

Misericordia University  
Mathematical Sciences Department

**Course Information**

<u>Course Number:</u>	<b>MIS 310 01</b>
<u>Course Title:</u>	<b>Managerial Applications of Object-Oriented Programming I</b>
<u>Course Credits:</u>	3
<u>Semester:</u>	2023 Fall
<u>Instructor:</u>	Fanchao Meng
- <u>Contact:</u>	fmeng@misericordia.edu
- <u>Office Location:</u>	McAuley Hall 310
- <u>Office Hours:</u>	<b>Mon. &amp; Wed. 1:00 ~ 3:00 PM EST, or by appointments</b>
<u>Recommended Texts/Readings:</u>	<ul style="list-style-type: none"><li>- Intro to Python for the Computer &amp; Data Sciences, Paul Deitel and Harvey Deitel.</li><li>- Introduction to Computation and Programming Using Python, 3rd Edition, John V. Guttag.</li><li>- Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib, 2nd Edition, Robert Johanson.</li><li>- Python Programming and Numerical Methods: A Guide for Engineers and Scientists, Qingkai Kong, Timmy Siau, Alexandre Bayen.</li></ul>
<u>Class Meeting Time:</u>	<b>Tue. &amp; Thu. 9:30 AM ~ 10:45 PM EST</b> (When the compressed schedule applies: <b>Tue. &amp; Thu. 11:15 AM ~ 12:15 PM EST</b> )
<u>Classroom:</u>	<b>Henry Science Center 265</b>

## Course Description

This course provides a study of an object-oriented programming language as it pertains to managerial applications. In addition, the course introduces the use of object-oriented programming methodologies.

## Course Objectives

Course Objective	After completion of this course, the student will:	Assessment Method
1. Understand environment settings for Python.	<ul style="list-style-type: none"><li>- Understand fundamental package management;</li><li>- Set up virtual environments (e.g. using Anaconda);</li><li>- Be able to use both IDEs and command-line to run Python programs inside virtual environments.</li></ul>	Assignments (both written and programming), quizzes and exams.
2. Understand fundamental Python programming.	<ul style="list-style-type: none"><li>- Understand Python grammar;</li><li>- Be able to use fundamental Python built-in data structures (e.g. lists, tuples, dictionaries and sets);</li><li>- Be able to use fundamental Python built-in packages.</li></ul>	
3. Understand Object-Oriented designs using UML.	<ul style="list-style-type: none"><li>- Understand use case and class designs;</li><li>- Be able to use UML to draw use case class diagrams.</li></ul>	
4. Understand Object-Oriented Programming (OOP)	<ul style="list-style-type: none"><li>- Understand encapsulation, abstraction, inheritance and polymorphism;</li><li>- Understand object serialization;</li><li>- Be able to use OOP in Python.</li></ul>	
5. Use cases in real problems.	<ul style="list-style-type: none"><li>- Understand how OOP is applied in solving real problems (e.g. hardware abstract layers and data transmission).</li></ul>	

## Course Delivery

### Teaching and Learning Strategies:

- In-person classroom sessions
- In-class quizzes

- Written assignments
- Programming assignments
- Exams

Grading Method:

Item	Amount
Assignments	40%
Quizzes	10%
Midterm 1	15%
Midterm 2	15%
Final	20%
Extra Credits	5%
<b>TOTAL</b>	100% ~ 105%

Grading System:

Grade	Range
A	$\geq 90$
A-	(90, 85]
B+	(85, 80]
B	(80, 75]
B-	(75, 70]
C+	(70, 65]
C	(65, 60]
C-	(60, 55]
D	(55, 50]
F	$< 50$

## Course and Academic Policies

Academic Policies:

All Misericordia University courses follow standard academic policies, described here: <https://catalog.misericordia.edu/content.php?catoid=8&navoid=490>, or on the portal under the "Students" tab.

Tutoring:

Information about tutoring and other services available through the Student Success Center are described here: <https://catalog.misericordia.edu/content.php?catoid=8&navoid=474#student-success-center>, or on the portal under the "Students" tab.

Emergency Alerts:

Both faculty members and students keep cell phone on in event of an MU alert.

Attendance Policy:

Attendance is required, but is not reckoned in grading. Any absence is required to be reported to the instructor.

## Tentative Schedules

Tentative Assignment & Exam Schedule:

Week	Assignments/Exams
3	Assignment 1 Out
5	Assignment 1 Due, Assignment 2 Out
6	Midterm 1
7	Assignment 2 Due, Assignment 3 Out
9	Assignment 3 Due, Assignment 4 Out
11	Midterm 2, Assignment 5 Out
11	Assignment 4 Due
14	Assignment 5 Due
16	Final

Tentative Topics:

- Virtual environments and package managements
- Python grammar
- Fundamental Python built-in data structures (e.g. lists, tuples, dictionaries and sets)
- Fundamental Python built-in packages
- Basic UML
- Use case diagrams and class diagrams
- Object-Oriented Programming concepts and techniques (e.g. encapsulation, abstraction, inheritance and polymorphism)
- OOP in Python
- OOP use cases in real problems