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, Testing, and Security (Fuzz Testing, Taint Analysis, Defect Prediction, Software Component Analysis, etc.)
- Email: zhangyq@sustech.edu.cn
- Office: Room 610, Engineering Building South
- 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
- Yitong Wang (王奕童)
- Yiwei Ren (任艺伟)
- Linkai Peng (彭琳凯)
- Shangxuan Wu（武尚萱）
- Hengchen Yuan (袁恒宸)
- Tianyunxi Wei (魏田纭溪)
- Chi Xu (徐驰)
- Runzhe Jiang (蒋润喆)
- Peiqi Yuan (原佩琦)

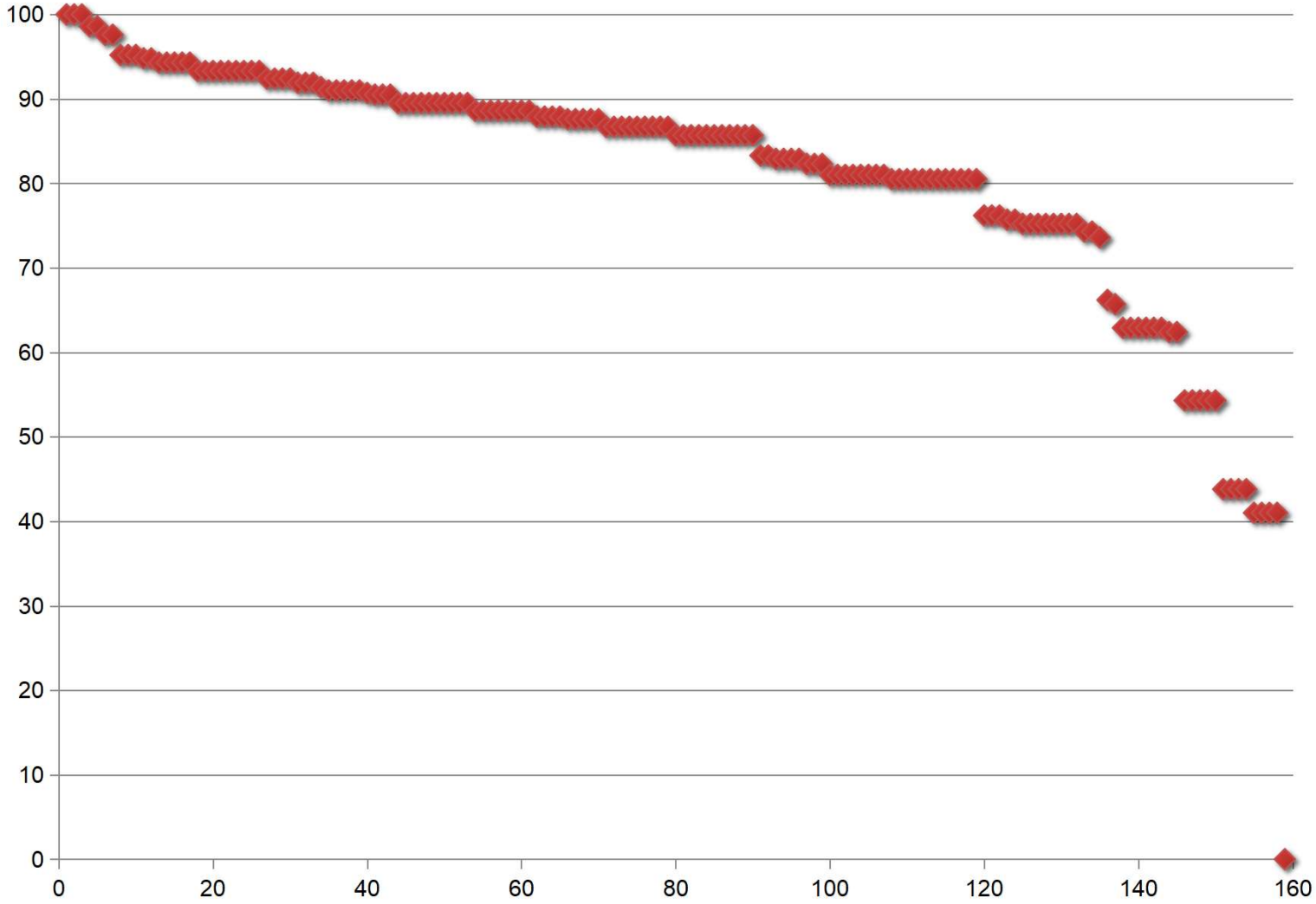
Textbooks

- 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*

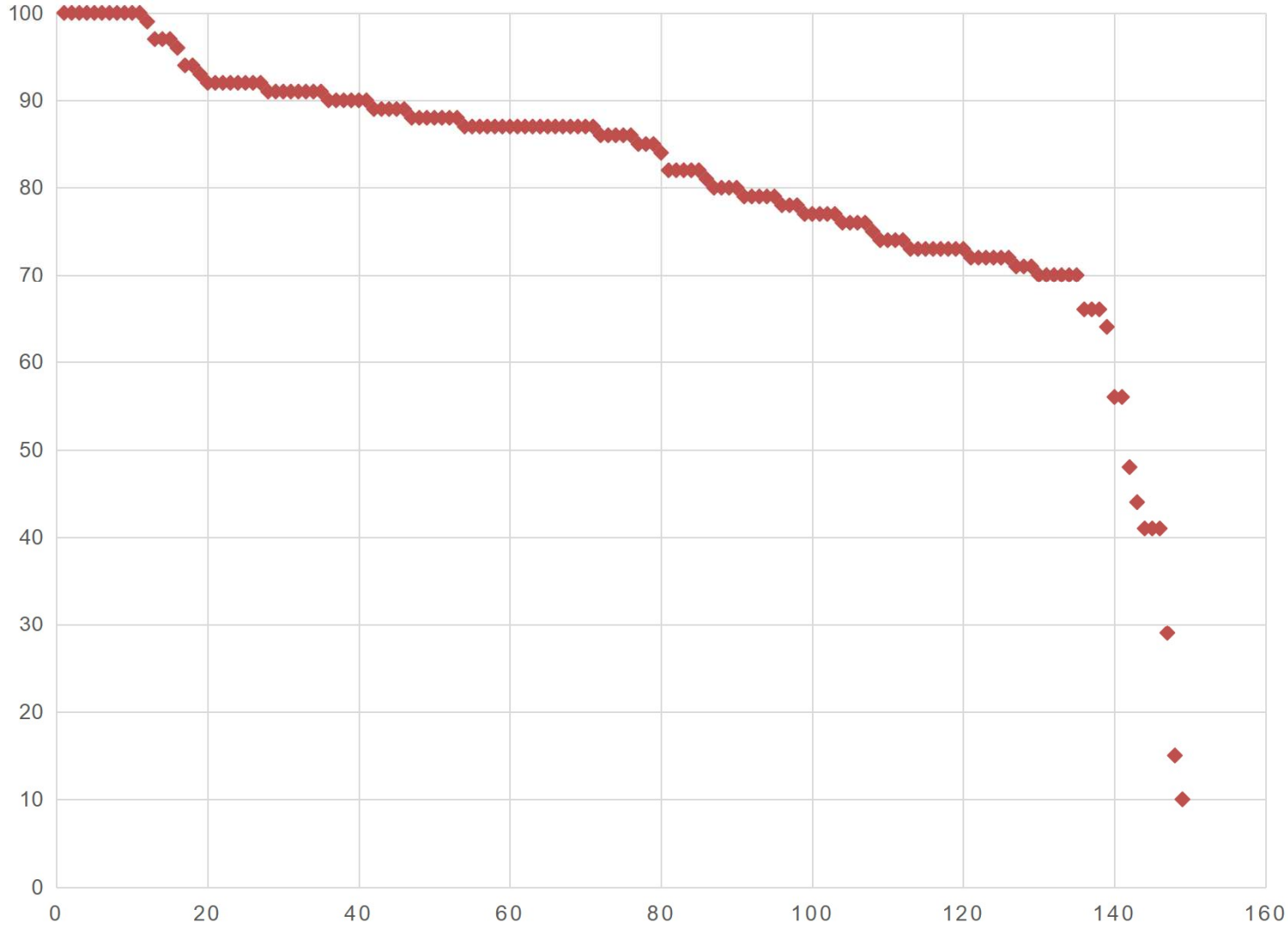
Evaluation and Grading

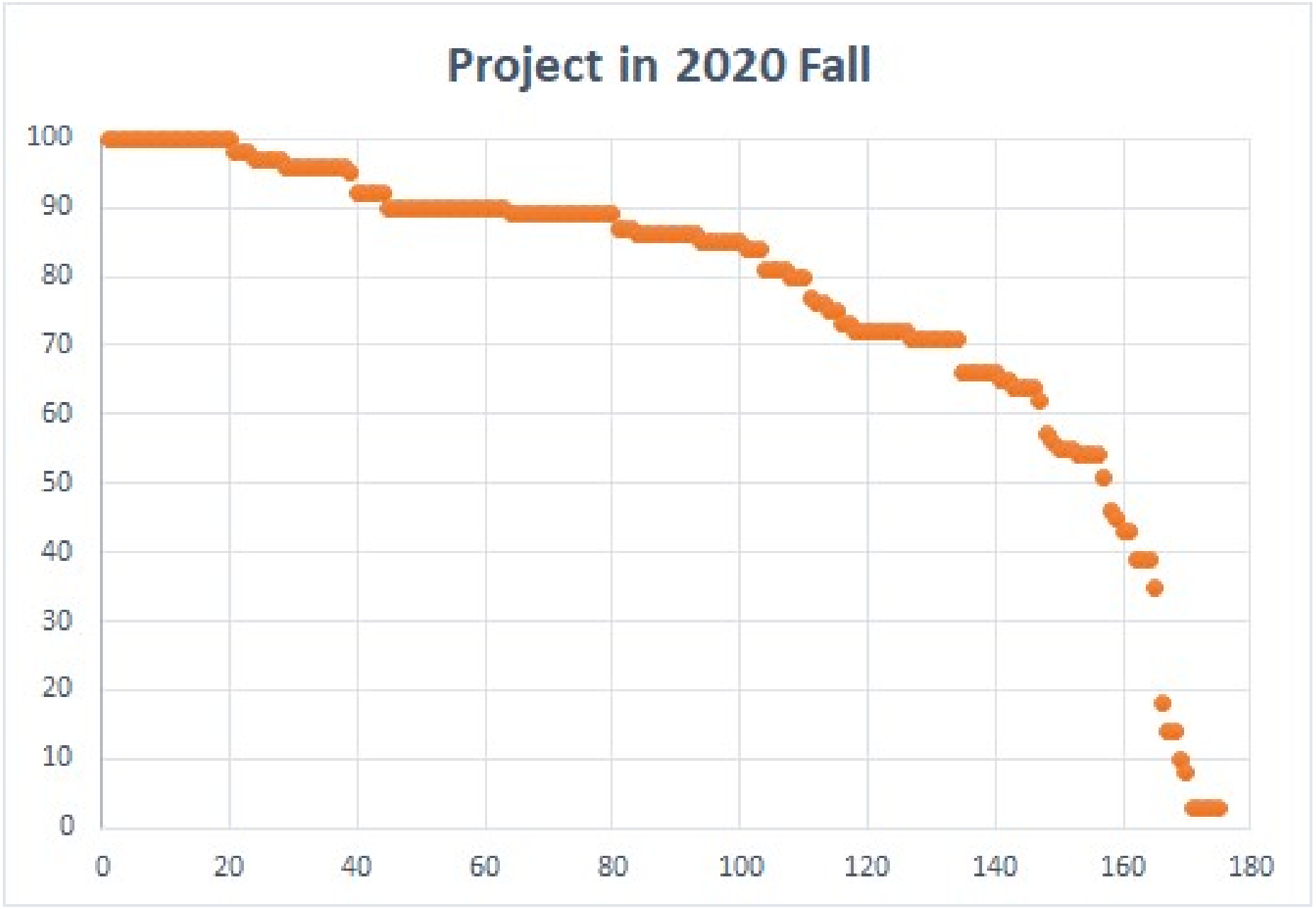
- Weekly Lab Tutorials– 25%
  - ~5 times
- Project – 35%
  - Web applications/Games
  - Group of 4 to 5 (before the end of next week)
    - We have ~220 registered students, so please be subject to the group size!!
  - 3 presentations (proposal, progress, final)
  - 1 written report
- Exams – 35%
  - 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 2018 Fall



