Fall 2021 CDS Graduate Course

CSS 605 / ECON 895

# Object Oriented Modeling for Social Sciences

Wednesdays, 4:30pm to 7:10pm
Music/Theater Building 1004
GMU Fairfax Campus


Dr. Stephen L. Scott

sscotta@gmu.edu

College of Science
Department of Computational and Data Science
Computational Social Science Program

Office hours: by appointment

# Overview

This course presents and applies concepts and principles from object-based modeling paradigm. It emphasizes Unified Modeling Language (UML) for design concepts and object-oriented programming languages such as Python and Java to render structure and operation of complex social systems and processes.

This course is designed to help students develop the basic skills required for Agent-Based Modeling (ABM) with special emphasis on social science and economic applications. We will learn principles of Object-Oriented Analysis (OOA) and Object-Oriented Design (OOD) using the Unified Modeling Language (UML). We will cover Object-Oriented Programming chiefly using the popular OO programming languages Python and Java. The course focuses on hands-on analysis, modeling, and programming to study socio-economic systems and acquire an object-oriented way of thinking.

**Recommended Prerequisites**: There are no formal prerequisites for this course. Students are expected to have some basic familiarity with computing and mathematics at the level of college algebra. However, discussions may include references to topics from calculus, vector and matrix algebra, descriptive statistics, and will include examples of elementary computer programming. Students without these prerequisites are absolutely welcome to enroll, as all the necessary mathematics and computer science topics will be introduced as needed and should not pose an impediment to completing the assignments successfully.

**Registration Restrictions:**
Enrollment in the course is limited to students with a class of Advanced to Candidacy, Graduate, Non-Degree or Senior Plus. The course is available to Graduate, Non-Degree or Undergraduate level students, however, students in a Non-Degree Undergraduate degree may not enroll. The course is offered by the Computational and Data Sciences Department and may not be repeated for credit.

**Schedule Type**: Weekly course for Fall 2021

# Objective

This course is designed to provide students with an understanding of Object-Oriented analysis, design, and programming using UML, Python, and Java. Specific topics are as follows.

- **Unified Modeling Language (UML):** UML is the de facto standard for expressing object-oriented analysis and design artifacts. The language consists of formal graphical elements combined with structured text and is used to express the design of object-oriented systems in a standard way.
- **Python**: an object-oriented programming language widely used in industry and academia. Python offers full-featured scientific computing environment using a clean, elegant syntax that is easy to learn.
- **Java**: an object-oriented programming language widely used in industry and academia. Java was one of the first mainstream object-oriented languages and is still in active use today.

# Course Organization

The course will be presented as a series of separate learning modules. The course is very much "hands-on" and does not delve deeply into mathematical or computational theories.

# Course Grading

The course has no midterm exam, no final exam, and no "pop-quiz" exams. The course grade will be determined based cumulative scores received on weekly lab assignments and a capstone project presentation.

Worksheets will contain materials drawn from lectures, and may include computational example problems, short answer questions, and/or programming assignments. Students should expect to spend about 1-2 hours to complete each worksheet.

The Capstone project presentation will provide students with an opportunity to show an object-oriented analysis of a social science model, either drawn from an example covered in class or selected by the student from their research interests.

Topics to be addressed in weekly lab assignments include (but are not limited to) the following.

**OOA/OOD Weekly Lab Assignments**

- Understand the principles of object-oriented analysis and design
- Learn about simulation modeling for the social sciences

**UML Weekly Lab Assignments**

- Install a UML modeling tool (exact software TBD)
- Learn UML Use cases diagrams and Use Case realizations
- Learn UML Activity diagrams
- Learn UML Class diagrams

- Learn UML Sequence diagrams
- Learn UML Collaboration diagrams
- Learn UML State Diagrams

**Python Weekly Lab Assignments**

- Install the Anaconda Python IDE
- Learn Python data types and built-in data structures
- Learn to write Python functions
- Learn basic Object-Oriented features of Python
- Learn to read and write data files and CSV files
- Learn how to use Python plots for data visualization
- Learn how to use Jupyter Notebooks for documenting and sharing analysis products
- Brief introduction to the SimPy discrete event simulation system

