EECS 381 Winter 2010 Schedule Version 2

Readings Source: K&R:= Kernighan & Ritchie, S:= Stroustrup, H := Handout on course website (*assigned handouts must be covered in your paper*)
Your paper must discuss each chapter, section, or handout listed; cover the entire chapter or section unless part of it is listed as "skip."
A section assigned as "skim" requires only a brief (e.g. one sentence) mention in your paper.
Topics discussed in lecture will often overflow into the next class period, but reading assignments are still due on the scheduled date.

No. Date **Lecture topics and due dates for readings and projects**

1 Jan 7 Th **Organizational and Introduction.**

2 Jan 12 T **C concepts: prototypes, headers, linkage.**
- K&R 1-4. Much will be familiar but watch for new information, especially in Ch. 4.
- H: *Header File Guidelines for C Programs* (see above)

3 Jan 14 Th **Pointers, Arrays, Function pointers, structures.**
- K&R 5. Read carefully about pointers, arrays, function pointers; skim 5.12 about complex declarations; K&R 6-6.4; skim 6.5-6.9

4 Jan 19 T **Pointers, etc, continued.**

5 Jan 21 Th **I/O, Type safety, memory allocation.**
- Read K&R 7 on I/O and other functions.

6 Jan 26 T **C++ preview, review, filling gaps.**
- Stroustrup: All three prefaces, Ch. 1, skim C++ tour (Ch 2), skim Std Lib tour(Ch 3).
- S 4 Types and Declarations: declaration terminology (4.9.1), scope (4.9.4), initialization (4.9.5), objects and lvalues (4.9.6), 4.10.
- S 5 Pointers, Arrays, Structures: pointers and zero as a pointer value (5.1), const (5.4), references (5.5),
void * (5.6), structs and incomplete (forward) declarations (5.7), 5.8,
- H: *Incomplete Declarations*

Throughout Stroustrup, take time to think about his "advice" sections at the end of each chapter - extremely valuable.
For Chs 1-7 and 9, if you don't recognize something in the advice, I recommend that you read that part of the chapter even if unassigned.

7 Jan 28 Th **C++ review, filling gaps (continued).**
- S 6 Expressions and Statements: Skim the extended example in 6.1, because he refers to it many places later.
Read evaluation order (6.2.2), new and delete(6.2.6), casts and constructor notation(6.2.7, 6.2.8), declaration positions (6.3.1, 6.3.2.1, 6.3.3.1), 6.4, 6.5.
- S 7 Functions: Introduction to 7.2, Overloaded functions (7.4), default arguments (7.5), 7.9.
- S 8 Namespaces and exceptions: Skim S on namespaces, but read
- H: *Using using*.
Read 8.3 on error handling concepts and exceptions.
- S 9 Source Files and Programs: The one-definition rule (9.2.3), program startup and termination (9.4), 9.5.
- H: *C++ Header File Guidelines*

*** **Jan 29 F** **Project 1 Due**

8 Feb 2 T **Classes, objects with dynamic memory contents, Operator Overloading, Templates.**
- S10 Classes: All of chapter. Some review, but study carefully. E.g. 10.2.7.1 &.2 are probably new to you, and 10.4.4.1 is vital,
and 10.4.10 explains many otherwise weird error messages. Skip 10.4.11 on "placement new".
- H: *Static Members*
- S 11 Operator Overloading. All of chapter, though it may be familiar. Don't overlook 11.4, 11.9, 11.11
If this material is unfamiliar, read the Handout *A Summary of Operator Overloading*.
- S13 Templates: Routine use of templates(13.1-13.3).