Project in 2019 Fall




Project in 2020 Fall
100
90
80
70
60
50
40
30
20
10
0
0
20
40
60
80
100
120
140
160
180

You could say this is a “breathing” class (not a “氵水” 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

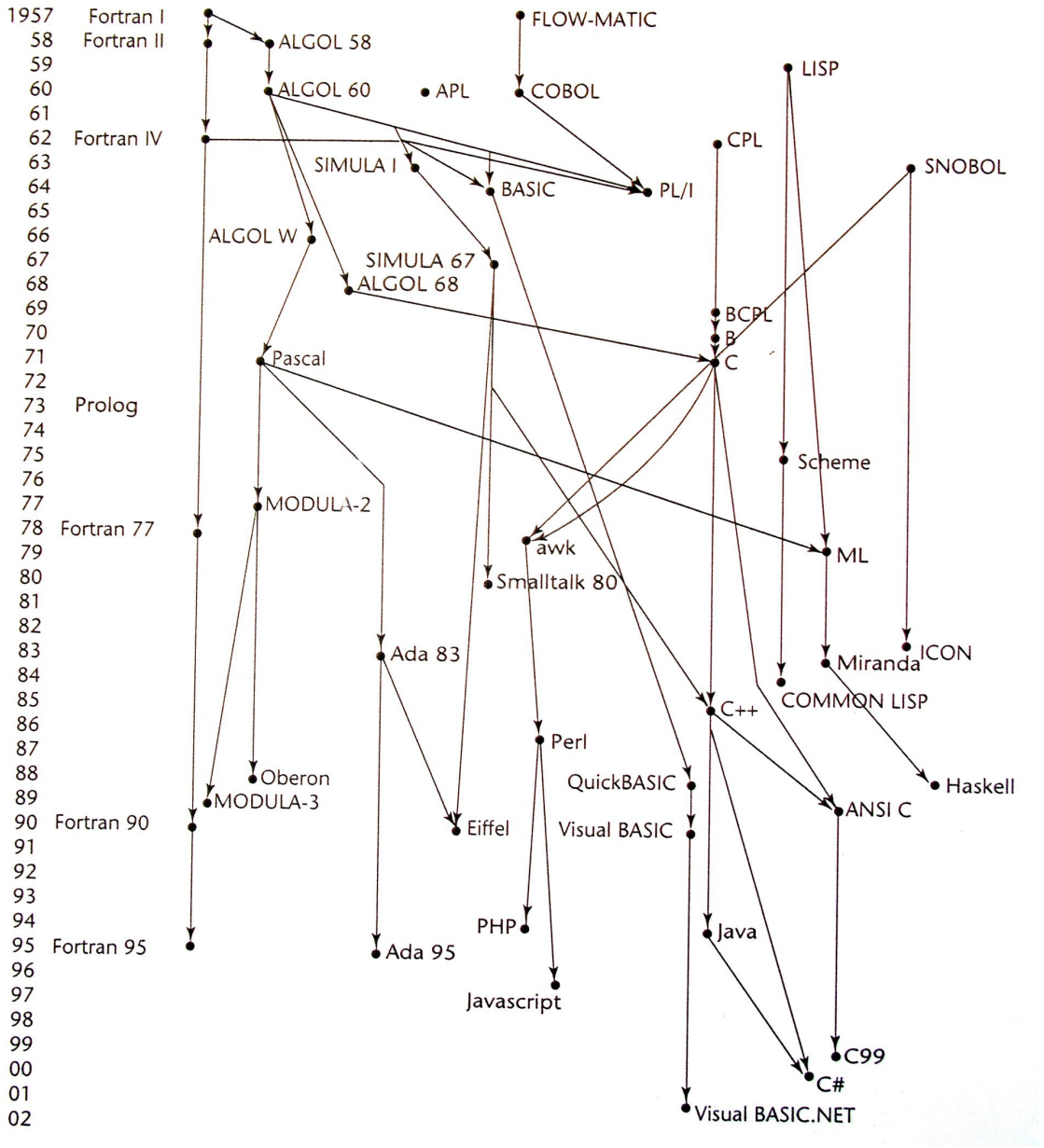
- Once you encounter problems (theory or practice), you are expected to
  - first, try your real best to solve them by yourself
  - if not working, try to talk with your cohorts.

