

CSC 4410  
Object Oriented Programming  
Syllabus  
Fall 2009-10

## Table of Contents

- [Number and Title of Course](#)
- [Catalog Description of Course](#)
- [Instructor](#)
- [Course Page on Blackboard](#)
- [Required Textbook](#)
- [General Objectives](#)
- [Specific Objectives](#)
- [Major Topics](#)
- [Instructional Methods and Techniques](#)
- [Assignments for Course](#)
- [Evaluation](#)
- [Course Outline](#)
- [Ground Rules for Programming Assignments](#)

### Number and Title of Course:

CSC 441 Object Oriented Programming

## Catalog Description of Course:

Object-oriented programming techniques, encapsulation, constructors, destructors, overloading, single and multiple inheritance, polymorphism, composition, templates, iterators, member function access, data hiding, abstract classes, exception handling, projects.

## Course Prerequisite:

This is an advanced programming class. Students should have already had at least one year of a college level programming course. Students who have completed a one-year sequence in a structured programming language such as C or Pascal C (i.e. CSC 171/172) will be adequately prepared for the class. Other students should discuss their background with the instructor.

Instructor: Dr. R. Michael Canjar

Office:	Engineering 323
Office Hours:	Monday: 11:00 AM – 1:30 PM, Tuesday: 11:00-11:30 AM, 2:30-3:00 PM Wednesday: 11:00 – 1 PM, Thursday: 11:00-11:30 AM, 2:30-3:00 PM & by Appointment If I am not in, please check the computer lab across the hall.(E372).
Phone:	(313)-993-1209
E-Mail:	<a href="mailto:CANJARRM@udmercy.edu">CANJARRM (canjarm@udmercy.edu)</a>

E-mail is the most efficient way to reach me. When you send me an E-mail, please indicate "Question from 4410" in the subject line. I give priority to responding to E-mails that have questions from students in my classes.

I would recommend that you use your **students.udmercy.edu** account for e-mails to me. I have "white-listed" that domain in my Spam blocker, so that it passes through all e-mails sent from that address.

## Course Pages

We will use the University's Blackboard system for course materials. If you have not already done so, please login to Knowledge Course server (<http://knowledge.udmercy.edu> ) and register your e-mail address. This way you will receive any e-mail communications which are sent to the entire class.

Your **user name** has probably **changed** since last year. I believe that they are now using your e-mail user name. Your password may be your "T-number." If you have not already done so, update your password to something more memorable.

Also note that your e-mail address in Blackboard has been reset to your UDM account. It is recommended that you use that account for all school business.

When you log onto Knowledge, you should be able to see the page for our course. The Course Page will contain several Content Areas, which you should be able to get to from the buttons in the left frame:

**Announcements:** will be posted in this area.

**Course Documents** containing class handouts

**Projects:** I will post the statements for your various projects here. **You will submit the final version of your project through this tool.**

**Zip Files :** For each example or project I will distribute source code, documentation, and data file, all packaged together in a Zip File. Those Files will be stored in this folder

**PDF Source Files** - I will also format the source code examples in a printer-friendly format to print, and store the output in PDF files in this folder

**My Grades:** will bring up the grade-book.

You may send e-mail from the page blackboard. However, I would also encourage you to make a note of my e-mail address, in case blackboard ever goes down.

## Required Text:

Ira Pohl; Object Oriented Programming Using C++, 2nd Edition.  
There will also be supplementary lecture notes and handouts.

<http://www.aw-bc.com/catalog/academic/product/0,1144,0201895501,00.html>

<http://btobsearch.barnesandnoble.com/booksearch/isbninquiry.asp?userid=2VLYC1S152&btob=Y&isbn=0201895501&TXT=Y&itm=11>

## Optional:

A Third-Part book on using Visual C++, especially those whose prior experience with Visual C++ is limited.

## Software:

The in-class examples and projects were originally developed for Microsoft Visual C++. You may work with either Visual C++.NET or with version 6.0 You should be able to obtain a copy of this from Briggs 16A.

<http://it.udmercy.edu/audiovisual/index.htm>

## General Objectives:

1. To elaborate on the concepts of Object Oriented Programming, emphasizing the role in Software Construction.
2. To expand students programming experience and abilities , focusing on the development of intermediate level programs.
3. To introduce the C++ language to students whose initial background was in other languages.

