INST326: Object-Oriented Programming
Course Syllabus

Fall 2020 – Section 102 – Online – Asynchronous

---

1 Instructor

Name:	Joshua A. Westgard, PhD
Email:	westgard@umd.edu
Phone:	301-405-9136 (office)
Office:	B0225 McKeldin Library (but not regularly on campus in Fall 2020)
Office Hours:	MW 1-2 pm, and by appointment

2 Catalog Description

This course is an introduction to programming, emphasizing understanding and implementation of applications using object-oriented techniques. Topics to be covered include program design and testing as well as implementation of programs. Prerequisite: Minimum grade of C- in INST126.

3 Extended Course Description

This course covers (1) the core features of the Python programming language, (2) using programs to collect, process, and analyze data, and (3) object-oriented programming. Object-oriented programs are built as collections of "objects", which are software representations of real-world entities and concepts. Objects combine data (attributes) with functionality (methods), and work through communicating with each other as the code is executed. By encapsulating code complexity within objects, OOP allows use and reuse of existing code in a relatively simple and easy manner. Advanced OOP concepts such as inheritance facilitate development of complex code without sacrificing robustness and possibility of code reuse. We apply computational thinking approaches such as abstraction, decomposition, algorithmic design, generalization, evaluation, and debugging.

This course also provides opportunities to develop an understanding of how programming is situated in and reflects broader social structures, constructs and issues, e.g. race, class or gender. Programming is often viewed as a value-neutral technical skill. However, the social and cultural impacts of information and technology are central concepts in our field, and the growing awareness of issues like algorithmic bias, ethical/unethical uses of algorithms and disparities in opportunities in tech jobs require that any informed professional needs to understand the larger context of programming. This is important to be ethical professionals and to be successful in the workplace. Through readings, discussion and writing, we will critically examine issues of racism, sexism and other forms of power and oppression that are pervasive in programming and related technical activities, and discuss what companies and individuals are doing to improve programming practices and professional work environments.

4 Student Learning Outcomes

After finishing this course, students will be able to:

1. Design, program, and debug Python applications to solve non-trivial problems;
2. Write scripts to collect, process, and/or analyze data;
3. Explain OOP concepts, principles, design patterns and methods;
4. Test and assess code quality;
5. Write clear and effective documentation;
6. Explain how programming is situated in and reflects social issues (e.g. racism, classism, ableism, or sexism) and describe actions that individuals or organizations are taking to counteract disparities and inequities.

5 Teaching Notes

This course assumes a basic understanding of procedural programming, and begins with a comprehensive review of Python fundamentals—including data types, variables, loops, and conditionals—that is designed to deepen your mastery of these concepts. The first part of the course will thus be an opportunity to consolidate and extend what was covered in INST126.

This is an asynchronous course, divided into weekly modules. Each module consists of a mixture of slides, recordings, readings in an online textbook, comprehension quizzes, and challenge exercises. These module activities can be worked through on your own schedule, but must be completed by 11:59 pm on the Sunday after the week of the module.

In addition to the module tasks and exercises, there will be five homework assignments in which you will help you apply, reflect and extend your understanding by working on a practical task. All homework assignments are to be completed on your own unless otherwise stated in the instructions.

We will also examine selected broader issues of programming and coding—-the social and organizational context, issues related to gender, race, disability, etc. This will help you prepare for situations that you are likely to encounter in your professional work. These are noted in the schedule as "Critical Reflections."

Finally, at the end of the term everyone will complete a final project that will be an opportunity for you to synthesize what you have learned over the semester by creating a useful Python application with real-world data.

Here is my suggested general strategy for working on programming assignments:

1. Start early—don't wait. That will give you time to work through the problem and get help as needed.
2. Plan out your solution, but follow the incremental coding procedure as you implement it. Incremental coding means getting a version that does some small part of the whole task, and then adding to it step by step, testing each time you add something. This approach makes it much easier to locate bugs, and you'll be able to see the data changing as you go.
3. Read error messages carefully and try to understand what they are telling you. Often they will point you directly to the cause of the error.
4. If the solution is not immediately obvious, spend an additional 5-10 minutes trying to solve it on your own, but then take a break. Sometimes this will allow you to come back and see something you missed.
5. If you've spent 20-30 minutes and still are stuck—ask for help! You can contact me via email or discussion board. Please provide as much information as you can. Often it helps to include a screenshot with the problem. I will respond as soon as I am able, usually within a day.

6. If you see a question on the discussion board that you can answer, or if you have an idea, please respond. Don't wait for me. You will be helping your colleagues.

6 Textbooks & Readings

The required book for this course is online through the zyBooks.com platform. To register:

- Go to learn.zybooks.com
- Enter zyBook code: UMDINST326WestgardFall2020
- Subscribe (cost is $58.00)

Other readings (generally available online, or through Library subscriptions) may be assigned as needed.

