1 Pizzas
4-2 Animals
4-3 Counting to Twenty
4-4 One Million
4-5 Summing a Million
4-6 Odd Numbers
4-7 Threes
4-8 Cubes
4-9 Cube Comprehension
4-10 Slices
4-11 My Pizzas, Your Pizzas
4-12 More Loops
4-13 Buffet

Week 3: Conditional loops and dictionaries

Topics explored in this week's presentation

- Review: conditions and control flow
- If statements
- Working with Dictionaries
- Looping through a Dictionary
- Nesting Dictionaries

What to read/watch this week

Chapters 5 and 6, “If Statements” and “Dictionaries.”

Exercises to post in Canvas

5-1 Conditional Tests
5-2 More Conditional Tests
5-3 Alien Colors #1
5-4 Alien Colors #2
5-5 Alien Colors #3
5-6 Stages of Life
5-7 Favorite Fruit

5-8 Hello Admin
5-9 No Users
5-10 Checking Usernames
5-11 Ordinal Numbers
5-12 Styling if statements
5-13 Your Ideas
6-1 Person
6-2 Favorite Numbers
6-3 Glossary
6-4 Glossary 2
6-5 Rivers
6-6 Polling
6-7 People
6-8 Pets
6-9 Favorite Places
6-10 Favorite Numbers
6-11 Cities
6-12 Extensions

Week 4: User input and while loops

This week's presentation will explore the following topics

- Review: functions
- The input() function
- While loops
- While loops with Lists and Dictionaries
- Break and Continue statements

What to read/watch this week

- Chapter 7 “User Input and While Loops.”

Exercises to post in Canvas

7-1 Rental Car
7-2 Restaurant Seating
7-3 Multiples of Ten
7-4 Pizza Toppings
7-5 Movie Tickets
7-6 Three Exits
7-7 Infinity
7-8 Deli
7-9 No Pastrami
7-10 Dream Vacation

Week 5: Functions

Topics explored in this week’s presentation

- How functions work
- Passing arguments
- Returning values
- Passing a list
- Types of arguments
- Scope of variables in/outside of functions
- Creating and importing modules

What to read/watch this week

- Chapter 8 “Functions.”

Exercises to post in Canvas

8-1 Message
8-2 Favorite Book
8-3 T-Shirt
8-4 Large Shirts
8-5 Cities
8-6 City Names
8-7 Album
8-8 User Albums
8-9 Magicians
8-10 Great Magicians
8-11 Unchanged Magicians
8-12 Sandwiches
8-13 User Profile
8-14 Cars
8-15 Printing Models
8-16 Imports
8-17 Styling Functions

Week 6: Classes and object-oriented programming

Topics explored in this week's presentation

- Review: object-oriented programming
- Creating and using classes
- Instances of classes
- Inheritance
- Importing classes
- The Python library

What to read/watch this week

- Chapter 9, “Classes”

Exercises to post in Canvas

9-1 Restaurant
9-2 Three Restaurants
9-3 Users

9-4 Number Served
9-5 Login Attempts
9-6 Ice Cream Stand
9-7 Admin
9-8 Privileges
9-9 Battery Upgrade
9-10 Imported Restaurant
9-11 Imported Admin
9-12 Multiple Modules
9-13 OrderdDict Rewrite
9-14 Dice
9-15 Python Module of the Week

Week 7 The Alien Invasion begins

Topics explored in this week's presentation

- Overview and game planning
- Installing Pygame
- Video demos

What to read/watch this week

- Chapter 12, “A Ship That Fires Bullets.”

Progress to post in Canvas

All game code covered in Chapter 12

Week 8 Aliens!

Topics explored in this week's presentation

- Review of project
- Video demos

What to read/watch this week

- Chapter 13, “Aliens!”

Progress to post in Canvas

All game code covered in Chapter 13

Week 9 Scoring

Topics explored in this week's presentation

- Review of project
- video demos

What to read/watch this week

- Chapter 14, “Scoring.”

Progress to post in Canvas

All game code covered in Chapter 14

Week 10: Sharing your work on Github

What to read/watch this week

- Appendix D, "Using Git for Version Control"
- (Optional) Chapters 10 and 11, "Files and Exceptions, and "Testing Your Code."

Topics explored in this week's presentation

- Github tutorial
- Resources for learning more

Work to post

- Create a Github account and repo; post your final game code and files to your repo and share the link on our Slack channel.

ASSIGNMENTS

Weekly exercises: These are incremental applications of the programming concepts and syntax covered in each chapter. If you get stuck on any of them you can ask for help on our Slack channel or from another student while working together, but only after following the "twenty minute rule, which means you must troubleshoot code for at least 20 minutes using the following guidelines:

- Look for simple typos or syntax errors. It can help to run your finger along the code or use a piece of blank paper to isolate each line of code.
- Review the textbook carefully, since each exercise builds on examples already given.
- Search Python.org documentation for Python 3.5
- Do a Google search (but be aware that Stack Overflow will often give you way too much information)

Once you've tried these strategies for 20 minutes you can post your exercise on Slack and describe the error/wrong results you're getting and a summary of the solutions you've already tried. If you're coding with another student(s) please follow the same process before asking for help. When giving help, try to give hints rather than simply solving the problem (unless it's just a typo). You'll remember far more if you can solve a problem with hints rather than just getting the solution or copying someone else's code.

Final Project: This will be the Space Aliens game covered in chapters 12-14. Project code is provided, but you will learn a huge amount from creating the project yourself. If you want to modify the game you're free to be creative, but your modifications must add variation or complexity rather than simplifying the game.

Final Reflection: This is a two-page summary of your final project coding experience and a comparison of the game project with the summaries provided of the other three projects in Module 10. The goal is to put your project work into a broader perspective based on the way the other projects are organized and the kinds of logic they use.

Slack participation: Each week I'll post a "big picture" question related to the concepts covered that week. You'll need to answer the question and are also encouraged to ask your own questions.

Video Conferences: These are brief check-ins that will help me see where you are in your understanding. You'll be expected to be able to walk me through your code exercises and describe how they work.

GRADING BREAKDOWN

- Weekly exercises = 40%
- Final Project = 45%
- Final Reflection = 5%
- Slack participation = 5%
- Video conferences = 5%

Grading scale

A	100–94	B–	82–80	D+	69–67
A–	93–90	C+	79–77	D	66–63
B+	89–87	C	76–73	D–	62–60
B	86–83	C–	72–70	F	59 or less

Library and Learning Commons
http://www.seattleu.edu/learningcommons/

Academic integrity tutorial
https://www.seattleu.edu/academicintegrity/

Support for students with disabilities
If you have, or think you may have, a disability (including an "invisible disability" such as a learning disability, a chronic health problem, or a mental health condition) that interferes with your performance as a student in this class, you are encouraged to arrange support services and/or accommodations through Disabilities Services staff located in Loyola 100, (206) 296-5740. Disability-based adjustments to course expectations can be arranged only through this process.

Late work
You should post your weekly exercises on time even if you are having problems getting your code to work. You can continue to troubleshoot the code and re-submit corrected work. If your work is posted late more than occasionally it will affect your grade.

Academic policies on the Registrar website
https://www.seattleu.edu/registrar/academics/performance/
Be sure that you understand the university policies below, posted on the Registrar's website:
Academic integrity policy
Academic Grading Grievance Policy