

CIS 542 – Program and Data Structures

Paul Wolfgang

Room 1041 Wachman Hall

wolfgang@joda.cis.temple.edu

<http://www.cis.temple.edu/~wolfgang>

Office Hours:	Monday	9:30 – 11:30
	Wednesday	9:30 – 11:30
	Friday	7:30 – 9:30

OBJECTIVE

The purpose of this course is to learn to write and to reason about programs that involve complex data structures and algorithms. This goal involves:

- Acquiring an in depth knowledge of a programming language. We will use C++, but the principals apply to other programming languages.
- Some software engineering skills in structuring, documenting, and testing programs.
- Some Mathematical/Logical skills to justify why a program does what it is supposed to do.
- Some Mathematical skills to determine the time and space complexity of an algorithm.
- A number of data structures and algorithms.

TEXT and References

- TEXT: Budd. *Data Structures in C++ using the STL*. Addison-Wesley, 1997.
Stroustrup. *The C++ Programming Language*, Third Edition. Addison-Wesley, 1997.
Lippman & Lajoie. *C++ Primer*, Third Edition. Addison-Wesley, 1998.
- References: **C++ programming language**
Koenig & Moo. *Accelerated C++: Practical Programming by Example*. Addison-Wesley, 1997.
Stroustrup. *The Design and Evolution of C++*. Addison-Wesley, 1994.
Schildt. *C++: The Complete Reference*. McGraw-Hill, 1998.
Koenig & Moo. *Ruminations on C++*. Addison-Wesley, 1997.
Cline, Lomow & Girou. *C++ FAQs: frequently asked questions*. Addison-Wesley, 1999
An abbreviated, constantly updated version available at:
<http://marshal-cline.home.att.net/cpp-faq-lite>

Data Structures using C++

Berman, A. Michael. *Data Structures via C++ Objects by Evolution*. Oxford University Press, 1997.

Budd. *Classic Data Structures in C++*. Addison-Wesley, 1994.

Drozke, Adam. *Data Structures and Algorithms in C++*. PWS Publishing Company, 1996.

Headington & Riley. *Data Abstraction and Structures Using C++*. D. C. Heath and company, 1994.

Main & Savitch. *Data Structures and other Objects Using C++*. Addison-Wesley, 1997.

Weiss, Mark Allen. *Algorithms, Data Structures, and Problem Solving with C++*, 2nd Ed. Addison-Wesley, 2000.

Data Structures using other languages

Korsh & Garrett. *Data Structures, Algorithms, and Program Style using C*. PWS-Kent, 1988.

Kruse, Leung, & Tondo. *Data Structures and Program Design in C*. Prentice Hall, 1991.

Horowitz & Sahni. *Fundamentals of Data Structures in Pascal*, Third Edition. Computer Science Press, 1990.

Looms. *Data Management and File Structures*, Second Edition. Prentice-Hall, 1983.

Knuth. *The Art of Computer Programming, 2d -- Vol 1 / Fundamental Algorithms*. Addison-Wesley, Reading MA, 1973.

Knuth. *The Art of Computer Programming, 2d -- Vol 2 / Seminumerical Algorithms*. Addison-Wesley, Reading MA, 1981.

Knuth. *The Art of Computer Programming, Vol 3 / Sorting and Searching*. Addison-Wesley, Reading MA, 1973.

Schedule of Lectures

Week	Date	Topic	Text Reference	Assignment Topic	Assignment Due
1	28 Jan	Introduction, Overview of C++	1, D		
2	4 Feb	Classes and Object-Oriented Design	2	1. Simple War Game	
3	11 Feb	Input/Output, Array, Vectors, and Strings	7, 8	2. Pay Checks	
4	18 Feb	Algorithms, Recursion, Analyzing Executing Time, Proof of Algorithm Correctness	3, 4, 5		
5	25 Feb	Lists	9	3. 'th' to 's'	1
6	4 Mar	Introduction to the Standard [Template] Library	6	4. Regression Coefficients	2
	11 Mar	SPRING BREAK			
7	18 Mar	Mid-Term Examination			
8	25 Mar	Stacks, Queues, Deques, and Priority Queues	10, 11	5. tbd	
9	1 Apr	Polymorphism and Simulation	15	6. Grocery Store Simulation	3
10	8 Apr	Trees and Binary Search Trees	13, 14	7. tbd	4
11	15 Apr	Red-Black Trees, Hashing Functions, Sets, and Maps	12, 16, 17	8. tbd	5
12	22 Apr	B-Trees			6
13	29 Apr	Graphs	19		7
14	6 May	Open for Semester Review			8
15	13 May	Final Examination			

Grading

There will be 8 assignments, each counting 5 points. There will be two examinations each worth 35 points. Thus, the maximum possible is 110.