## Specific Objectives:

Upon completion of the course, students will

1. Understand the **C-language kernel** of C++.
2. Understand and be able to employ **data encapsulation** and **abstraction** and be able to create their own **Data Types**, possessing the same robustness as Native Types.
3. Understand the use of classes and objects in Software Development and be able to use them in a **team project**.
4. Understand the application of OOP to **Containers and Iterators** and be able to write and employ container classes in their programs.
5. Understand Polymorphism, both parametric (templates) and inherited and be able to use **class libraries**.

## Major Topics:

1. Introduction to C/C++ Programming. The C Kernel of C++. Stream based Input-Output.
2. Pointers and Arrays in C and the semantics of C-pointer operations.
3. Introduction to Classes. Operator Overloading. Class Example: Fraction
4. Introduction to Data Abstraction. Class Examples Vect. and Stack
5. Constructors, Destructors, and Copy-Semantics.
6. Using OOP in large software projects. Team Project: Fraction Calculator.
7. Containers and Iterators: Enhancing the class Vect
8. Inheritance and the Use of Virtual Functions.
9. Templates.

## Instruction Method and Techniques:

1. There will be two lectures of 75 minutes each week.
2. Software and Example programs will be demonstrated in class. Examples will be made available on class handouts and may be downloaded from the Internet.
3. Students will have 4-5 individual lab assignments to be done outside of class.
4. Extended Office Hours will be held in the Lab when necessary to provide additional help in 3.
5. There will be one Team Project, done outside of class over several weeks..

## Assignments for the Course:

1. There will be readings from the Text supplemented by class handouts.
2. There will be 4-5 individual programming assignments, some of which may be in several parts. There will be a major Team Project.
3. There will be an in-class Midterm and Final Exam. The current plan is for this exam to be open-book open notes, closed computer, and closed friends.

## Course Evaluation

Individual Assignments	30%
Team Project	20%
MidTerm Exam	20%
Final Exam	30%

## Attendance/Participation :

Students are expected to attend class on a regular basis and participate in the discussions. They are responsible for all the material presented therein. Formal attendance records will not be maintained; however attendance is highly correlated with performance on the projects and the exams.

The instructor will attempt to make reasonable accommodations for students who miss a class due to illness, death in the family, or other legitimate reasons. However students who are forced to miss several classes will have difficulty completing the course in a satisfactory manner.

## Academic Integrity

Students are expected to conform to a high standard of honesty and integrity in this course. Copying the work of someone else and other forms of cheating are strictly prohibited. Permitting or tolerating such behavior is also prohibited. The minimum penalty for any offense is a 0 on that assignment. The culprits may be subject to additional sanctions, up to and including expulsion from school for serious offenses, as prescribed by the University Catalog and the Engineering Science Student Handbook.

For the programming assignments, you may freely discuss the concepts and ideas of the projects. You may also get any help regarding use of the hardware/software. However any code written should be your own; you should not copy all or part of the code of someone else. Generally speaking, access to another's computer files in either computer or printed form can constitute a *prima facie* case of cheating. Do not "loan" your files to anyone in any form for any reason.

## Tentative Course Outline

Project 1:	Introducing C++ :GCD Program	Thurs. 9-17
Project 2:	Arrays, Pointers, and Classes	
	Part A: Arrays	Thurs. 10-1
	Part B: Pointer	Thurs. 10-8
	Part C: Intro to Classes	Thurs 10-15
MidTerm Exam		Thurs 10-22
Project 3	Team Project, Fraction Calculator	Thurs. 11-12
Project 4:	Containers, Iterators	Thurs. 12-1
Project 5	The Standard Template Library; Inheritance	Thurs. 12-10
Final Exam		Wed. 12-16 : <b>2:00 PM</b>
Grades Dues		Mon. 12-21

### Inclement Weather /Class Cancellation Policy :

Unless, you are notified otherwise: in the event that a cancelled by the University for whatever reason, then any exam or assignment that was due that due will be due the next class, and any other exams/projects will be similarly rescheduled. I will attempt to post an announcement on Blackboard, and attempt to contact you by e-mail to confirm this

### Make Up Policy :

Make Up exams will only be given to students who miss an exam for legitimate reason (as defined above under "Attendance") and who notify the instructor in advance. At the instructor's options, the weights of the other assignments will be adjusted *in lieu* of a make-up.