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.


1957
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
00
01
02
Fortran I
Fortran II
Fortran IV
Prolog
Fortran 77
Fortran 90
Fortran 95
FLOW-MATIC
ALGOL 58
ALGOL 60
APL
COBOL
LISP
CPL
SNOBOL
SIMULA I
BASIC
PL/I
ALGOL W
SIMULA 67
ALGOL 68
BCPL
B
C
Pascal
Scheme
MODULA-2
awk
ML
Smalltalk 80
Ada 83
ICON
Miranda
COMMON LISP
C++
Perl
QuickBASIC
Haskell
Oberon
MODULA-3
ANSI C
Eiffel
Visual BASIC
PHP
Java
Ada 95
Javascript
C99
C#
Visual BASIC.NET

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

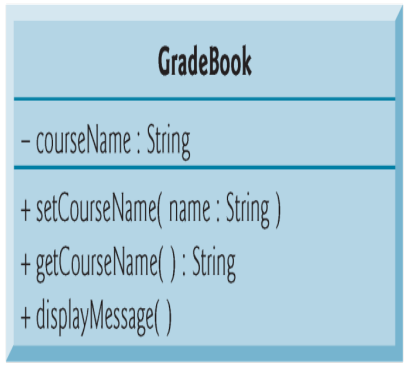


Fig. 7.3 | UML class diagram for class GradeBook.

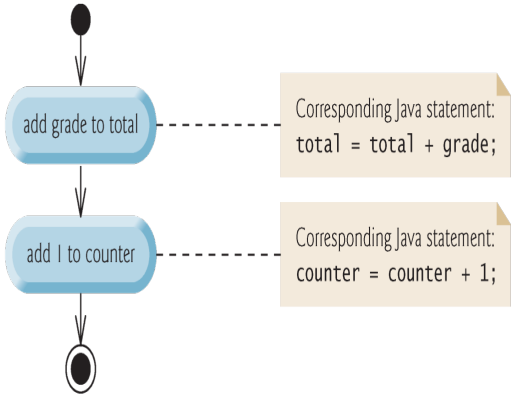


Fig. 3.1 | Sequence structure activity diagram.

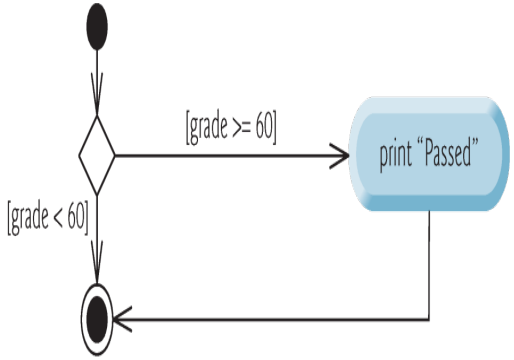


Fig. 3.2 | `if` single-selection statement UML activity diagram.

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 5 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, all the project requirements are proposed by student assistants. It is pretty necessary to contact them during your progress. They are the bosses!).

The Project List

- SUSTech Live Video Streamer: Build a website where students can learn courses via online video streams
- Website for Chain Hotel: Build a website to manage the customer transactions on chain hotels.
- Code Management Platform: Build a website to manage code (similar as GitHub)
- SUSTech War Chess: Build a game with role-playing strategy that moves characters on a map to fight (better involving SUSTech components).
- SUSTech Shooting Game: Build a FPS shooting game (better involving SUSTech components).

SUSTech Live Video Streamer:

- Basic Requirement：
  - Support three login permissions: system administrator, course teacher and user
  - Course creation needs to be applied by the course teacher and approved by the system administrator
  - The course consists of the following five modules:

    1. Video module: Teachers can upload videos and users can watch them online.

    2. Question-making module: Course teachers can assign questions (selection, judgment) and users can answer questions online. The answering process should be time-limited.