7 Required Technology

- Computer: You will need a computer (Windows, Mac OS, or Linux) on which you are able to install software.
- Python: The Python interpreter (version 3), freely available from the Python Foundation.
- Editor: An advanced text editor (such as Sublime Text or BBEdit) or an integrated development environment (such as VSCode or Atom). Installation instructions will be provided.

  *Please note that we will install all necessary environments together in class during the first week.

8 Grading

Your final grade for the course is computed as the sum of your scores on the individual elements below (100 possible points total), converted to a letter grade:

A+	97-100	B+	87-89.99	C+	77-79.99	D+	67-69.99		
A	93-96.99	B	83-86.99	C	73-76.99	D	63-66.99	F	0-59.99
A-	90-92.99	B-	80-82.99	C-	70-72.99	D-	60-62.99		

Final grades will be calculated based on the following components:

Module Exercises/Quizzes (14 x 3 pts. each, dropping two points)	40 points
Homework/Labs (5)	25 points
Reflections (3)	15 points
Final Project	20 points
TOTAL	100 points

9 University Course Policies

The essential purpose of the university's undergraduate course policies is to enable all of us to fully participate in an equitable, accessible and safe academic environment so that we each can be challenged to learn and contribute most effectively. They address issues such as academic integrity, codes of conduct, discrimination, accessibility, learning accommodations, etc. We are all responsible for following the policies at http://www.ugst.umd.edu/courserelatedpolicies.html. You must read them and send me any questions by the first week of classes.

10 Late Work

Submission instructions are provided with each assignment, but as a general rule the on-time submission window will close in ELMS at 11:59 pm on the date due. If you have to miss a deadline, you should inform me as soon as possible, indicating the reason and when you propose to submit your work. Due to the pandemic, excuse documentation will consist of a self-certification form.

11 Syllabus Revision Policy

This syllabus is a guide for the course and is subject to change with advance notice. Changes will be posted in ELMS. The ELMS course site together with the course OER website, are the definitive locations for all course materials, communication, assignments, and deadlines.

12 Course Schedule

The following table shows the current projected schedule. The course content can be roughly divided into three interrelated units:

- Unit 1: Procedural Programming Review Using Python (∼weeks 1-3)
- Unit 2: Object-Oriented Programming Using Python (∼weeks 4-9)
- Unit 3: Data Analysis Using Python (∼weeks 10-14)

Week	Monday	Wednesday	Friday
1	08/31 Module 1: Fundamentals 1	09/02 Module 1: Fundamentals 1	09/04 Module 1: Fundamentals 1
2	09/07 LABOR DAY	09/09 Module 2: Fundamentals 2	09/11 Module 2: Fundamentals 2
3	09/14 Module 3: Git & Testing	09/16 Module 3: Git & Testing	09/18 Module 3: Git & Testing **HW1**
4	09/21 Module 4: Basics of OOP	09/23 Module 4: Basics of OOP	09/25 Module 4: Basics of OOP
5	09/28 Module 5: Adv. Data Structures	09/30 Module 5: Adv. Data Structures	10/02 Module 5: Adv. Data Structures **HW2**
6	10/05 Module 6: Serialization & File I/O	10/07 Module 6: Serialization & File I/O	10/09 Module 6: Serialization & File I/O **CR1**
7	10/12 Module 7: Regular Expressions	10/14 Module 7: Regular Expressions	10/16 Module 7: Regular Expressions **HW3**
8	10/19 Module 8: Remote Collaboration	10/21 Module 8: Remote Collaboration	10/23 Module 8: Remote Collaboration
9	10/26 Module 9: Advanced OOP	10/28 Module 9: Advanced OOP	10/30 Module 9: Advanced OOP **HW4**
10	11/02 Module 10: Databases & SQL	11/04 Module 10: Databases & SQL	11/06 Module 10: Databases & SQL **CR2**
11	11/09 Module 11: Data on the Web	11/11 Module 11: Data on the Web	11/13 Module 11: Data on the Web
12	11/16 Module 12: Web Scraping	11/18 Module 12: Web Scraping	11/20 Module 12: Web Scraping **HW5**
13	11/23 Final Project Planning Day	11/25 THANKSGIVING	11/27 THANKSGIVING
14	11/30 Module 13: Data Analysis	12/02 Module 13: Data Analysis	12/04 Module 13: Data Analysis **CR3**
15	12/07 Module 14: Bringing It All Together	12/09 Module 14: Bringing It All Together	12/11 Module 14: Bringing It All Together
16	12/14 Course Wrap-Up		**Projects Due 11:59pm on 12/17 (Thursday)**