



# CS309

## OBJECT-ORIENTED ANALYSIS AND DESIGN

---

Yuqun Zhang (张煜群)

Department of Computer Science and Engineering  
Southern University of Science and Technology

# Who and Where Am I?

- Dr. Yuqun Zhang (张煜群)
- Research Interests: Software Analysis and Testing (Fuzz Testing, GPU programs, Defect Prediction, Code Summarization, etc.)
- Email: [zhangyq@sustech.edu.cn](mailto:zhangyq@sustech.edu.cn)
- Office: Room 604, 创园10栋
- Office Hours: 4-6pm, Thursday, or appointment by email

A LITTLE SOMETHING  
ABOUT ME...

---

# My Styles and Rules

- Casual
- Interaction
- Mutual Respect
- **NO CHEATING!!!!**
  - You may work together in this class, as specified on each specific assignment. Do **NOT** use any resource without citation.

# Instructor and Teaching Assistant

- Yueming Zhu (朱悦铭)
  - Email: zhuym@sustech.edu.cn
- Tiancheng Yu (余添诚)
- Jiachen Zhang (张佳晨)
- Zheng Bian (卞证)
- Chaozu Zhang (张超祖)
- Xinghe Yao (姚星河)
- Qingqiao Hu (胡青翹)
- Zhiwei Wang (王智伟)
- Yiming Chen (陈毅铭)
- Yubin Hu (胡玉斌)