**Java Weekly Lab Assignments**

- Install the Java Development Kit (JDK)
- Install the Eclipse Integrated Development Environment (IDE)
- Learn Java data types and built-in data structures
- Learn to write Java classes
- Learn basic Object-oriented features of Java
- Learn how to read and write data files and CSV files
- Learn how to use JFreeChart for plots and data visualization
- Brief introduction to the MASON discrete event simulation system

There are 13 lab worksheet assignments in the course, and each worksheet is weighted at 100 points. In addition, students will prepare a capstone project presentation at the end of the semester weighted at 200 points. Semester grades will be determined as follows.

| **Cumulative Score from Weekly Lab Assignments and Capstone Project Presentation** | **Semester Grade** |
|---|---|
| 1350 and above | A |
| 1200 to 1349 | B |
| 1050 to 1199 | C |
| 900 to 1049 | D |
| Below 900 | F |

**Late homework policy**

Students are expected to complete assignments in a timely manner. Worksheets should be uploaded to the BlackBoard course site by 11:59:59 pm Eastern time on the due date, typically late Wednesday evening. Late assignments will be considered on a case-by-case basis.

**Homework grading policy**

I will generally grade assignments within 24 to 48 hours of posting. Students should expect to receive feedback on their homework assignments using the BlackBoard system. Assignments will be graded on the basis of successful demonstration of proficiency in the assigned topics rather than computational elegance or run-time efficiency.

# Academic Honesty and Collaboration

The integrity of the University community is affected by the individual choices made by each of us. GMU has an Honor Code with clear guidelines regarding academic integrity. Three principles to follow at all times are that: (1) all work submitted be your own; (2) when using the work or ideas of others, including fellow students, give full credit through accurate citations; and (3) if you are uncertain about the ground rules on a particular assignment, ask for clarification. No grade is important enough to justify academic misconduct.

# Disability Statement

If you have a documented learning disability or other condition that may affect academic performance you should: 1) make sure this documentation is on file with the Office of Disability Services (SUB I, Rm. 222; 703-993-2474; http://www.gmu.edu/student/drc ) to determine the accommodations you need; and 2) talk with me to discuss your accommodation needs.

# Title IX Statement

As a faculty member, I am designated as a "Responsible Employee", and must report all disclosures of sexual assault, interpersonal violence, and stalking to Mason's Title IX Coordinator per University Policy 1202. If you wish to speak with someone confidentially, please contact one of Mason's confidential resources such as Student Support and Advocacy Center (SSAC) at 703-380-1434, or Counseling and Psychological Services (CAPS) at 703-993-2380. You may also seek assistance from Mason's Title IX Coordinator by calling 703-993-8730 or emailing titleix@gmu.edu .

## GMU Library Resources

The GMU library includes a number of helpful resources. Here are some examples and points of contact. Feel free to make use of these GMU resources if and as needed during the semester.

Technical Librarians
Margaret Lam, Physical Sciences & STEM Data Librarian
Debby Kermer, Data Services Research Consultant

They can assist with the following topics.

- Python for Data: https://infoguides.gmu.edu/learn_python
- Working with Data: https://infoguides.gmu.edu/data-work

Theresa Calcagno, IT & Engineering Librarian, can provide guidance on the following topics:

- identify and select appropriate publication/presentation venues
- prepare a publication proposal
- prepare a paper for publication/presentation
- survive the peer review process
- identify and avoid ethical issues related to publication and copyright

# Textbooks, Journal Papers, and other Resources

The course has three recommended textbooks.

Fowler, Martin. (2004). *UML distilled: a brief guide to the standard object modeling language, 3rd Edition*. Addison-Wesley Professional.

Downey, A. (2012). *Think Python*. O'Reilly Media, Inc.
Available as a free PDF download at http://greenteapress.com/wp/think-python/ . Make sure you obtain the 2nd edition for Python 3.x, not the 1st edition for Python 2.x.