    3. Assignment module: Course teachers can post assignments. Users can upload assignment files, and the uploaded files can be viewed online.

    4. Payment module: For paid courses, users need to pay to watch.

    5. Notification module: Teachers can post notifications in the course and choose whether to send emails.

SUSTech Live Video Streamer:

- Basic Requirement：
  - The system needs to implement following functions:

    1. The course contains different chapters and teachers can set a score for each chapter. Users can see the completion progress and the score of each chapter in real time.
    2. Teachers can manage users and see their learning status in the background
    3. Instructors can export users' per-chapter scores
- Advanced Requirement:
  - The resolution of the video being played can be adjusted according to the network speed (or implement a CDN-like function).
  - You can implement anti-cheating function. For example, users must watch a certain time, only one video can be played at the same time. Users cannot directly call the end of the course request, etc.
  - You can realize the video comment area or the bullet screen sending function, which is convenient for teachers and students to post comments.
  - You realize beautiful pages.
- Stakeholders: Shangxuan Wu, Yiwei Ren

Website for Chain Hotel:

- Basic Requirement：
  - Chain hotel hierarchy：
    - Support multiple cities and different branches
    - Each hotel has various types of rooms, and the same type has the same price
    - Each type has one or more rooms, with different locations
  - Consumer: can select one branch hotel and view some basic information of any room type available, query rooms by specific filtering criteria, and reserve rooms.
  - Hotel administrator: can modify the basic information of any room, query customer booking records.
  - Hotel floor plan: consumers can view the floor plan and book a specific room by clicking it on the plan.
  - Order management system: Consumers can view their own orders, modify the basic information of the order, and give some comments about the service (text, picture, satisfaction score) after checking into the hotel.
  - Message notification system: consumers can consult hotel customer service and receive notifications of special events (successful reservation, cancellation, etc.).
  - Virtual account payment (or invoke WeChat Pay/Alipay and other electronic payment API).
  - The hotel administrators can query some statistics about turnover, room bookings and so on.

Website for Chain Hotel:

- Advanced Requirement:
  - Beautify the webpage, especially the hotel room plan, chart or a more graphic way to show the turnover calculation.
  - Launch time-limited flash kill and promotion activities for specific room types, or launch some discount coupons.
  - High concurrency processing, when the traffic surges, there will not appear too high latency or data confusion.
  - Collection function.
  - Map search function: search the city on the map first, and show the location of each branch in the city.
  - Points system, which can be used to exchange gifts.
  - Upload videos to rate rooms.
  - Support PC layout mode, making the page beautiful and not messy.
- Stakeholders: Peiqi Yuan, Tianyunxi Wei

Code Management Platform:

- Basic Requirement：
  - Repository Management
    - Supporting creating and deleting repositories, uploading and downloading files;
    - Displaying basic information of repository, supporting viewing markdown files and code files with keywords highlighting;
    - Authority management, including visiting (private/public) and editing (creators and participants) each repository.
  - Version Management
    - Supporting code branches;
    - Version management when editing files; Support version rollback;
  - User Management
    - Supporting sign-up and sign-in;
    - Personal homepage displaying users' information;
    - Open source community development support, including issues, pull request, watch/star/fork etc.

Code Management Platform:

- Advanced Requirement:
  - Frontend:
    - Embellishing web pages;
    - Responsive design, adapting web pages with other type of devices such as phone;
    - Advanced components such as data visualization chart and timeline components;
  - User Community:
    - Third-party login; user following and activities displaying;
    - Developers' forum;
    - Private chat and news feed;
    - Repository recommendation and fuzzy search;
  - System Architecture:
    - High concurrency and load balancing;
    - System encryption and security policy;
    - Simplified website deployment process, etc.
  - Advanced Repository Features:
    - CI/CD with automated scripts;
    - Automated user pages deployment;
    - More plugins for the repositories;
  - Interaction with Local Code:
    - Git support or build local command line tools from scratch;
    - Specialized local graphical client for code management.
- Stakeholders: Yitong Wang, Zengyi Wang

SUSTech War Chess:

SUSTech War Chess:

- Basic Requirement
  - Complete game structure: game main interface, in-game interface, etc.
  - At least implement two mode of game:
    - Single Player Mode (Human-Machine Battle): The AI of human-machine battle needs to have a certain behavior logic and cannot be set to simple random. Design at least two levels.
    - Local Multiplayer Mode: Players can alternate battles on the same device.
  - Design at least three pawn classes, each containing at least three attributes (e.g. Attack, Defense, Health, etc.). Different classes need to have a clear distinction in addition to the basic attributes. Examples: archers, warriors, magicians, priests, knights, etc.
  - Simple terrain features, including at least:
    - There is a distinction between movable and immovable terrain.
    - The map can be customized and read as a file. There is no limit to the storage format.
  - Simple rpg elements, including at least:
    - Pieces can be leveled up, carrying equipment/learning skills/using items (at least one of them).
    - Design at least 2 different abilities/powers/equipment.

SUSTech War Chess:

- Basic Requirement
  - Basic UI interactions, at least included:
    - Displays a range that can be moved when the pieces are selected。
    - The difference between the selection of the pieces and the display effect after the action is completed。
    - You can choose which object to fight when there are multiple enemies around.
  - Complete game logic, at least included:
    - When the two sides are fighting, a reasonable and correct attack sequence and damage calculation are required to show a simple process.
    - In a specific situation or when one side of the pieces is completely destroyed, correctly judge the game victory or defeat.
    - Normally, pieces can only be moved once per turn, attacks/skills once or use items once.
    - Normal round interaction between the enemy and our army. Requiring all of our pieces to end their turn after action or manually end the round early. (Player round/piece turn, both modes are available).

SUSTech War Chess:

- Advanced Requirement (the ones marked * are opitonal and not capped when scoring):
  - Implement the function of any number of retracting a false move in a chess game.
  - Designing the game memory and reading function interface in single-player mode requires at least three or more fixed gears. Saves can be overwritten and destroyed, and the game's save time and approximate progress are displayed on the file reading interface.
  - Designed as a 3D perspective battle flag game, it can realize the movement, contraction and rotation of the perspective.
  - Design richer terrain elements and make them impact battles. Such as different terrain heights, occupiable neutral areas, etc. *
  - Richer in-game RPG elements such as characters, skills, equipment, classes, attributes, etc. *
  - Rich in extra-combat elements such as town exploration, blacksmith shop, tavern, game map, etc. *
  - Design a networked battle mode to explore more interactive gameplay between players, and even design a separate game server. *
  - More beautiful game interface, character model design, and smooth UI interaction. Originality is not required here. *
  - Add more unique combat mechanics and gameplay, such as: multiple game factions, class matching, character combination skills, etc. The gameplay and mechanics of the battle flag game on the market are very rich and interesting, and students can learn from and realize it, and even achieve a new way of playing. (If you can design a complete and reasonable battle flag game play yourself, you can ignore some of the requirements in the basic requirement.)
  - Contains positive themes: such as revolutionary historical themes, etc.
- Stakeholders: Hengchen Yuan, Chi Xu

SUSTech Shooting Game:

SUSTech Shooting Game:

- Basic Requirement：
  - Complete game structure: main game interface, game progress interface, settlement interface, etc.
  - Various game elements: HP, weapons, bullets, money (buying equipment), etc.
  - Game map: certain game map scenes, including at least two different maps.
  - Single-player mode (PVE mode): players can move, kill the enemy according to the requirements and achieve the goal of victory. the scripts of the main roles need to be designed by yourself (you need to write a report to explain the technical implementation), you can download the existing models.
  - Enemy: can move autonomously, can launch effective attacks on players, etc.
  - Contains settings that players can adjust.

SUSTech Gamer

- Advanced Requirement:
  - Beautiful game UI, vivid appearance and action modeling.
  - Switch between first person and third person.
  - Support small map to display the status of each unit in real time.
  - Support online mode, not limited to PVP or PVE mode.
  - Support a variety of game modes (such as team competition mode, blasting mode, challenge mode, etc.).
  - Contains positive themes: such as revolutionary historical themes, etc.
- Stakeholders: Linkai Peng, Runzhe Jiang

Tips for your projects

- Frequently contact your stakeholders. They manage the requirements and have written detailed descriptions.
- 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.
- 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 a rigorous manner.

QUESTIONS?