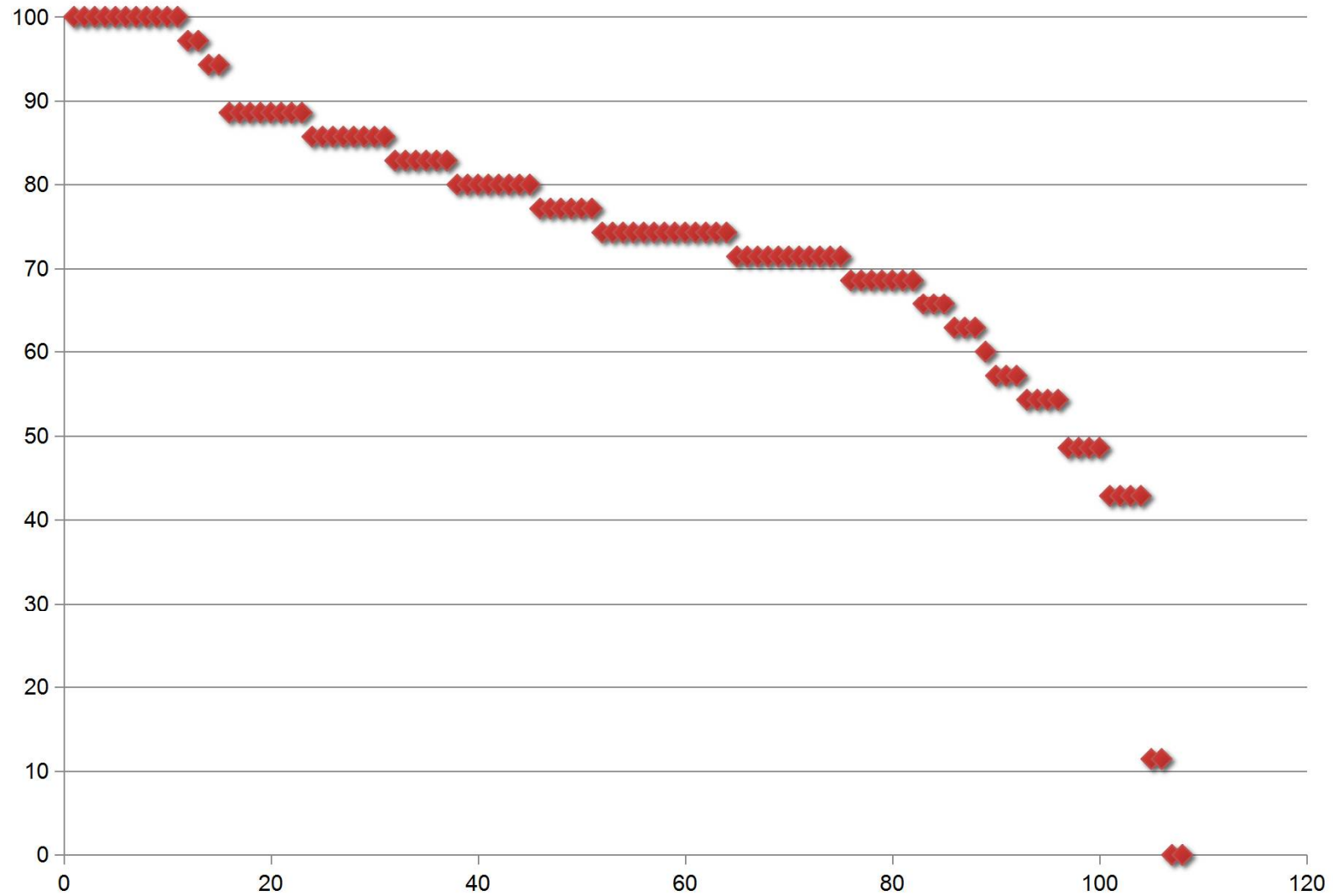
# Textbooks

- Freeman et al., *Head First Design Patterns*
- Martin Fowler, *Refactoring*
- Block, *Effective Java*
- Zeller and Krinke, *Essential Open Source Toolset: Programming with Eclipse, JUnit, CVS, Bugzilla, Ant, Tcl/TX and More*
- McConnell, *Code Complete: A Practical Handbook of Software Construction*
- Pilone, *UML 2.0 Pocket Reference*