Downey, A. B., & Mayfield, C. (2016). Think Java: How to Think Like a Computer Scientist. " O'Reilly Media, Inc. Available from Green-Tea Press as a free PDF download at http://www.greenteapress.com/thinkapjava/thinkapjava.pdf

In addition to the aforementioned textbooks, there are several additional recommended references listed below and students are highly encouraged to obtain these resources for the course and for future use in Computational Social Science research.

## Recommended Supplemental Books

Epstein, Joshua M. *Generative social science: Studies in agent-based computational modeling*. Princeton University Press, 2006.
Epstein, Joshua M., and Robert Axtell. *Growing artificial societies: social science from the bottom up*. Brookings Institution Press, 1996.
Miller, John H., and Scott E. Page. *Complex adaptive systems: An introduction to computational models of social life*. Vol. 17. Princeton University press, 2009.
Page, Scott E. *Diversity and complexity*. Vol. 2. Princeton University Press, 2010.
Page, Scott E. *The model thinker: What you need to know to make data work for you*. Hachette UK, 2018.

## UML Resources

**Quick Reference and Guides**

http://umich.edu/~eecs381/handouts/UMLNotationSummary.pdf
https://holub.com/uml/

## Python Resources

**Quick Reference**

https://s3.amazonaws.com/assets.datacamp.com/blog_assets/PythonForDataScience.pdf
http://www.cogsci.rpi.edu/~destem/igd/python_cheat_sheet.pdf

**Online References**

Downey, A. (2012). *Think Python*. O'Reilly Media, Inc.
Available as a free PDF download at http://greenteapress.com/wp/think-python/ . Make sure you obtain the 2nd edition for Python 3.x, not the 1st edition for Python 2.x.

http://docs.python.org contains additional documentation at the Python website.

**Trade Publications**

Lutz, M. and Asher, D. (2007) Learning Python, 3rd Edition. O'Reilly & Associates.

Zelle, J. (2016). Python Programming: Introduction to Computer Science. Franklin, Beedle and Associates Inc.

## Java Resources

**Quick Reference**

https://www.tutorialspoint.com/java/pdf/java_quick_guide.pdf
https://www.unomaha.edu/college-of-information-science-and-technology/computer-science-learning-center/_files/resources/CSLC-Helpdocs-Java1.pdf

**Online References**

Downey, A. B., & Mayfield, C. (2016). Think Java: How to Think Like a Computer Scientist. " O'Reilly Media, Inc. Available from Green-Tea Press as a free PDF download at http://www.greenteapress.com/thinkapjava/thinkapjava.pdf

The Oracle web sites https://www.java.com/en/ and https://www.oracle.com/java/ are the definitive websites for Java information and software distributions.

**Trade Books**

Deitel, P., & Deitel, H. (2011). Java How to program. Prentice Hall Press.

Eckel, B. (2003). Thinking in Java. Prentice Hall Professional. May be available online for free download, as the author has made the book open source.

# Required Software

Students are expected to download and install the following tools and packages. Version numbers are the most recent versions available and the ones we'll be using in class. Although not recommended, you may install other versions at your own discretion.

- We will be using the open-source Modelio tool for editing UML diagrams. Modelio is available as a free download for a variety of platforms (Windows, Mac, Linux). See the website at https://www.modelio.org/downloads/download-modelio.html .

- We will be using Python version 3.7x or above, which is available as a free download for a variety of platforms (Windows, MAC OS, Linux). However, students are strongly encouraged to use the Anaconda scientific distribution (use Anaconda 3 version 2021.05), which contains a large number of pre-installed scientific and research libraries. Anaconda downloads for various platforms are available at https://conda.io/docs/user-guide/install/download.html ; be sure to download the free edition as there is no need to pay for a commercial license for this course. The course will be taught using the built-in Spyder editor and Integrated Development Environment (IDE), so students using alternative Python installations or code editors will be responsible for resolving any library compatibility issues.

- We will be the Java Development Kit, version 8 or above. Note: Java is currently owned by Oracle, which has recently changed its licensing policy for the use of Java. Previously, Java was

considered open-source and freely available for use. However, Oracle has changed the policy so that Java Development Kits (JDK) and Java Runtime Environments (JREs) require an annual licensing fee of $30 USD per user per year. As an open-source alternative to Oracle's licensed product, I recommend using the Amazon Coretto 11 platform, available at: https://docs.aws.amazon.com/corretto/latest/corretto-11-ug/downloads-list.html . Select the appropriate OS and download a Java Development Kit (JDK), not a Java Runtime Environment (JRE).

