SWE-619: Software Specification and Construction

Spring '25

Course Page https://nguyenthanhvuh.github.io/class-oo

Meetings:	Wed 4:30PM – 7:10PM	**Place:**	ENGR 1103, Piazza (code **TBD**)
Instructor:	ThanhVu Nguyen	**Email:**	tvn@gmu.edu (Piazza is preferred)
Office Hr:	(email first) Mon 2:00PM – 3:00PM	**Place:**	ENGR 4430
GTA:	TBD	**Email:**	tbd@gmu.edu
Office Hr:	TBD	**Place:**	TBD

1 Description

Course Overview This course is designed to give students a solid understanding of modern object-oriented programming (OOP) language, focusing on abstraction, modularity, information hiding, OOP design and specifications, testing and debugging. The students will learn and apply these concepts to develop and test software in the modern software industry. This course is part of the core of the SWE program.

Learning Outcomes By the end of the course, students will gain a solid understanding of the principles of OOP and be able to apply them to develop software systems.

- Understand Abstraction: Explain the role of abstraction in OOP development and how it helps manage complexity in modern software.
- Write and Use Specifications: Develop precise specifications for functions and data types to specify their behaviors.
- Using Mutable and Immutable Types: Understand and differentiate between mutable and immutable data types and make informed design decisions based on the trade-offs.
- Apply OOP techniques: Use OOP design principles, such as inheritance, polymorphism, Liskov Substitution Principle, to effectively structure programs.
- Test and Debug Software: Blackbox testing techniques and debugging strategies to find and fix bugs in software.
- Verify Program: Use formal method concepts and techniques, including assertions, loop invariants, pre/post-conditions, weakest preconditions, Hoare logic, to reason about the correctness of a program.
- Advanced topics (if time permits): concurrent programming (multithreads, synchronotization), model checking and abstraction

1.1 Prerequisite

- SWE Foundation Courses or equivalent.
- OOP Language (e.g., Python or Java)

1.2 Course Materials

- ThanhVu Nguyen. *Formal Software Specification and Reasoning* (**Required**, free)
- Barbara Liskov with John Guttag. *Program Development in Java: Abstraction, Specification, and Object-Oriented Design*. Addison Wesley, 2001. (**Recommend**, free through O'Reilly)[1]
- Luciano Ramalho. *Fluent Python*. O'Reilly Media, 2015. (**Recommended**, free through O'Reilly)

1.3 Assignment Submission and Communication

We will use **Canvas** for submitting assignments and quizzes, and to keep track of grades (§3). It's the student's responsibility to ensure that your grade records are correct.

When submitting assignments, you can either submit a PDF, Word, or text (code) file. If you manually write your answers, take a picture and submit it. **DO NOT** submit link (e.g., to Google Docs or some other services); you will receive a 0 for the assignment if you submit a link.

We will use **Piazza** for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions, I encourage you to post your questions on Piazza.

2 Weekly Schedule

This class is a group-based, *in-person* offering. You need to be present in class at class time. You will also need to schedule regular meetings with your group.

Each week will cover a topic, which is a small number of related technical issues (see course schedule). Each topic will follow roughly the same sequence of preparation, in-class learning activities, homework completion and (possibly) presentation, and knowledge assessment. Many of these activities will be group based. The sequence is:

- Preparation: Complete assigned readings, if any. No formal submission for this activity.
- Class Meeting:
  - Combined session on the day's topic, followed by in-class exercises (group breakouts)
  - Break
  - Combined session on the day's topic, may followed by in-class exercises (group breakouts)
- Homework (group-based)
- Assessments via quiz: takes place at the **end** of class.

[1] Don't worry about the publication date of Liskov, which is basically a math book and therefore ages well.

3 Grading

Assignments	**Percentage**
Group Functioning (group-based)	5%
Homework assignments (group-based)	35%
Weekly Quizzes (individual)	35%
Final exam (individual)	25%
Total	100%

3.1 Scale

A+	$\geq 97\%$	A	$\geq 93\%$	A-	$\geq 90\%$
B+	$\geq 87\%$	B	$\geq 83\%$	B-	$\geq 80\%$
C	$\geq 70\%$	D	$\geq 60\%$	F	$< 60\%$

3.2 Group Functioning

Every student needs to be part of a group. I would prefer that groups stay stable throughout the semester, but if there is a good rationale to reconfigure a group, we'll do that.

Creating groups You will have a chance to form your own group. If you can't find one, we can help. Each has **2 to 4 students**. If your group dwindles to just yourself, you'll need to join another group.

At the end of the semester, each individual will provide an assessment of the rest of their group. This assessment will determine the "Group Functioning" part of the grade (§3).

3.3 Homework Assignments

There are weekly *group homework assignments*, which are given through the class schedule web site. Your group will submit assignments via **Canvas**.

Because of the way in which this class is taught, it is important to stay on pace. Homework assignments are due **before class**. Late submissions are not accepted except in truly exceptional circumstances.

Some important notes:

- Each group should be prepared to present their homework solution in class.
- **Statement of who did what**. Homeworks are group exercises. Each submission must contain a specific statement of who did what.
- There are **no make-ups**.
- Other than the first assignment (where we might not have formed all groups), only one submission per **group**. Everyone in the group gets the same credit.

3.4 Weekly Quizzes

We will have a quiz every week. Each quiz is worth 10 pts. The quiz will be based on the material covered in the previous weeks. Each quiz happens during the last part of class. *You must be present to take the quiz.*

Quiz Make-up Policy You will have the opportunity to make up a quiz if you miss it. The grading and make-up policy is as follows:

- Contact the GTA and schedule a make-up quiz (likely will be offered during TA office hours). The make-up can be different from the quiz given in class, but focuses on the same topics.
- The make-up must be taken promptly and within *a week of the quiz*.
- All quizzes count towards the final grade. Each quiz is scored on a 10 point scale. Missed quizzes score 0/10.
- The maximum possible score on the make-up is 8/10.

3.5 Class Attendance

I place great emphasis on peer learning and interactive engagement. The class is structured to leverage group interactions to the largest extent possible for the purpose of maximizing learning gain throughout the semester. Moreover, the *quizzes are given in class*.

3.6 In-Class Exercises

I plan an in-class exercise for every class. Students will work in their designated group. Very often, the in-class exercises will be closely related to an upcoming homework assignment.

---

4 GMU Policies

4.1 Honor Code

As with all GMU courses, this class governed by the GMU Honor Code. In this course, all assignments carry with them an implicit statement that it is the sole work of the author.

4.2 Learning Disabilities

Disability Services at George Mason University is committed to providing equitable access to learning opportunities for all students by upholding the laws that ensure equal treatment of people with disabilities. If you are seeking accommodations for this class, please first visit `https://ds.gmu.edu/` for detailed information about the Disability Services registration process. Then please discuss your approved accommodations with me. Disability Services is located in Student Union Building I (SUB I), Suite 2500. Email: ods@gmu.edu — Phone: (703) 993-2474