# Evaluation and Grading

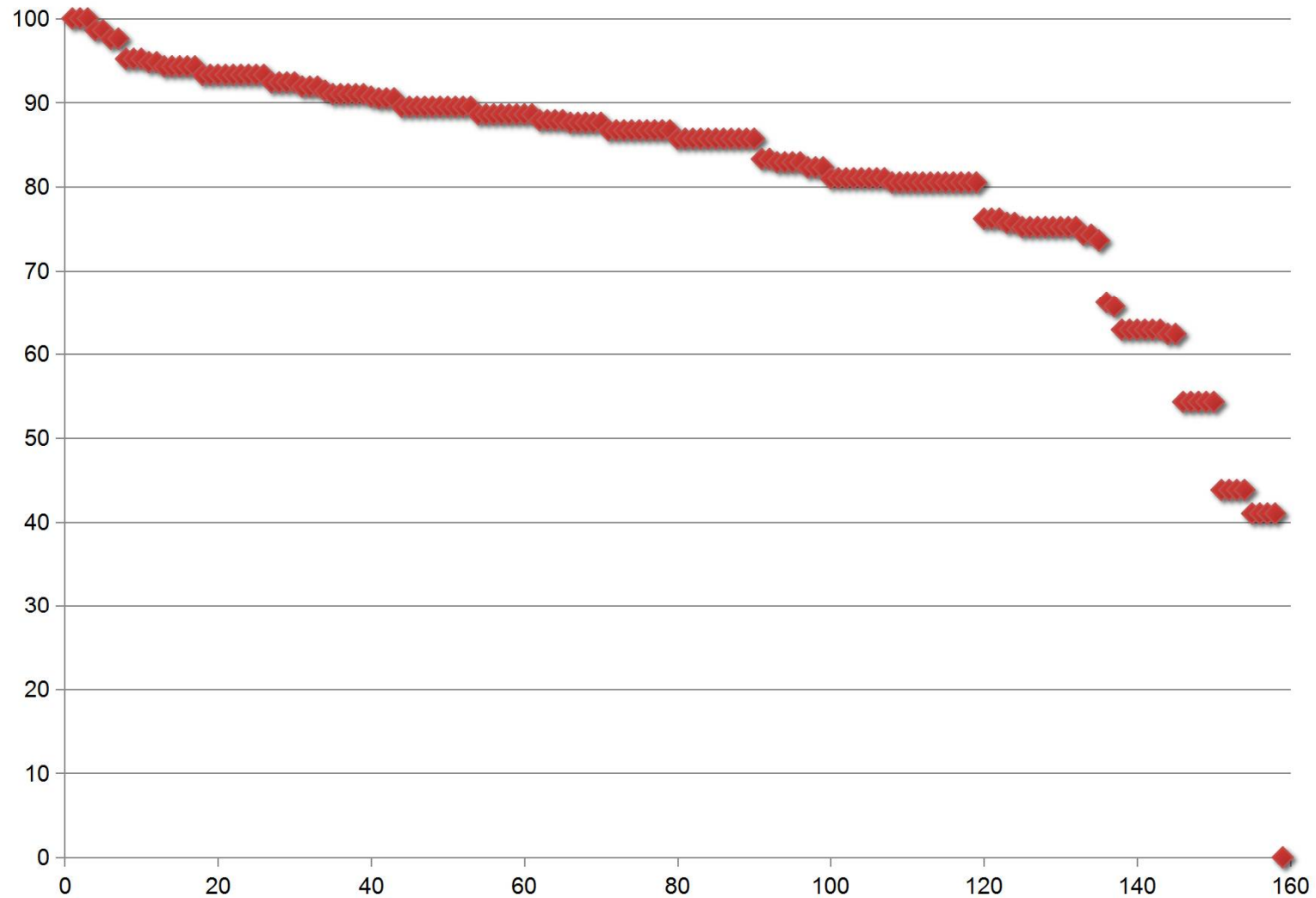
- Weekly Lab Tutorials– 25%
  - ~5 times
- Project – 35%
  - Web applications/Games
  - Group of 4 to 5 (before the end of next week)
  - 3 presentations (proposal, progress, final)
  - 2 written reports
- Exams – 35%
  - Final:
    - What's on an exam? Anything from any aspect of class, including lab sections.
    - No hints (重点)
- In-Class Exercises/Attendance – 5%
  - Spontaneous (That means in general I do not call the roll. But I have my own moves 😊 😊 )

