# CS 244
# Data Structures and
# Analysis of Algorithms

Fall 2006

M-W-F 12:50 – 2:00pm

HOS-123

http://www.cs.moravian.edu/cs244

**CS 244. Data Structures (1u)**
An examination of issues dealing with static and dynamic aggregates of data. Topics covered include logical characteristics of various data organizations, storage structures implementing structured data, design and implementation of algorithms to manipulate such storage structures, and classical applications of data structures. Representative data structures include stacks, queues, ordered trees, binary trees, and graphs. Both contiguous and linked storage implementations are considered and performance issues discussed. Prerequisite: **CS 121** and **Mathematics 170 (or 106-166)**.

## Instructor

```
Stephen Corbesero
PHAC 213, 610-625-7786
corbesero@cs.moravian.edu
Office Hours: MW 10:30--11:30
              R 1:30-2:30, and by appt.
```

## Goals

- You will improve your object oriented design and implementation skills..

- You will become familiar with classic data structures and their manipulation algorithms.

- You will learn to develop and test larger programming projects.

- You will become more familiar with the C++ programming language.

## Text

The primary text for the course is Problem Solving with C++: Walls and Mirrors, 5/e by Carrano . If you are new to C++, you would be wise to find a book to act as a language reference such as C++: How to Program by Deitel and Deitel. The additional required programming related texts are the C++ Pocket Reference by Loudon and

Gnu Emacs Pocket Reference by Cameron, both of which are from O'Reilly. Two additional pocket references from O'Reilly that may be helpful but are optional are the STL Pocket Reference and C Pocket Reference.

There is one additional required textbook: Guide to LaTeX, 4/e by Kopka and Daly.

## Prerequisites

Students are expected to have a strong background in fundamental structured and object-oriented programming and basic data structures as covered in CS 120-121.

## Assignments, Programs, and Tests

Homework and programming assignments will mostly consist of ANSI C/C++ programs. This is a programming intensive course. Tests will consist of three hour exams and a final.

### Tests

No makeup exams will be given. Students missing one or more tests, in a properly excusable fashion, will be graded based on the available scores as the total score. The hour exams are open book and open notes, unless explicitly stated otherwise.

### Homework

Each homework (non-program) will be graded out of a possible 100 points. Late homework will be penalized with the same schedule as late programs (see below).

### Programs

- Each program will be graded out of 100 points, but will be weighted to reflect its relative complexity. Programs will be graded (roughly) on correctness (~70%), style (~20%), and documentation (~10%). If an assignment is one *class-day* late, it will be penalized 10%. If it is no more than *class-week* late, it will be penalized no more than 50%. After one class-week, it will be worth little credit, if any.

- Unless explicitly stated otherwise, programs are due electronically on midnight on the due date.

- Failing to turn in correct programming assignments in a timely fashion is hazardous to your grade, directly and indirectly. If you start missing assignments, I will notify your academic advisor.

Keep in mind the following items about submitting programs.

1. Program source files must contain the program header in a comment section as well as a code section. This header consists of the program title, number, author, course, and due date. Other style requirements will be made known as the semester progresses.

2. Programs are collected electronically. Pay close attention to directory and files names, including case.

   - You must execute `touch DONE` in the proper directory, *cs244.064/x*.
   - A collect program will periodically look for these *DONE* files.
     - If the *DONE* file is found, the contents of the directory will be copied, a *.collected* file will be deposited, and a congratulatory email message will be sent.
     - If no *DONE* file is found, an email will be sent pointing out that no collection was done.
   - If you are asked to resubmit a program you must delete (`rm`) the *.collected* file and `touch DONE` again. You must also physically resubmit the grading worksheet.

## Computer Resources

The primary computer resources will be the Unix-based Sun Solaris 2.x workstations on MoCoSIN. The software will be the `gcc/g++` compiler and related utilities from the Free Software Foundation. All programs must compile and run on this platform.

You are expected to comply with all MoCoSIN, CIT and campus policies with respect to use of the computer resources. This includes, but is not limited to, such policies as not locking workstations, not using an account other than your own, etc.

## Grading

### Weighting

| | |
|---|---|
| Programs | 33 |
| Homework and Quizzes | 12 |
| Hour Exams | 30 |
| Final Exam | 25 |
| **Total** | 100 % |

### Policies

- Incomplete grades will **not** be assigned for failure to do the work as required during the semester.
- Attendance is very important, and pop quizzes, which would count in the homework category, may spontaneously occur. You are responsible for everything discussed in class.

## Important Dates

| | | |
|---|---|---|
| **Aug 28** | M | First day of classes |
| **Sep 4** | M | Labor day–no classes |
| **Sep 5** | T | Last Day to Add/Drop |
| **Sep 22** | F | **Hour Exam I** |
| **Oct 6** | F | MidTerm |
| **Oct 7–10** | S–T | Fall Recess |
| **Oct 20 or 27** | F | **Hour Exam II** |
| **Oct 27** | F | **CCSCE-06** |
| **Nov 10** | F | Last Day to Withdraw with a W |
| **Nov 10** | F | **CCSCSE-06** |
| **Nov 20** | M | **Hour Exam III** |
| **Nov 22–26** | W–U | Thanksgiving Break |
| **Dec 11** | M | Last Day of Classes |
| **Dec 12**, | T | Reading Day |
| **Dec 13–16** | W–S | Final Examinations |
| **Dec 17** | U | Reading Day |
| **Dec 18–19** | M–T | Final Examinations |

## General

- Keep backups of all assignments, especially during program development.
- Special circumstances, will, of course, be considered on an individual basis. Please see us as soon as possible if any such circumstances arise.
- All work, unless explicitly stated in the problem definition, is to be an individual effort. Students are encouraged to discuss approaches so long as the final submission has a single, clearly identifiable author. Violations of this will be dealt with as a case of academic dishonesty, see below.

## Academic Dishonesty Policy

Students are encouraged to read and understand the college policy on academic honesty. Violations of this policy will certainly result in reduced (0?) scores on the assignments and may result in a failure of the class. In addition, students are expected to read and comply with the course specific policy on improper collaboration.

## Terms and Conditions

I consider this syllabus to be a contract between myself as instructor and you as student. Therefore, I will do my best to adhere to the policies herein. However, if the circumstances warrant, there may need to be changes. Such changes will clearly be communicated to the class.