



East China Normal University
GEC International Summer School

COMP303-F: Software Development

Term: June 16th to August 22th, 2025

Class Hours: Monday through Friday, 110 minutes each day (4,400 minutes in total)

Instructor: TBD

Home Institution: TBD

Office Hours: TBD

Email: TBD

Course Description

This course provides a comprehensive exploration of software development principles, mechanisms, techniques, and tools, with a primary focus on software design. Students will delve into foundational principles such as separation of concerns and encapsulation, alongside essential mechanisms like exception-handling and concurrency. Techniques covered include design patterns and unit testing, complemented by practical sessions on well-designed algorithms and UML. By the course's conclusion, students will possess the skills to design and develop robust software solutions effectively, aligning with the course's objective of equipping them with essential tools and insights for success in the field of software engineering. Upon completion, students will demonstrate proficiency in using the proper terminology to discuss Principles, Mechanisms, Techniques, and Tools, understanding their contexts and applications. They will be able to analyze problems and select the most appropriate principles, mechanisms, techniques, and tools for software development. Additionally, students will create complete object-oriented applications, applying software development techniques and utilizing software engineering tools to ensure quality and efficiency.



Course Objectives

By the end of this course, students will be able to:

1. Utilize appropriate terminology to discuss Principles, Mechanisms, Techniques, and Tools in software development.
2. Explain the contextual application of each principle and mechanism.
3. Apply software development techniques in a timely manner to address diverse programming challenges.
4. Analyze problems and select the most suitable principles, mechanisms, techniques, and tools for software development.
5. Develop complete object-oriented applications, integrating principles of object-oriented software development and structured programming language mechanisms, while employing software engineering tools for quality assurance.

Recommended Reading

Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional.

ISBN 978-0134757599.

Course Hours

The course has 40 class sessions in total. Each class session is 110 minutes in length, for a total of 4400 minutes of in-class time. The course meets from Monday to Friday. ECNU awards 3 credits for this course. Different universities may count course credits differently. Consult officials at your own home institution.

Attendance

Summer school is very intense and to be successful, students need to attend every class. Occasionally, due to illness or other unavoidable circumstance, a student



may need to miss a class. ECNU policy requires a medical certificate to be excused. Any absence may impact on the student's grade. Moreover, ECNU policy is that a student who has more than 3 absences will fail the course. Arriving late or leaving early will count as a partial absence.

Grading Policy

ECNU awards grades of A, A-, B+, B, B-, C+, C, D, and F. Most colleges and universities do not award transfer credit for grades of D or F.

In this course, grading will be based on the following:

Assignments	15%
Test 1	20%
Test 2	20%
Final Exam	45%

General Expectations

Students are expected to:

- Attend all classes and be responsible for all material covered in class and otherwise assigned. Any unexcused absence may impact a student's grade.
- Arrive to class on-time: Late arrivals are disruptive to your fellow students and to the conduct of the class.
- Complete the day's required reading and assignments before class.
- Review the previous day's notes before class; make notes about questions you have about the previous class or the day's reading.
- Refrain from texting, phoning or engaging in computer activities unrelated to class during class (不要用手机). It is highly disrespectful to the professor and to the class.
- Participate in class discussions and complete required written work on time.

Course Schedule

The planned schedule sketched out below may be modified to suit the interests or abilities of the enrolled students or to take advantage of special opportunities or events that may arise during the term.



Week 1: Fundamentals of Software Development and Design Principles

- Day 1
 - *Exploring the Concept of Effective Design: Emphasis on simplicity, efficiency, accuracy, coding conventions, and the role of comments as documentation*
- Day 2
 - *Principles of Object-Oriented Design: Delving into class construction, encapsulation, polymorphism, inheritance, and the significance of abstract classes*
- Day 3
 - *Advanced Object-Oriented Concepts: Investigating object identity, life-cycle management, the object model, and the utilization of generic types*
- Day 4
 - *Practical Application of Design Principles: Implementing design by contract principles and exploring graphical user interface (GUI) development with Swing*
- Day 5
 - *Tutorial/Discussion*
 - *Assignment: Practical project: Design and implement a simple object-oriented application, applying learned principles and techniques*
 - *Lab (Optional)*

Week 2: Tools and Techniques for Effective Software Design



- *Day 1: Fundamentals of Software Testing and Documentation*
 - *Using jUnit for automated testing*
 - *Writing and generating documentation with javadoc*
- *Day 2: Fundamentals of Software Testing and Documentation*
 - *Importance of error handling and maintaining software quality*
- *Day 3: Advanced Modeling with UML*
 - *Creating and interpreting class diagrams*
 - *Understanding and using sequence diagrams*
- *Day 4: Advanced Modeling with UML*
 - *Developing activity diagrams to model workflows*
 - *Practical exercises using UMLet to design software systems*
- *Day 5*
 - *Test 1*
 - *Assignment: Practical project: Test and document using automated testing tools, master error handling mechanisms*

Week 3: Utilizing Static Analysis and Coverage Tool

- *Day 1*
 - *Introduction to Integrated Development Environments (IDEs)*
 - *Exploring features of modern IDEs (e.g., IntelliJ IDEA, Eclipse)*
- *Day 2*
 - *Setting up projects, navigating the interface, and using version control*
- *Day 3*



- *Understanding the importance of code quality and static analysis*
- *Introduction to tools like SonarQube, Checkstyle, and FindBugs*
- *Day 4: Design Patterns*
 - *Overview of design patterns and significance in software development*
 - *Exploring basic design patterns: Composite, Decorator, Iterator, Singleton*
- *Day 5*
 - *Test 2*
 - *Lab (Optional)*

Week 4: Design Patterns

- *Day 1*
 - *Understanding the Model-View-Controller (MVC) pattern and its components*
 - *Introduction to the Observer pattern and its implementation*
 - *Implement MVC and Observer patterns*
- *Day 2*
 - *The Strategy pattern and its use cases*
 - *Understanding the Template Method pattern and its advantages*
 - *Applying Strategy and Template Method patterns in practical scenarios*
- *Day 3*
 - *Threads and Synchronization: Exploring producer-consumer problem and locks for synchronization*



- *Day 4*
 - o *Advanced Design Patterns: the Visitor pattern and its role in traversing complex data structures*
- *Day 5*
 - o *Assignment: Practical Project: Automated Testing and Documentation Using Testing Tools, Mastery of Error Handling Mechanisms*
 - o *Lab (Optional)*

Week 5: Project Development and Conclusion

- *Day 1*
 - o *Advanced Design Patterns: the Factory Method pattern and its variants (Adapter, Command)*
 - o *Practical examples and group discussions: Applying advanced design patterns*
- *Day 2*
 - o *Concurrency and Synchronization*
 - o *Exploring synchronization techniques like locks and semaphores*
 - o *Practical examples demonstrating concurrent programming*
- *Day 3*
 - o *Review course highlights, clarify doubts, reinforce knowledge*
- *Day 4*
 - o *Tutorial/Discussion*
 - o *Final Exam Review Session*
- *Day 5*



- *Final Exam*
- *Lab (Optional)*

Academic Honesty

Students are expected to maintain high standards of academic honesty. Specifically, unless otherwise directed by the professor, students may not consult other students, books, notes, electronic devices or any other source, on examinations. Failure to abide by this may result in a zero on the examination, or even failure in the course.