# Project in 2017 Fall

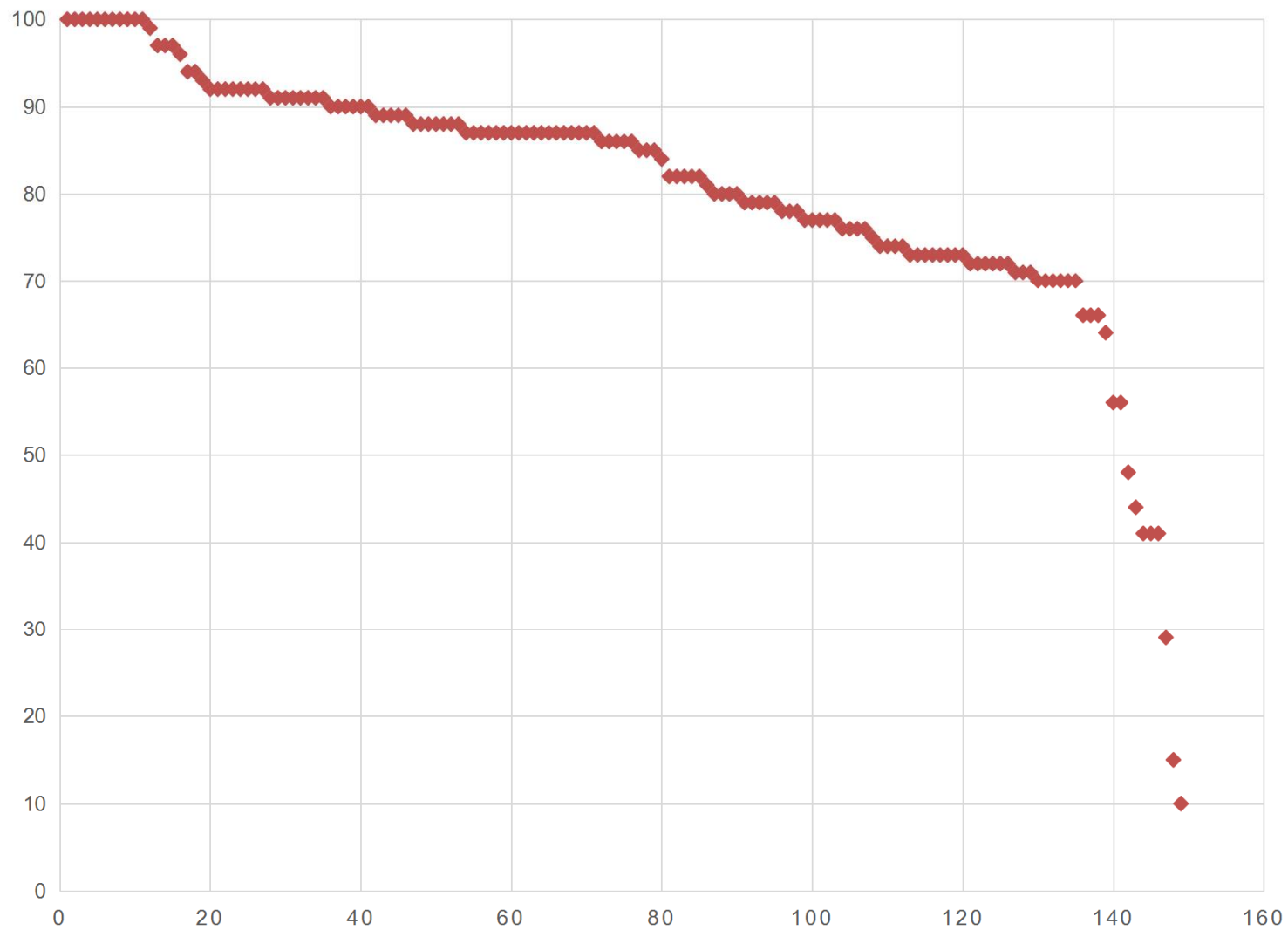





# Project in 2018 Fall



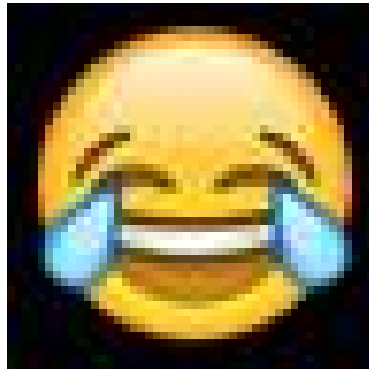
# Project in 2019 Fall





You could say this is a “breathing” class (not a “水” class).

I just want you to be happy in this semester.



ALRIGHT, LET'S GET REAL

---

# Expectations

- You're going to have to “own” your education in this class
  - I have a feeling this is going to be an awesome semester...
- But...
  - Expect that I may not be able to give you an immediate answer (I'm alright if my response to your question is “I don't know,” so you're going to have to be alright with that, too)
  - I (or the TAs) WILL always try to help find you the answers you need in a timely fashion. Be patient.

# Tips of Handling Problems

- Once you encounter problems (theory or practice), you are expected to
  - first, try your real best to solve them by yourself

# Tips of Handling Problems

- Once you encounter problems (theory or practice), you are expected to
  - first, try your real best to solve them by yourself
  - if not working, try to talk with your cohorts.

