Course Syllabus
VE280 Programming and Elementary Data Structures
Summer 2020

Course Description:

This course is an introduction to programming and provides a foundation for data structures. The emphasis of this course will be techniques and principles to quickly write correct programs. This course is not simply a course about programming. Rather, it focuses on concepts and methods underpinning the C++ language. Specific topics of this course include: basics of Linux, procedure abstraction, abstract data type, inheritance, dynamic memory management, template, polymorphism, linked list, stack, and queue.

Instructor:

Paul Weng
Email: paul.weng@sjtu.edu.cn
Phone: 3420-6765 (ext. 4061)
Office: Room 406, JI Building

Teaching Assistants:

1. Chen, Yaxin
   Email: leepace666666@sjtu.edu.cn

2. Fan, Yiqing
   Email: fyq990508@sjtu.edu.cn

3. Ma, Ziqiao
   Email: marstin@umich.edu

4. Xie, Jinglei
   Email: xie_jinglei@sjtu.edu.cn

5. Yin, Guoxin
   Email: yinguoxin2017@sjtu.edu.cn

6. Zhu, Zhuoer
   Email: zhuzhuoer@sjtu.edu.cn

Textbook (Recommended but not required):

1. *C++ Primer, 4th Edition*, by Stanley Lippman, Josee Lajoie, and Barbara Moo, Addison Wesley Publishing, 2005.

2. *Absolute C++, 4th Edition*, by Walter Savitch, Addison Wesley Publishing, 2009.

3. *Problem Solving with C++, 8th Edition*, by Walter Savitch, Addison Wesley Publishing, 2011.

Class Webpage:

Log into Canvas at https://umjicanvas.com/courses/1604. Lecture slides, assignments, and grades will be posted on the class webpage. Check also Piazza for discussions.

Course Prerequisites:

Prior programming experience using C, C++, Java, etc. (preferably in VG101).

Grading Policy:

There will be about one graded lab per week, five programming projects, one midterm exam, and one final exam. The grading distribution is:

Participation	5%
Projects	40%
Labs	15%
Midterm Exam	20%
Final Exam	20%

Participation include online etiquette, attendance, answering to polls and quizzes (incorrect answers are not penalized), posting on Piazza.

Any questions about the grading of the labs, projects, or exams must be brought to the attention of your TAs and the instructor **within one week** after the item is returned.

Programming Assignments

1. Programming Environment

We require you to develop your programs on a Linux operating system using compiler g++. We will grade your programs in the Linux environment: they must compile and run correctly on this operating system.

2. Assignment Grading

Each project will be evaluated along three dimensions: behavioral correctness, adherence to course principles, and general coding style. Labs will mainly be evaluated along behaviorial correctness, but depending on the week, other criteria (which would be announced in advance if that were the case) can be taken into account.

A program is "correct" if it behaves as specified in the project handout. Project descriptions in VE 280 should clearly spell out all required behaviors, though sometimes we do make mistakes. If you believe something in the specification is inconsistent or not sufficiently detailed, please let us know.

For any given specification, there are many possible programs that behave correctly. Some of those programs will make use of the principles we discuss in this course, while others will not. To receive full credit, you must write your programs in accordance with these principles.

It is our belief that programs are meant to be read by other programmers, not just executed. Projects that are generally unreadable will not receive full credit. You should choose a good programming style and follow this style consistently throughout your code.

Finally, an important skill the projects teach is how to show that your code exhibits exactly the behaviors required by the project specification. So, while we will usually give you a few simple test cases and their results to try out, we will not give you all of the test cases we plan to use to evaluate correctness. Instead, you are required to come up with as many test cases as you can.

3. Due Date

Each programming assignment will be given a due date. Your work must be turned in by 11:59 pm on the due date to be accepted for full credit.

However, we still allow you to submit your homework within 3 days after the due date, but there is a penalty for late submission. The late submission penalty is listed in the following table. For example, if you submit your work late by 23 hours and 40 minutes, your grade will be scaled by 80%. If you submit your work late by 24 hours and 10 minutes, your grade will be scaled by 60%. No work will be accepted if it is more than 3 days late—you will receive a zero.

h Hours Late	Scaling Factor
$0 < h \leq 24$ hours	80 %
$24 < h \leq 48$ hours	60 %
$48 < h \leq 72$ hours	40 %

In very occasional cases, we accept a deadline extension request. Deadline extension requests will only be considered if you contact the course instructor (not TAs!) in person. Such requests will normally only be considered if they are made before the assignment is due, and will only be granted for documented medical or personal emergencies that could not have been anticipated. If you can't see the instructor in advance due to the emergency, then see him/her as soon as you possibly can. In all cases, you will be required to substantiate any extension request with written proof of the emergency. Extensions are not granted for reasons such as accidental erasure/loss of files and outside conflicting commitments. In order to prevent accidental erasure/loss, make sure that you make copies of your code frequently.

Exam

The exams can be organized online or on campus depending if the Minhang campus reopened to students or not during the semester. No electronic devices are allowed in the exams.

You are expected to take both exams at the scheduled times. If you miss an exam, and a medical or personal emergency is not involved, you will receive a zero for that exam. If you anticipate an exam in another course, you must notify the instructor at least one week before the exam date.

Academic Integrity:

1. All students are expected to attend all of the classes.

2. All programming assignments must be done by yourself. You may discuss the project in oral with other students. However, you may not read/copy others' solution. In all cases in which we have reason to believe that cheating has occurred, we will submit relevant materials to the Honor Council for

evaluation.

3. Exams will be given under the JI's Honor Code and will require individual efforts.

Teaching Schedule (Subject to Change)

Lecture	Teaching Activities (Topics and Exams)
1	Introduction; Linux;
2	Linux;
3	Linux; Developing and Compiling Programs on Linux;
4	Review of C++ Basics;
5	Review of C++ Basics; Procedural Abstractions;
6	Recursion; Function Pointers;
7	Function Call Mechanism; Enum Types; Program Arguments;
8	I/O Streams; Testing;
9	Testing; Exceptions;
10	Exceptions; Abstract Data Types (ADT);
11	Abstract Data Types (ADT);
12	ADT Efficiency;
13	Midterm Exam Review;
14	Subtypes and Inheritance;
	Midterm Exam;
15	Virtual Function and Interface;
16	Invariant; Dynamic Memory Allocation;
17	Destructor; Deep Copy;
18	Dynamic Resizing; Linked Lists;
19	Linked Lists;
20	Templates; Container of Pointers;
21	Polymorphic Containers;
22	Operator Overloading; Stacks;
23	Queues;
24	More on Maintenance Methods;
25	Functors;

26	Standard Template Library (STL);
27	Final Review;
28	STL Associative Container;
	Final Exam;