Course Syllabus

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.

Prerequisites: must have completed or be concurrently enrolled in (INST201; or INST301); and (INST126; or CMSC106; or CMSC122), or permission of instructor. Credit only granted for: INST326 or CMSC131.

Extended Course Description

This course introduces object-oriented design and programming concepts and methods using the Python programming language. 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.

Student Learning Outcomes

After finishing this course, students will be able to:

- Explain OOP concepts, principles, design patterns and methods;
- Design, program and debug Python applications to solve non-trivial problems;
- Test and assess the quality of object-oriented code;

- Write clear and effective documentation;
- Explain how programming is situated in and reflects social issues (e.g. racism, classism or

  sexism) and describe actions that individuals or organizations are taking to counteract disparities and inequities.

Teaching Notes

This course builds on a basic understanding of procedural programming, so you have to understand data types, variables, loops, conditionals, etc. and how to use them to write and debug a program. If you are fluent in a language such as JavaScript, Java, C#, Visual Basic, etc. you can readily apply your knowledge to learn Python. If you know a bit of Python already, you might find the first part of the course a bit of review.

- Final grades will be assigned based on your weighted average in the class using the following categories:

A+	More than 97.0*
A	93.0 - 96.9
A-	90.0 - 92.9
B+	87.0 - 89.9
B	83.0 - 86.9
B-	80.0 - 82.9
C+	77.0 - 79.9
C	73.0 - 76.9
C-	70.0 - 72.9
D+	67.0 - 69.9
D	63.0 - 66.9
D-	60.0 - 62.9
F	Less than 60

* Note: To receive an A+ you must have demonstrated significant contributions to the class in addition to achieving this numeric grade.

Homework Assignments

Weekly assignments where students work on practical problems will be a major part of the course. The goal is to get you doing a lot of coding practice because it is the best way to learn.

Detailed Schedule

- See Canvas for a detailed weekly schedule.

Extensions

If you have to miss a deadline, you should inform the instructor as soon as possible, indicating when you will submit your work. The instructor will try to accommodate your needs. You should use this clause only for extraordinary personal reasons (e.g., personal illness, death in the family, etc.), not because you are busy with other classes, have a career fair to attend, etc. The general policy is that late work will be deducted 20% of its total grade per calendar day, starting on the same day it is due. It is at the instructor's discretion to accept late work and assign late penalties.

There are absolutely no extensions or re-dos on quizzes.

University Policies

For university course policies, review go.umd.edu/ug-policy (http://go.umd.edu/ug-policy)

Course Summary:

Date	Details	
Mon Feb 4, 2019	Homework 1: Review with dice (https://myelms.umd.edu/courses/1261386/assignments/4803531)	due by 11:59pm
Mon Feb 11, 2019	Homework 2: Condition Review (https://myelms.umd.edu/courses/1261386/assignments/4834463)	due by 11:59pm
Mon Feb 18, 2019	Week 3 Quiz (https://myelms.umd.edu/courses/1261386/assignments/4842150)	due by 11:59pm
Mon Feb 25, 2019	Homework 3: Ship, Captain, Crew (https://myelms.umd.edu/courses/1261386/assignments/4842151)	due by 11:59pm
	Homework 4: Data structures (https://myelms.umd.edu/courses/1261386/assignments/4842744)	due by 11:59pm

Date	Details
	Week 2 Quiz (https://myelms.umd.edu/courses/1261386/assignments/4835071)
	Week 4 Quiz (https://myelms.umd.edu/courses/1261386/assignments/4842742)