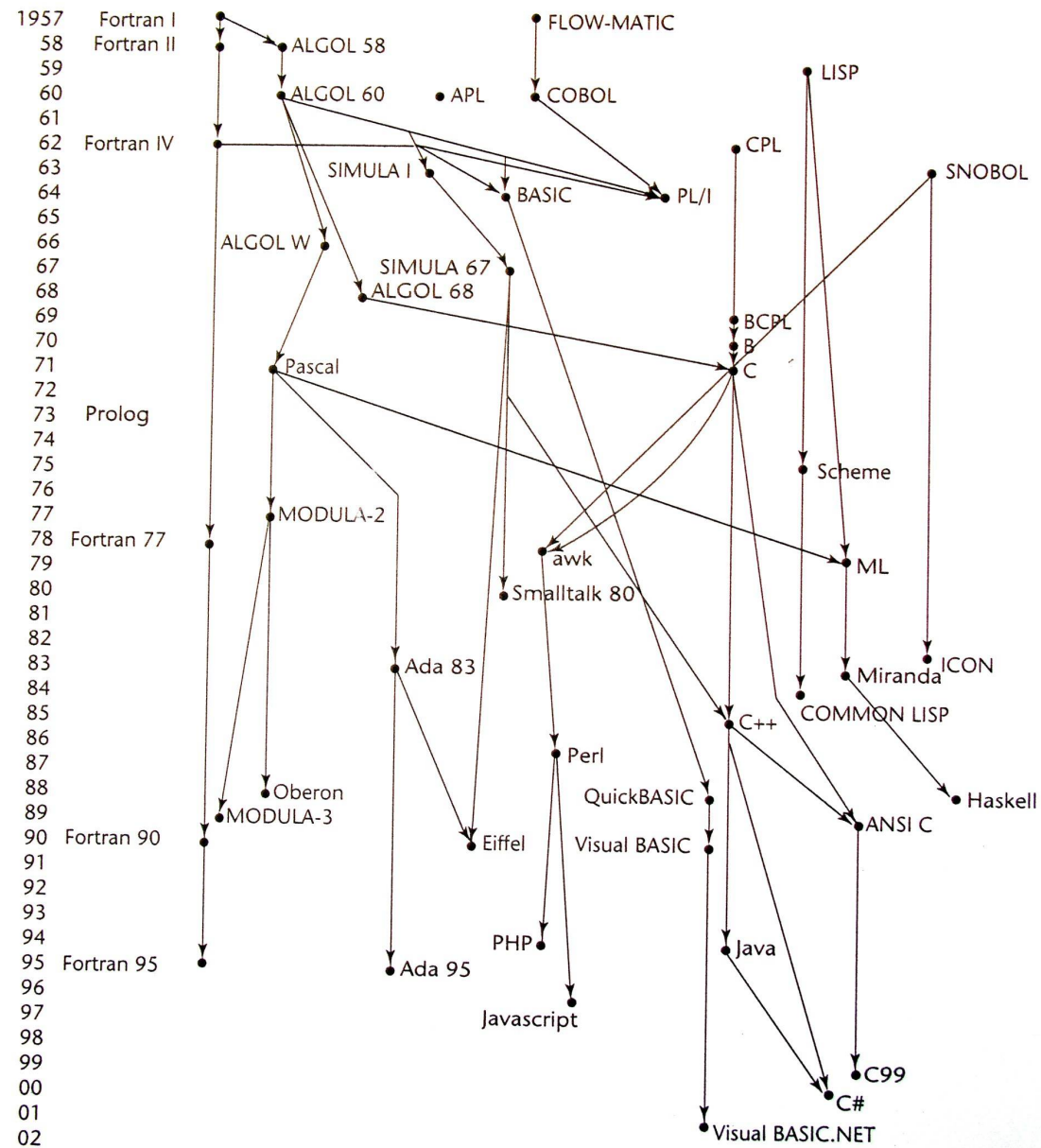


# Tips of Handling Problems

- Once you encounter problems (theory or practice), you are expected to
  - first, try your real best to solve them by yourself
  - if not working, try to talk with your cohorts.
  - if not working, then ask us