- We will be using the open-source Eclipse Integrated Development Environment (IDE) for coding in Java. Eclipse offers several variants of the IDE, and I recommend using the 2019-06 version of the **Eclipse IDE for Enterprise Java Developers** available at : https://www.eclipse.org/downloads/packages/

- We will be using the open-source Java charting system JFreeChart. It's available for download at https://sourceforge.net/projects/jfreechart/ . Download this for your OS and extract the Java JAR file (don't worry if you don't know what a JAR file is - we'll cover this in class).

- We will be using the MASON (Multi Agent Simulation of Networks) discrete event simulation system, version 19. This is a Java-based discrete event simulation environment, developed and maintained by the GMU Department of Computer Science. The latest release of MASON can be downloaded at : https://cs.gmu.edu/~eclab/projects/mason/

Students are expected to provide their own computing resources to complete assignments. GMU has several computing labs available if you need access to computing platforms.

# Inclement Weather Policy

We will normally meet at our scheduled time on the GMU Fairfax campus. However, in the event of inclement weather or other extraordinary circumstances, other class options may be provided. Check the website at www.gmu.edu and local radio/TV media sources for updates, and also check the class Blackboard site for announcements and updates.

# COVID-19 Policy

We will comply with federal, state, local, and GMU guidance regarding COVID-19 public health policies. Students should wear protective masks at all times regardless of vaccination status. In the event that in-person classes are suspended due to public health conditions, we will meet virtually via Zoom and additional details will be provided as needed.

# Office Hours

Office hours are available by appointment, typically Friday afternoons between 1pm and 3pm. If you would like to schedule office hours, please send me an email at sscotta@gmu.edu by Thursday, and I will set up a Zoom room. Here are the specifics for Zoom conferencing.

**Zoom Meeting Dial In Information**

Topic: Stephen L Scott's Personal Meeting Room

**Join Zoom Meeting**

https://gmu.zoom.us/j/2881593065

Meeting ID: 288 159 3065

**One tap mobile**

+13017158592,,2881593065# US (Washington DC)

+12678310333,,2881593065# US (Philadelphia)

**Dial by your location**

+1 301 715 8592 US (Washington DC)

+1 267 831 0333 US (Philadelphia)

Meeting ID: 288 159 3065

Find your local number: https://gmu.zoom.us/u/adspAEAV3O

**Emails**

I will be checking my GMU email regularly as well, so if you have questions or comments, send email to sscotta@gmu.edu. Note that per GMU university policy and compliance with federal FERPA regulations, I can only respond to student emails from a GMU email address, i.e., an address ending in @gmu.edu.

# Course Outline

The following schedule shows the topics to be presented. Some topics may be modified during the semester. Students are encouraged to visit the class Blackboard website frequently for the latest updates and announcements. Note the due date for weekly lab assignments: these should be submitted to the online BlackBoard system by the date and time specified, using the file format(s) specified in the worksheet problem sets. Worksheets that require software source code should be formatted according to the guidelines listed at the end of this syllabus.

| Week | Class Date | Topics | Homework assignment | Due Date (11:59:59 on Wed PM) | Points |
|---|---|---|---|---|---|
| 1 | 8/25/2021 | Soc Sci modeling, OOA/OOD Principles, Modelio software installation, UML Overview | OO Worksheet 1 | 9/1/2021 | 100 |
| 2 | 9/1/2021 | Intro to UML diagrams. Advanced UML diagrams. Emblishments and adornmnents for Class Diagram, Use Case Diagram, Activity Diagram, Collaboration Diagram, Sequence diagram, State diagram. | OO Worksheet 2 | 9/8/2021 | 100 |
| 3 | 9/8/2021 | Python installation, Intro to Programming, Variables Expressions and Statements, Functions, Formatted Printing | Python Worksheet 1 | 9/15/2021 | 100 |
| 4 | 9/15/2021 | Conditionals and Recursion, Fruitful functions, Iteration, CSVs | Python Worksheet 2 | 9/22/2021 | 100 |
| 5 | 9/22/2021 | Strings, Lists, Dictionaries, Tuples, basic 2D matplotlib graphics (Python Graphics and Plot Intro) | Python Worksheet 3 | 9/29/2021 | 100 |
| 6 | 9/29/2021 | Classes and objects, Classes and functions, Classes and methods, Inheritance, Special topics (i.e. "goodies") | Python Worksheet 4 | 10/6/2021 | 100 |

