COLLEGE OF INFORMATION STUDIES
MARYLAND'S iSCHOOL

I. Course Information

Course:	**INST 326 – Object Oriented Programming**
Term:	**Fall 2017**
Instructor:	**Dr. Timothy M. Richards**
Class Time & Location:	**Tues & Thurs 12:30-1:45PM / PHY 4221**
Office/Office Hours:	**By appointment**
E-Mail:	**To be announced**

Required Textbook/Materials:
Python 3 Object-oriented Programming, 2nd ed, ($35 print, free online), by Dusty Philips
Paperback: 460 pages
Publisher: Packt Publishing - ebooks Account; 2 edition (August 20, 2015)
Language: English
ISBN-10: 1784398780
ISBN-13: 978-1784398781

Python 3 Object-oriented Programming, 2nd ed, ($35 print, free online), by Dusty Philips
Paperback: 460 pages
Publisher: Packt Publishing - ebooks Account; 2 edition (August 20, 2015)
Language: English
ISBN-10: 1784398780
ISBN-13: 978-1784398781

Required Technology
Laptop - We will do live programming exercises during most classes, so bring your laptop and be prepared to write code. Any reasonably current operating system can be used. If you don't have access to a laptop, contact me before the first class.

Python - Python programming language (version 3) platform that supports object-oriented programming. The programming platform is freely available from https://www.python.org/downloads/.

Editor - An advanced text editor (such as Sublime Text, Notepad++, Crimson Editor) and/or an integrated development environment (such as NetBeans, Eclipse, PyCharm, Geany).

II. Course Details

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: (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 any informed professional needs to understand how these issues manifest in a variety of circumstances. Through readings, discussion and writing, we will critically examine issues of racism, sexism and other forms of oppression that are pervasive in programming and related technical activities.

Student Learning Outcomes

After finishing this course, students will be able to:

1. Explain OOP concepts, principles, design patterns and methods;
2. Design, program and debug Python applications to solve non-trivial problems;
3. Test and assess the quality of object-oriented code;
4. Write clear and effective documentation
5. Explain how programming is situated in and reflects broader social structures, constructs and issues, e.g. race, class or gender.

Teaching Notes

Each week will typically follow this pattern, with some exceptions:

Before class (preparation):

- Do assigned readings; watch assigned videos;
- Do pre-class activities – these help you confirm that you understand the basic material or help you identify specific aspects that you have questions about;
- Complete a low-stakes assignment or quiz the night before class. I review these before class and use them to prepare for class.

In class:

- We will use a mix of lecture, discussion and lots of hands-on activities to help you apply the materials;
- We will make extensive use of paired and group work in class;
- Class is not a time for solo learning. As members of a learning community, we are mutually responsible to each other as learners. Each of us has to be fully engaged with each other in the activities. We have to be supportive of each other as we try to explain or demonstrate something new, as we inevitably make mistakes. We aren't successful unless everyone is learning.

After class (homework):

- There will be weekly follow-up activities as homework to help you practice, reflect and extend your understandings. You usually have the option of either working together or solo on these, as long as the final work product is your own.

During the second half of the semester, after you have had time to focus on the essential elements of Python, we will 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 perspectives."

Our time together in class is precious. To use it effectively, you must come to class on time and prepared. Being prepared for class means that you have:

a. Completed all the readings/videos;
b. Attempted all the pre-class activities;
c. Either successfully completed them or submitted your questions the night before class, so I have time to prepare and answer them in class.
d. Arrived 5 minutes before class starts; are in your seat, with your computer running. You have downloaded any notes or materials for the day from ELMS. Any paper assignments are ready to hand in. You are ready to take notes.

III. Evaluation

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	F 0-59.99
A 93-96.99	B 83-86.99	C 73-76.99	D 63-66.99	
A- 90-92.99	B- 80-82.99	C- 70-72.99	D- 60-62.99	

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

Graded item	**Percent of final grade**
Preparation – Low stakes assessment or quiz before class.	10
In class participation – Reflects your active and visible engagement with the in-class activities.	10
Homework – Will be assigned weekly, with some exceptions. Includes coding problems (of course), but will also include analysis questions, brief reflective writing, and other activities.	30
Quizes (5) - In class (10-15 min) paper-based.	10

Midterms – There will be two in-class, paper-based mid-term exams. These are diagnostics for you to assess your understanding of the programming basics that are necessary as you proceed through the course and prepare for term project. You will want to address any weaknesses these diagnostics identify to ensure you are well prepared for the project.	20
Project – The term project will give you an opportunity to apply and extend what you learn in class.	15
Critical reflection - At the end of the semester you will submit a critical reflection that builds on our critical readings. It will also address the course, your learning, and your plans for professional growth in this area.	5

IV. Course Policies and Expectations

1. Regular punctual attendance is expected of all students. Students are expected to remain for the entire class period. Students are responsible for all announcements, material covered, and assignments due when absent from class. The instructor recommends exchanging contact information with other students to share lecture notes. Tardiness and repeated class interruptions may reduce the student's participation grade.

2. **Students are expected to read the all chapter assignments before coming to class and be prepared to discuss the topics and participate in class/group activities and exercises in class.**

3. **Late project submissions are not accepted**. Projects not submitted by the deadline will receive an automatic grade of zero.

4. Each assignment must be submitted as requested in the instructions. *Assignments submitted via email will not be graded*. Assignments not submitted as required by the instructions will not be graded.

5. Students are expected to put away all electronic devices during lectures. The use of mobile devices (i.e. phones, tablets, etc) during the lecture is disruptive and disrespectful. Texting, using email, playing games, chatting and browsing the web is not permitted during the lecture session unless doing so is a part of the class session's planned activities and students are instructed to do so by the instructor. Failing to follow this expectation may result in a reduced participation grade.

6. This class frequently requires group work, in-class exercises, and in-class research so DO bring a laptop to class for use during designated times.

7. Exams must be taken as scheduled. If you are unable to take your exam at the scheduled time due to an emergency (hospitalization, car accident, etc.) contact the instructor prior to the exam time to make arrangements to take the exam. Documentation will be required**. Make-up exams will only be given in the event of an extreme emergency and at the sole discretion of the instructor.**

8. The instructor will reply to student emails within 72 hours Monday through Friday. Emails received on university holidays or during the weekend will receive a response when the university reopens.

9. Students shall use APA formatting for the all written assignments in this course.

10. In the event of university closure due to inclement weather or other unanticipated events, due dates as posted on the course schedule will remain in effect unless a change is made by the instructor. Such a change will be posted in ELMS as an announcement and/or sent via email.

11. Participation Policy: Participation grades involve engaging effectively with in-class exercises, participating in group work, interacting with your instructor and peers, and attending class regularly.

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.

Syllabus Change 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 is the definitive location for all course work, and communication, including class schedules, assignments and deadlines.

V. Course Schedule

Please see the schedule sheet or ELMS.

The course schedule is tentative and subject to change based on the needs of the class and by the sole discretion of the instructor.