# Tips of Handling Problems

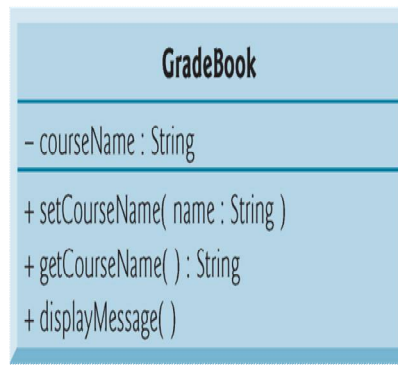
- Once you encounter problems (theory or practice), you are expected to
  - first, try your real best to solve them by yourself
  - if not working, try to talk with your cohorts.
  - if not working, then ask us
- If we find that you are not paying effort by yourself, we would be reluctant to help you at later time.



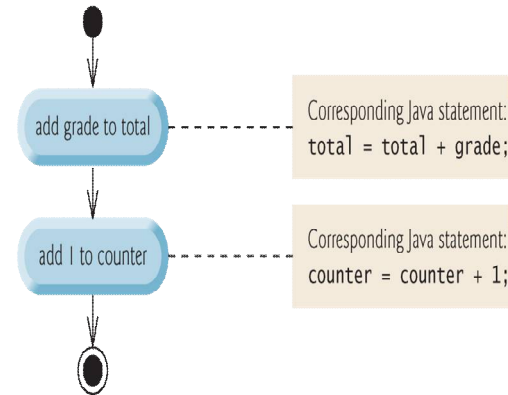
# What You Would Learn

- Of course the object-oriented design and analysis
- Typically, you are going to learn something about
  - requirement engineering (UML)

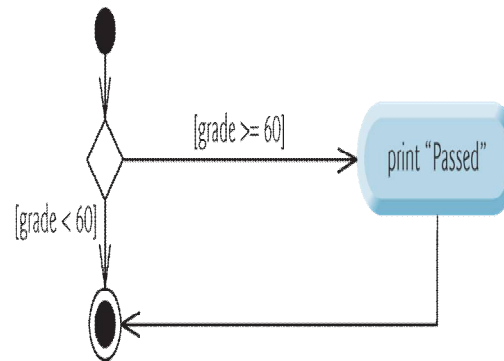
# UML Examples



**Fig. 7.3** | UML class diagram for class GradeBook.



**Fig. 3.1** | Sequence structure activity diagram.



**Fig. 3.2** | if single-selection statement UML activity diagram.

# What You Would Learn

- Of course the object-oriented design and analysis
- Typically, you are going to learn something about
  - requirement engineering (UML)
  - design pattern (including information hiding, design principles, etc)
  - refactoring
  - web app frameworks? testing?

# Your Projects

- Each group picks one problem from a pool
- We created 6 projects for you. Yet you can work on your own if you want to.
- Come talk to me if you want to come up with your own ideas. DO EXPECT THAT YOUR IDEAS MIGHT BE ASSIGNED WITH A LOWER STARTING SCORE.
- One contact person is needed for each group (**This year, almost all the project requirements are proposed by student assistants. It is pretty necessary to contact them during your progress. They are the boss!**).

# The Project List

- **Project Helper**: Build a website for assisting teachers and students to access course projects
- **SEMojo**: Build a website to provide services of advanced software engineering techniques (automated testing, repairing, code generation, code summarization, etc.).
- **DBOJ**: Build an online judge system that can be specifically adopted by the Database projects
- **SUSTech Gamer**: Build a website for accessing the games developed by SUSTechers
- **SUSTech Tower Defense**: Design a tower defense game with Sustech as components.
- **SUSTech Landlords**: Design a landlord game with Sustech as components



# Project Helper:

- You need to support two user groups: teachers and students.
- For teachers,
  - setting the project groups, including the group size, available presentation time, oversized groups
  - releasing and updating the project notices, to all or partial students.
  - online grading the projects/assignments and uploading the mark files with the "excel" or "CSV" formats.
  - randomly or semi-randomly grouping students
  - sending comments/grade/messages to students on time (through mails associated with student accounts)
  - recording the project submission information
  - allowing students to sign up for presentation
  - authorization management