## Other Important Dates

You can also read the tentative academic calendar online. Look at AY 2009-2010 which now is the first column of

<http://www.udmercy.edu/registrar/academic-calendar/1012calendar3yr.pdf>.

Some important dates to note are

September 14 Last Day to Add or Delete a Class

September 24 Celebrate Spirit <http://www.udmercy.edu/celebrate-spirit/>

September 26 Last Day to Drop a Course Without a "W"\*

October 27 Mid-term Grades Due From Faculty

November 2 Advising For Term II Winter and Summer 2009-2010 Begins

November 9 Registration for Term II and Summer 2009-2010 Begins

November 23 Last Day to Withdraw From Class

November 26 - 29 Thanksgiving Recess (UNIVERSITY CLOSED)

December 14 - 19 Final Exam Week

### Tuition Refund Policy

This is primarily of concern to part-time students. You can find the refund policy <http://www.udmercy.edu/weblink/registration/faqs/index.htm> Note that it falls off very fast. No refund is available after 2 weeks.

**This is a change from previous years.**



## Ground Rules For Programming Assignments

1. **Source Language and Compiler** : All projects are to be done in C++, using aspects of the language that are discussed in class. You may use either Microsoft Visual Studio 6.0 or Visual C++.NET. Both are available in University Labs. Version 6.0 works best with the pre-standard headers (such *iostream.h* , *fstream.h*, etc) and .NET works best with the modern standard ( *iostream*, *fstream* ).

I will give you a header file *oop-rmc1.h* which you should include with your source code. It will detect which version you are using and include the appropriate header files. It will also include the proper *namespace* statement if that is appropriate.

2. **Hand-in:** Projects will be submitted through Blackboard, through the Projects page (and NOT trough) For all projects, students should hand in ZIP file that contains a copy of their source code and output from a sample run of the program, **using any test data that have been provided**. In addition, the Output should always be date-stamped by the computer, indicating the date on which the output was produced. (Functions to assist in producing this output will be provided in the header file *oop-rmc1.h* ) .
3. **Grading:** Programs will be graded for scope, correctness, style, documentation, attractiveness of output, and timeliness. A program that "works" will not receive full credit unless it is well-written, properly documented, and uses the appropriate style. Please refer to the comments below regarding style. .
4. **Late projects** will not receive full credit. Unless otherwise specified, projects turned in late may lose as much as 20% and projects may not be accepted at all after that week. Students who are unable to complete a project within the allotted time should discuss the situation with the instructor as soon as practical.

Projects turned in on time will usually be returned at the next class meeting. No such guarantee is available for late projects.

5. **Style:** Style is an important component of programming. Programs should employ meaningful identifiers and informative comments. Capitalization may be used provided it is in a consistent, coherent manner. (You may also choose to follow the text and use exclusively lower case identifiers with underscore characters to enhance readability.)

Unlike programs written in more self-documenting language like Pascal, CL programs tend to be cryptic and difficult to understand. Students should make judicious use of informative comments in their projects. Note that

comments that merely repeat the meaning of an instruction are not informative. Comments should explain the purpose and functioning of code, not merely repeat it. (Some of the class handouts and text examples will have comments targeted at students learning the language for the first time. These comments would, of course, be inappropriate for other programs and for the homework assignments.)

Each program should certainly include a comment near the beginning giving the name of the project, the author, and his/her section number. Some explanation of each function is usually

6. **Academic Integrity** Students are expected to conform to a high standard of honesty and integrity in this course. Copying the work of someone else and other forms of cheating are strictly prohibited. Permitting or tolerating such behavior is also prohibited. The minimum penalty for any offense is a 0 on that assignment. The culprits may be subject to additional sanctions, up to and including expulsion from school for serious offenses, as prescribed by the University Catalog and the Engineering Science Student Handbook.

For the programming assignments, you may freely discuss the concepts and ideas of the projects. You may also get any help regarding use of the hardware/software. However any code written should be your own; you should not copy all or part of the code of someone else. Generally speaking, access to another's computer files in either computer or printed form can constitute a prima facie case of cheating. Do not "loan" your files to anyone in any form for any reason.