9 Feb 4 Th **Library Organization & Standard Containers; Strings & Streams**
- S 16. Skip 16.2.1, 16.2.2 for now - we'll come back.
Rest of chapter presents <vector> as an example of how container classes work. Skip 16.3.11. Read 16.4 Advice.
- S 20 Strings. Skim this complete presentation of an extremely important and elaborate class; plan to look up as needed.
- S 21 Streams. Skim through 21.5. Skip 21.6-21.8, read Advice 21.9. If confused, re-read the streams Handouts.
- H: *Notes on Basic C++ Stream I/O*
- H: *Using C++ File Streams*

10 Feb 9 T **Standard Containers.**
• S 17. A survey of the containers. Learn to read this stuff without getting bogged down in the details.
Focus on overviews and examples of use. Notice how he describes a container by presenting its public interface.
Skip 17.5.3. Skim 17.6 about defining a new container - you might want to do it sometime, but not for a while! Read Advice (17.7).
• H: *Fill'er Up: Winners and Losers for Filling an Ordered Container*

11 Feb 11 Th **Algorithms, Iterators, and Function Objects.**
• S 18. There are lots of goodies in the Library. Read the overviews in each section,
focus on the specific things called out in the following list, and skim the rest. Specifically:
Read 18.1, 18.2, 18.3 (skip 18.3.1), 18.4 (skim 18.4.4). Read 18.5 intro, focus on 18.5.1, .2;
read 18.6 intro, focus on 18.6.1 copy, 18.6.2 transform, 18.6.4, .5, 18.6.8.
read 18.7 intro, focus on 18.7.1, .2, 18.9, 18.10, 18.12
• S 19 Iterators & Allocators. Skip 19.2.2, 19.2.6.1, skim 19.2.3, skip 19.3, 19.4, read 19.4.5, .6.
• H: *Why std::binary_search of std::list works, sorta ...*

***** Feb 12 F Project 2 Due**

12 Feb 16 T **Basic Class Design.**
No reading assignment, but bring H: *Basic Class Design* to lecture to mark up

13 Feb 18 Th **Midterm Exam, room TBA**
Notice: *When "no reading assignment" is listed below, attending lecture is* ***absolutely essential****.*

14 Feb 23 T **Simple forms of inheritance and polymorphism: Inheritance & Virtual Functions**
• S 12 Derived Classes. All of the chapter is important.
• S 16.2 Now go back and see one of the ways inheritance can be a bad idea in class design.

15 Feb 25 Th **More on virtual functions**

***** Feb 26 F Project 3 Due**

Mar 2 T *Winter Recess - no classes*

Mar 4 Th *Winter Recess - no classes*

Notice: *When "no reading assignment" is listed below, attending lecture is* ***absolutely essential****.*

16 Mar 9 T **Introduction to OO Design**
Lecture: Project 4 design overview. No reading assignment is due, but bring to lecture:
a hard copy of H: *Introduction to UML*, and H: *Basic OOP Concepts* (BasicOOPConcepts-HO.pdf) or the lecture notes on Basic OOP Concepts.

17 Mar 11 Th **Multiple inheritance and run-time type identification.**
• S 15. Class Hierarchies. Skip 15.6 intro and 15.6.1, but read 15.6.2 carefully.

18 Mar 16 T **Exceptions and memory management, RAII, "smart pointers"**
• S 14. Exception Handling. Skim 14.6.

19 Mar 18 Th **Some Idioms and Design Patterns** (no reading assignment - bring hardcopy of Lecture Notes: IdiomsDesPattsX.pdfs to mark up)

***** Mar 19 F Project 4 Due**

20 Mar 23 T **Some Idioms and Design Patterns** (no reading assignment - bring hardcopy of Lecture Notes: IdiomsDesPattsX.pdfs to mark up)

21 Mar 25 Th **More Idioms and Design Patterns** (no reading assignment - bring hardcopy of Lecture Notes: IdiomsDesPattsX.pdfs to mark up)

22 Mar 30 T **More Idioms and Design Patterns** (no reading assignment - bring hardcopy of Lecture Notes: IdiomsDesPattsX.pdfs to mark up)

23 Apr 1 Th **Designing Software**
• Ch 23. Development and design. Read all of chapter. There is much more that could be said about this topic, but Stroustrups's overview is a good one.

***** Apr 2 F Project 5 Due**

24 Apr 6 T TBA

25 Apr 8 Th TBA

26 Apr 13 T *No class meeting - Kieras at ACM SIGCHI Conference*

27 Apr 15 Th *No class meeting - Kieras at ACM SIGCHI Conference*

28 Apr 20 T ***** Project 6 Due - time and place for submission of hard copy materials to be announced.**

Apr 23 F FINAL EXAM, 4:00 PM - 6:00 PM, room TBA