# Project Helper:

- For students,
  - updating personal information, including technology stack, skill set, desired group members
  - filling the group information, including advertisements for group members
  - completing presentation time and member information
  - submitting the overall project with access authorization
- Potential bonus,
  - adding the role of admin
  - fancy UI
  - student evaluation and score record for each project
  - information synchronization when changing the project group
  - CAV authorization associated with the account
  - deployed as wechat App
- Stakeholders for Requirements:
  - Yiming Chen
  - Chaozu Zhang

# SEMojo:

- You need to support three user groups: admin, contributors, and customers
- For admin,
  - managing (approving/adding/deleting/...) contributor/user signups
  - managing the uploaded techniques, e.g., inspecting/approving/disapproving the uploaded techniques, ensuring that the uploaded techniques are runnable
  - interacting with contributors and users
- For contributors, (you might want to refer to the GitHub designs)
  - signing up/in the website
  - managing personal information
  - uploading the advanced software engineering techniques, including source code, test cases, documents, etc.
  - updating/deleting their own techniques voluntarily or as required.
  - interacting with admin and users.

# SEMojo:

- For customers,
  - signing up/in the website
  - managing personal information
  - downloading the techniques
  - feedbacking the techniques
  - interacting with admin and contributors
- You need to apply at least two real-world software engineering techniques in your website, one by your own search and one provided by us, and make sure they can be successfully executed.
- Possible bonus (inclusive but not restricted):
  - Admin determines whether the uploaded techniques incur copyright violations
  - More GitHub features are favored.
  - Payment options
  - Enabling SaaS platform
  - More techniques installed
- Stakeholders for Requirements:
  - Yubin Hu
  - Jiachen Zhang

# DBOJ

- The platform should support multiple SQL query execution, such as SQLite, PostgreSQL (at least these two).
- You need to support three user views.
  - For regular students, the view only should be the submitted history records and current execution results. Maybe adding the submitted time ranking table is also fine. (The limitation for the memory usage and the running time should be declared to the regular users and well maintained on the backend)
  - For the admin user, the view should include all the submitted history records from all the regular users. Moreover, the admin user could also see more statistical graphs, such as the submitted time ranking table, the precision rate, and so on. Moreover, the admin user should be able to create the assignments and publish the assignments, or the test examples. The admin user should also be able to publish the possible solution for a certain assignment. The admin user could authorize other users.
  - For student assistants, they could have the same view as the admin user with lower authority (e.g., the student assistants cannot create assignments or publish ones.)
- You need to take care of security issues. Suspicious SQL code cannot be executed by regular users. Actions cannot be transported without encryption.

# DBOJ

- Task taxonomy
  - basic:
    - SQL query execution
    - Multiple SQL dialect supports, e.g., sqlite, PostgreSQL
    - At least two authority level supports
    - The code review function
    - The execution results and the credits for regular users (students)
  - upgraded:
    - The admin user could publish assignments on the platform.
    - The admin user could modify and delete the assignments.
    - The statistical graphs and tables for the admin user should be supported.
  - potential bonus (inclusive but not restricted):
    - The previous user states should be kept when the user re-logins.
    - Multiple authority level supports.
    - Multiple device supports
- Stakeholders for Requirements:
  - Xinghe Yao
  - Qingqiao Hu

# SUSTech Gamer

- You need to support three user groups.
- For admin,
  - creating new users, assigning user roles, modifying user profiles and moderating the community
  - offering detailed documentation to both users and developers about how to use the platform
- For developers,
  - uploading their games to the platform for users to download and set price, also a game's announced (visible) / release (downloadable) date.
  - setting price/discount (percentage and time period) for each of their games. Payment should be added to the developer's account.
  - releasing multiple branches of a game for users to select which branch they want to play
- For users,
  - signing up/in
  - showing profiles, uploading avatar and setting their bio for other users
  - providing and categorizing a list of all available games for users to select a category to view its games
  - allowing users to purchase any available game using their account balance

