Well-Trained Mind Academy

Introduction to Computer Programming

Course Blackboard site: wtma.blackboard.com

Required Online Text (no need to purchase):

- Wentworth, Pete; Elkner, Jeffrey; Downey, Allen B.; Meyers, Chris, How to Think Like a Computer Scientist: Learn with Python 3 Documentation (Release 3rd Edition), March 17,2020

Course Description: Introduction to Computer Programming teaches the foundations of basic computer programming and will introduce students to object-oriented programming. The goal of this course is to help students develop their problem-solving and critical thinking skills. This course is for beginners with little to no experience in computer programming. Students will learn computer commands, variables, looping, If/else statements, While and For loops, and much more. This course will use Python, a programming language. You will need to download Python for the course: https://www.python.org/downloads/.

Note: Students must be able to download Python for the course.

Is my child ready for Introduction to Computer Programming?

This course is designed for students that have no experience in programming. However, algebraic concepts will be covered in some of the programs, so students need to have successfully completed Pre-Algebra in order to take the course. Students who have successfully completed Pre-Algebra may take the course.

Requirements and Prerequisites:

- PC with Windows 8+ Operating System (Python 9.5 can not be downloaded on Windows 7 or earlier.)
- Download the Windows embeddable package (64-bit) found here: https://www.python.org/downloads/windows/
- Have permissions to read, write, and edit files
- Have a plain text editor (ex: Notepad)
- Basic computer literacy
    - able to find file folders and locations
    - save, delete, edit files
    - download files
    - use file manager to find the location of files
- Basic mathematical knowledge (Student should have passed Pre-Algebra.)
    - Area of rectangles, triangles, and circles
        - Solve mathematical equations
        - Understand and compute exponents
        - Solve order of operation problems

- Evaluate mathematical equations given numbers to input (including the distance formula)

Course Assignments: The course assignments are based on programs from the end of the chapters and are assigned weekly.

Grades will be based on:

Homework (Designing Computer Programs) – 50%

Homework is assigned weekly. At the end of each course section, students will work through computer programs using the skills learned in that section. Students will demonstrate their understanding of the lessons through chapter quizzes.

Chapter quizzes – 20%

Midterm/Final Programs – 20%

Participation (Weekly Exit Tickets) - 10%

An exit ticket asks students to explain or use something covered in the week's lesson ensure their understanding of the concepts taught.

Example Schedule

Week	Text Sections/Topics (subject to change)
	ORIENTATION
1	Chapter 1 (1.1 – 1.12) The Way of the Program 1.12 Exercises 1 - 6
2	**Ch 1 Quiz** Chapter 2 (2.1 – 2.3) Variables 2.14 Exercises 1 - 4
3	Ch 2 (2.4 – 2.9) Variables cont'd 2.14 Exercises 5 - 6
4	Ch 2 (2.5 – 2.12) Variables cont'd 2.14 Exercises 7 - 8

5	**Ch 2 Quiz** Ch 3.11 – 3.16 Program Flow (Hello, little turtles!) 3.4.4 Exercises 1 - 2
6	3.2 Program Flow cont’d (Conditionals) 3.4.4 Exercises 3 - 5
7	3.3 Program Flow cont’d (Iteration) 3.4.4 Exercises 6 - 8
8	3.3 Program Flow cont’d (Iteration) 3.4.4 Exercises 9 - 11
9	3.4 Some Tips, Tricks, and Common Errors 3.4.4 Exercises 12- 14
10	**Ch 3 Quiz** 4.1 Functions 4.9 Exercises (1 – 2)
11	4.2 – 4.4 (Functions can call other functions) 4.9 Exercises (3 – 6)
12	4.5 – 4.10 (Flow of execution) 4.9 Exercises (7 – 10)
13	4.11 – 4.15 4.17 Exercises (1 – 9)
14	4.16 – 4.19 4.17 Exercises (10 – 15)
15	**Thanksgiving Break**

16	4.20 – 4.22 4.17 Exercises (16 - 19)
17	**Ch 4 Quiz** Semester Review
18	**(MIDTERM PROGRAM) END OF 2nd Quarter**
Break	**NO CLASSES, WINTER BREAK**
19	5.1 Strings 5.1.19 Exercises (1 – 4)
20	5.1 Strings cont'd 5.1.19 Exercises (5 - 9)
21	5.1 Strings cont'd 5.1.19 Exercises (10 – 13)
22	5.2 – 5.3 (Tuples & Lists) 5.26 Exercises (1-3)
23	5.3 (Lists cont'd) 5.3.23 Exercises (1 – 10)
24	5.4 Dictionaries 5.46 Exercises (1 – 4)
25	**Ch 5 Quiz** Ch 6 Numpy (6.1 – 6.8) 6.8 Exercises (1 – 2)
26	**Ch 6 Quiz** Ch 7 Files 7.10 Exercises (1 – 5)
Break	**No Classes - Spring Break**

27	**Ch 7 Quiz** 8.1 – 8.4 Modules 8.10 Exercises (1 – 3)
28	8.5 – 8.8 Modules continued 8.10 Exercises (4 – 10)
29	**Ch 8 Quiz** 9.1 More datatypes Review Ch 8 exercises
30	**Ch 9 Quiz** Ch 10 Recursion (10. 1 Drawing Fractals) 10.9 Exercises (1 – 3)
31	10.2 – 10.3 (Recursive data structures & Processing recursive number lists) 10.9 Exercises (4 – 6)
32	10.4 – 10.5 (Case study: Fibonacci numbers & Example with recursive directories and files) 10.9 Exercises (7 – 8)
33	10.6 – 10.7 (An animated fractal, using PyGame and Mutual Recursion) 10.9 Exercises (9 – 11)
34	**Ch 10 Quiz**/ Review
35	**No Classes - Exam Week (FINAL PROGRAM)**