The final grade will be determined based upon the total number of points as follows:

≥ 100	A	≥ 85	B	≥ 70	C
≥ 95	A-	≥ 80	B-	≥ 65	C-
≥ 90	B+	≥ 75	C+	< 65	F

Homework Grading

Credit for the assignments is as follows:

On time, correct	5
Late, correct	3
Late, incorrect	1 with the option to up-grade to a 3
Early, incorrect (see note)	2.5 with the option to up-grade to a 5
On time, incorrect (see note)	2 with the option to up-grade to a 4
Re-submission, if originally submitted on time	4

Note: You are strongly encouraged to show progress by submitting whatever you have by the date due. However, you are also expected to test your programs prior to submission. Credit may not be given for submissions that fail to compile due to syntax errors, or that do not demonstrate actual work on the assignment.

To receive early submission credit, the submission must be at least one week early. It also must represent what you believe to be a working program. Programs containing syntax errors or other obvious flaws will not be given credit.

POLICY

Homework assignments demonstrate your ability to apply class material. It is desirable for classmates to work together and help each other. The final product must be your own. The first submission that appears to be a copy of another student's will receive **minus one** as the final grade. Copies from previous year's students or from other external sources will also receive minus one. Furthermore, all of your submissions will be viewed with suspicion.

If more than one assignment is a copy, you will receive a final grade of F for the course.

You may provide advice to fellow students, and are encouraged to do so. However, you should not provide copies of your completed programs to other students neither should you copy programs from other students.

Assignments will not be counted as having been submitted under the following circumstances:

- E-mail subject line is not correct
- Name is missing from any submitted file.

Homework is due on the date stated. E-mail must be received by 8:00pm. E-mail will be read once a week on Mondays at 8:00pm. Assignments will be returned the following Monday.

Homework Instructions

1. Homework is due in class on the date stated. If submitted by e-mail, it must be received by 8:00pm of the date stated. An automatic e-mail reader daemon will run at that time each Monday. Note: the e-mail will only be checked at that time. Homework will be graded and returned the following Monday. I cannot provide any earlier feedback.

2. How to submit homework.

E-Mail assignments to c542100@snowwhite.cis.temple.edu

the subject line **must** be as follows:

<family name><space><given name><space>CIS542<space><assignment type><number>

where <family name> is your family name (known as last name in the US) as shown on your Temple records

Do not include the "<" and ">"!

<given name> is your given name (known as first name in the US) as shown on your Temple records.

<space> is one or more space characters.

<assignment type> is either HW for regular homework or EC for extra credit.

<number> is the assignment number -- do not use leading zeros.

For example, I would submit assignment 4 as

Wolfgang Paul CIS542 HW4

It is very important that you follow these instructions exactly. I am using an automatic mail reader on that will be *extremely stupid*. Anything that does not meet these rules will be discarded.

2. Format for submittal

Each source file must begin with the following three comment lines:

```
// Student Name
```

```
// Assignment number : nnn
```

```
// File name
```

FILES MISSING YOUR NAME WILL NOT BE COUNTED AS HAVING BEEN SUBMITTED.

3. Language and computer to be used

You may use any computer legally available to you to perform your assignments.

Note that while C++ is now standardized, many compilers do not fully conform to the standard, therefore there may be some incompatibilities between what is expected of the assignment and the capabilities of your compiler. Use of the GNU compiler, g++, version 2.95.3 or later is encouraged. Microsoft Visual C++, version 6, will work. Older versions of the GNU, Microsoft, and Borland compilers have certain incompatibilities with the current standard that may make them unacceptable for assignments. If possible, standard's incompatibilities will be accommodated.