YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CPSC 427a: Object-Oriented Programming *Handout #1*
Professor M. J. Fischer *September 2, 2010*

Syllabus (Fall 2010)

1 Official Yale course listing

CPSC 427 01 (13092) /
CPSC 527 01 (10866)
Object-Oriented Programming
Michael Fischer
TTh 1.00–2.15 AKW 200
Object-oriented programming as a means to efficient, reliable, modular, reusable code. Use of classes, derivation, templates, name-hiding, exceptions, polymorphic functions, and other features of C++. *After* CPSC 223b.

Fall 2010
No regular final examination

2 Course Description

2.1 Nature and Purpose of the Course

Programming can be viewed in two ways: It is the activity of instructing the computer in the solution of a particular problem, or it is the activity of constructing a piece of software with desired properties. The former view focuses on the end product – the results of the computation, and is what is typically taught in beginning programming courses. The latter view focuses on all aspects of the software – its size, speed, complexity, reliability, verifiability, security, ease of repurposing, and ease of deployment, as well as on the results it produces.

This course is concerned with software construction. It exposes students to modern object-oriented programming techniques that have proved successful in the development of large complex software systems by many programmers. It teaches object-oriented design in C++ and explores techniques for building efficient and robust systems.

2.2 Main Topics to be Covered

Objects and classes in C++. Construction and destruction. Types, casts and conversions. Storage management. Achieving isolation through name-hiding. Derivation, inheritance, and abstract classes. Polymorphism, virtual functions, and multiple inheritance. Templates. Exceptions. Use of class libraries such as STL, GTK+ / gtkmm, and I/O streams. Design paradigms. Programming for reliability. Programming for efficiency. Performance measurement and debugging.

3 Course materials

Required textbooks:

- A. Fischer, *Exploring C++*, manuscript, 2009. Available free online.
- Herbert Schildt, *C++: The Complete Reference, 4th edition*, McGraw-Hill, 2002. Paperback ISBN: 0072226803 / 9780072226805. eBook ISBN 0071502394 / 9780071502399.

Other materials will be posted on the course web site from time to time during the term.

Website: I maintain a course website at http://zoo.cs.yale.edu/classes/cs427/2010a/index.html. You should bookmark it in your browser and visit it often. It will grow as the term progresses and will contain announcements, handouts, homework assignments, programming hints, and links to documents in the course directory and elsewhere on the web. *Access may be restricted to machines on the Yale network.* If so, for off-campus use, you will need to follow the instructions for accessing Yale resources from home and while traveling.

4 Course Mechanics

Prerequisite: The prerequisite for this course is CPSC 223 (Data Structures) or equivalent. The ability to write significant programs in the C programming language is required. The course also assumes a familiarity with basic computer science concepts such as are covered in CPSC 201 and CPSC 202.

Requirements: Course requirements include weekly programming assignments and/or written problem sets ($\sim$35%), two hour exams ($\sim$20% each), and a significant programming project ($\sim$25%). The approximate weights of each in determining the course grade are subject to change depending on the number and difficulty of the assignments actually given. There is no final exam. Graduate students taking the course will be expected to perform at a higher level than undergraduates and may be required to do additional work.

Assignments and other announcements: Written problem sets and programming assignments will posted on the handouts page of the course web site from time to time during the course. Other course announcements will be posted on the course home page. It is your responsibility to check these pages frequently.

Email: I am always available for email consultation at fischer-michael@cs.yale.edu. I can't always promise to respond right away, but I can often be reached by email when I am not in the office. Email is also the preferred way to arrange an appointment with me.

5 Policies

Late Policy: Late work will be accepted at the discretion of the instructor and/or TA and will generally be subject to a penalty unless accompanied by a Dean's excuse. Work will not be accepted after graded papers have been returned or solutions released. However, alternative means for making up missed work may be arranged on an individual basis with a Dean's excuse.

Please contact the instructor or TA as soon as you find out that you are unable to submit work on time or to attend a scheduled exam so that suitable makeup arrangements can be made.

Policy on Working Together: Work turned in under your name must be your own work. Plagiarism is unethical and will not be tolerated. You may neither copy from others nor permit your own work to be copied. If you have benefited substantially from hints or solutions received from fellow students or from outside sources, then you must explicitly acknowledge your sources in your work. Using outside resources in solving a problem is acceptable as long as due credit is given, but plagiarism is not.

You may of course discuss the lectures and readings with your classmates in order to improve your understanding of the subject matter. However, all written work must be your own.

Under no circumstances should your work be in the possession of another student. It is important that you keep your files protected so that others cannot read them and that you carefully guard your password. Likewise, do not leave printed work in public areas such as the Zoo or in accessible wastebaskets. If you think your password may have been compromised, you should change it immediately.

You are also always free (and encouraged) to come in and ask the TA or instructor for help about anything concerning the course. Please talk to the instructor if you have any questions about this policy.

Policy on Computer Problems: The Yale College policy on "Use of Computers and Postponement of Work" in the Yale College Programs of Study applies to this course. It is reproduced below.

> "Problems that may arise from the use of computers, software, and printers normally are not considered legitimate reasons for the postponement of work. A student who uses computers is responsible for operating them properly and completing work on time. (It is expected that a student will exercise reasonable prudence to safeguard materials, including saving data on removable disks at frequent intervals and making duplicate copies of work files.) Any computer work should be completed well in advance of the deadline in order to avoid last-minute technical problems as well as delays caused by heavy demand on shared computer resources in Yale College."

Particularly relevant for this course are the cautions against leaving a programming assignment to the last minute when machines might be busy, printers broken, and so forth, and about safeguarding your data.

6 Computing Facilities

The Zoo: This course will be supported by the Computer Science Department's educational computing facility, otherwise known as the Zoo. This facility contains Pentium-based PC's running Linux. You will need to learn how to use these machines if you don't already know how in order to read email, browse course-related web pages, edit and compile C++ programs, and access the course directory. Look at

```
http://zoo.cs.yale.edu/help/
```

for information on getting started if you are new to the Zoo.

The Zoo machines support remote access via the SSL and VNC protocols. These enable you to do your work remotely when it is inconvenient to go in person to the Zoo. Instructions on how to configure your machine for remote access will be posted to the course web site.

While it is possible to do your course work locally on your own adequately-configured machine, I do not recommend it because of problems that arise in trying to work in a different (and sometimes incompatible) hardware and software environment. In any case, regardless of where the code is developed, your assignments must be submitted from your Zoo course account, and they will be graded according to how well they work on the Zoo.

Your course account: You should request a course account for this course *even if you already have a Zoo account*, for otherwise you will be unable to submit work electronically. To obtain your account, go to

```
http://zoo.cs.yale.edu/help/accessing-zoo.shtml
```

and follow the instructions there.

Course directory: The shared course directory, `/c/cs427`, is located on the Zoo server. You can access it from your Zoo course account. It will contain any software needed for this course and miscellaneous documentation and files. It will also contain software to allow you to submit assignments electronically. Public files there can be accessed via the web as well as from a Zoo node. Your class account home directories will also be located there.