| | | | | | |
|---|---|---|---|---|---|
| 7 | 10/6/2021 | Advanced Matplotlib plots, Pandas. Microeconomics model of simple supply demand market. | Python Worksheet 5 | 10/13/2021 | 100 |
| 8 | 10/13/2021 | Case study: OO modeling to study the Diffusion of Innovation. OO Modeling for Population dynamics. | Python Worksheet 6 | 10/20/2021 | 100 |
| 9 | 10/20/2021 | Java Installation (JDK, JRE, Eclipse IDE), Intro to Programming, Variables and Operators, Input and Output, Void methods, Conditionals and Logic | Java Worksheet 1 | 10/27/2021 | 100 |
| 10 | 10/27/2021 | Value Methods, Loops, Arrays, Strings and Things, Objects, Classes | Java Worksheet 2 | 11/3/2021 | 100 |
| 11 | 11/3/2021 | Arrays of Objects, Objects of Arrays, Objects of Objects | Java Worksheet 3 | 11/10/2021 | 100 |
| 12 | 11/10/2021 | Jfreechart plots, Java2D Graphics | Java Worksheet 4 | 11/17/2021 | 100 |
| 13 | 11/17/2021 | Overview of MASON architecture, examples of MASON sims | Java Worksheet 5 | 11/24/2021 | 100 |
| 14 | 11/24/2021 | Thanksgiving Break | NA | 12/1/2021 | 0 |
| 15 | 12/1/2021 | Student Presentations | Presentations and Models | 12/8/2021 | 100 |
| 16 | 12/8/2021 | Student Presentations | Presentations and Models | 12/15/2021 | 100 |

# Source Code Formatting Conventions

For assignments involving creation of technical diagrams or computer source code, the following conventions are expected.

**UML conventions**

UML diagrams should include your name, your class (CSS 605/ECON 895), the assignment number, and the date.

**Python conventions**

Python source code should include the following comment block at the beginning of each source code file.

```
# filename: file_name.py
#
# description: what is this code supposed to do
#
# your name
# your class: CSS 605 / ECON 895
# date
# assignment number
#
# anything else important for me know when grading your assignment
#
# any use of 3rd party code or libraries must be properly cited
```

Python classes and functions (which we'll cover in lectures) should be documented as such.

```
#
# function name
# function description (what is this function supposed to do)
#
def myfunc():
      code here
      code here

#
# class name
# class description (what is this class supposed to do)
#
class myclass():
      code here
      code here
```

**Java conventions**

Java source code should be indented using standard indentation conventions. The Eclipse IDE has a built-in code formatter; use this to format your code properly.

Java source code should include a comment block at the beginning of each source code file. You may use single line comments (//) or multi-line comment blocks ( /* …. */ ).

```
// filename: file_name.java
//
// description: what is this code supposed to do
//
// your name
// your class: CSS 605 / ECON 895
// date
// assignment number
//
// anything else important for me know when grading your assignment
//
// any use of 3rd party code or libraries must be properly cited
```

or

```
/*
 filename: file_name.java

 description: what is this code supposed to do

 your name
 your class: CSS 605 / ECON 895
 date
 assignment number

 anything else important for me know when grading your assignment

 any use of 3rd party code or libraries must be properly cited
*/
```

Java classes, methods, and attributes should be documented as well. You may use any format of your choosing but be consistent. Java source code should be indented or "formatted" using the built-in Eclipse Java source code formatter, available from the Source item on the main Eclipse menu bar.