# SUSTech Gamer

- For users,
  - the store page for each game should be able to add videos, screenshots and detailed descriptions
  - downloading their games (web or client)
  - After buying a game/DLC, users should be able to leave comments and scores which can be demonstrated in the store page.
- Potential bonus (inclusive but not restricted)
  - DRM System: offering an SDK in popular programming languages(s) to provide API(s) for game developers to verify the user, get user's unique ID, name, avatar, etc. and all the user-owned DLC
  - Cloud Save: offering settings to developers to enable cloud-saving on their games
  - Friends: offering an SDK with API(s) for developers to obtain user's friends and invite other friends to join their game, i.e., facilitating multiplayer-friendly games.
  - DLCs: adding DLC (downloadable contents) to their games.
  - Incremental upgrade: downloading and applying the incremental upgrade (the difference between an older version and the latest version).
- Stakeholder for requirements
  - Tiancheng Yu



# SUSTech Tower Defense



# SUSTech Tower Defense

- You need to make sure that your fundamental game logic is tower defense. Accordingly, you can expand the logic as you wish, for instance,
  - exploring the diversity of enemies and towers, Such as Plants vs. Zombies or Kingdom Rush
  - introducing heroes to make the game similar as Honor of Kings
  - more movable characters similar as Red Alert or Warcraft
- Standalone required. Online optional.
- Basic:
  - 3D game
  - outsourcing allowed, e.g., art materials, yet limited
  - runnable and complete
- Upgraded:
  - plentiful game logics
  - concise game structure (e.g., use OOAD design patterns)
  - beautiful layout

# SUSTech Tower Defense

- Potential bonus (inclusive but not restricted):
  - as funny as possible
  - fancy framework/tools for development
  - more features
- Stakeholder for requirements:
  - Zhiwei Wang

# SUSTech Landlords



# SUSTech Landlords

- Basic (70%):
  - Running smoothly. No evident bugs.
  - Designing your logics of playing cards with SUSTech as your components
  - At least four logics of playing cards (inclusive but not restricted)
    - 双王，炸弹，单卡（单张牌），对子（两张相同的牌），三张牌
    - 三带一，三带二
    - 单顺：连续五张或更多张牌。不包括2和大小王(例如34567，10 JQKA)
    - 双顺：三对或更多连续对牌。不包括2和大小王(如778899，1010JJQQKK)
    - 三顺：连续两张或两张以上的牌。不含2分和双旺(如555666、777888999)
    - 飞机带翅膀：三顺+同数量的单牌或对牌(例如555666带7带8，55666带77带88)
    - 四带二：四张相同值的卡+两张单卡或对面卡(如4444带3带5、4444带33带55, 这样就不是炸弹了)

# SUSTech Landlords

- Potential bonus (30%, inclusive but not restricted):
  - plentiful ways of playing cards
  - opening animations, CGI
  - 癞子玩法
  - beautiful UI
  - gamepad, VR-oriented
  - game store, payment players
  - standalone (smart AI player)
  - client-server game
  - other modes of playing cards, such as 炸金花, 钓鱼, 比大小, 21点, 十点半, 拖拉机
- Stakeholder for requirements
  - Zheng Bian

# Tips for your projects

- Frequently contact your stakeholders. They manage the requirements and have written detailed descriptions. (This year's new)
- Launch your projects ASAP. You don't want to start off just two weeks before the final ddl.
  - Whoever accomplish the projects and present them on mid-term presentation can be awarded with a bonus of 10% of your final project score. (This year's new)
- If you want to be better graded, you should go for as many bonus points as possible.
- You need to run your deliverables with test cases.
- Don't simply rely on the technical leader (大腿) in your team. We would grade you based on your individual contributions to the team in an even more rigorous manner than before. (This year's new)

QUESTIONS?

---