

## Study Guide to Exam 1

For the exam, you are responsible for the contents of lectures 1–11, the class demos, the concepts used in problem sets 1–3, and the corresponding sections of the textbook. In greater detail, you are responsible for the following chapters and sections of the textbook:

- All of Chapters 1–8 except for section 8.4.
- All of Chapters 10 and 12.
- Chapter 14, section 14.1.4 only.
- Chapter 15, sections 15.1 and 15.2 only.
- Chapter 16, sections 16.1 and 16.2 only.

In addition, the remainder of Chapter 15 presents some detailed code examples that use polymorphism and virtual functions. While we haven't gone through the examples in class, you should read this chapter and make sure you understand how these concepts work in the examples presented there.

Below is an index to the lecture notes. It lists all of the sections, subsections, and slide titles from lectures 1–11.

### 1 About This Course [lecture 01]

- Where to find information
- Course mechanics
- Topics to be Covered
- Course goals - practical
- Course goals - conceptual

### 2 Kinds of Programming [lecture 01]

- Two views of programming
- Problem solving
- Software Construction
- Programming in the large

### 3 C++ Programming Standards [lecture 01]

- Three commandments for this course
- Can is not the same as should!

## 4 C++ Overview [lecture 02]

### 4.1 C++ Goals

- Why did C need a ++?
- C++ was Designed for Modeling
- General properties of C++

### 4.2 Comparison of C and C++

- C++ Extends C
- Some Extensions in C++

### 4.3 Tools

- Tools
- Recommended IDE's

## 5 Example [lecture 02]

### 5.1 Insertion sort

- Generic Insertion Sort

### 5.2 C version

- C version  
ee code demo02

### 5.3 C++ version

- C++ version  
ee code demo02

## 6 Example [lecture 03]

- C++ version  
ee code demo03

### 6.1 Header file

- dataPack.hpp
- class DataPack
- Class elements
- Inline functions
- Visibility
- Constructor
- Constructor
- Destructor
- Destructor

## 6.2 Implementation File

- `dataPack.cpp`
- File I/O

## 6.3 Main Program

- `main.cpp`

## 7 Building Your Code [lecture 03]

- Manual compiling and linking
- Makefile
- Integrated Development Environment (e.g., Eclipse)
- Integrated Development Environment (e.g., Eclipse)
- Integrated Development Environment (e.g., Eclipse)

## 8 Remarks on Laboratory Work [lecture 04]

- Toolset to use for course work
- Working remotely
- 1. Replicate the Zoo environment on your own machine
- 2. Remote login to the Zoo
- 3. Set up a virtual Zoo desktop on your machine
- Homework submission

## 9 Review and Readings [lecture 04]

- A brief course review to date
- How to use the textbook

## 10 A Survival Guide for PS1 [lecture 04]

- Operator extensions
- Adding new methods
- Two kinds of functions
- An ambiguity with operator extensions
- Operator call example: Top-level function
- Operator call example: Member function
- Back to PS1

## 11 More on C++ I/O [lecture 04]

- Opening and closing streams
- Reading data
- Writing data
- Manipulators

- End of file and error handling

## 12 Functions and Methods [lecture 05]

### 12.1 Parameters

- Call by value
- Call by pointer
- Call by reference
- I/O uses reference parameters

### 12.2 Choosing Parameter Types

- How should one choose the parameter type?
- Sending data to a function: call by value
- Sending data to a function: call by reference or pointer
- Receiving data from a function

### 12.3 The Implicit Argument

- The implicit argument
- `this`

## 13 Simple Variables [lecture 05]

- L-values and R-values
- Simple variable declaration
- Simple assignment
- Automatic dereferencing

## 14 Pointers [lecture 05]

- Pointer values
- Pointer creation
- Pointer variables
- Pointer assignment
- Following a pointer
- Pointer example
- Pointer declaration syntax

## 15 References [lecture 05]

- Reference types
- Reference declarators
- Use of named references
- Reference parameters
- Reference return values

- Custom subscripting
- Constant references
- Comparison of reference and pointer

## 16 IO Demos [lecture 07]

- Handling data errors and end of file
- How to write a test program

## 17 Introduction to Classes [lecture 07]

- Classes, visibility, functions, inline

## 18 BarGraph Demo [lecture 07]

- Bar Graph Demo

### 18.1 Specification

- Bar graph sample input and output
- Bar graph data structure
- UML Diagram

### 18.2 graph.hpp

- Notes: graph.hpp

### 18.3 graph.cpp

- Notes: graph.cpp

### 18.4 row.hpp

- Notes: row.hpp

### 18.5 row.cpp

- Notes: row.cpp

### 18.6 rowNest.hpp

- Nested classes: rowNest.hpp

## 19 Storage Management [lecture 08]

- Variables and storage
- Example of a variable
- Properties of variables
- Storage classes
- Assignment and copying
- Static data members
- Static function members
- Five common kinds of failures

## 20 Bells and Whistles [lecture 08]

- Optional parameters
- `const`
- `const` implicit argument
- Operator extensions

## 21 Classes [lecture 08]

- What is a class?

## 22 Derivation [lecture 09]

- Class relationships
- What is derivation?
- Instances
- Some uses of derivation
- Example: Parallelogram
- Example: Rectangle
- Example: Square
- Notes on Square

## 23 Construction, Initialization, and Destruction [lecture 09]

- Structure of an object
- Example of object of a derived class
- Referencing a composed object
- Referencing a base object
- Initializing an object
- Construction rules
- Destruction rules
- Constructor ctors
- Initialization ctors
- Initialization not same as assignment
- Copy constructors

## 24 Polymorphic Derivation [lecture 09]

- Polymorphism and Type Hierarchies
- Polymorphic pointers
- Virtual functions
- Unions and type tags
- Virtual destructors

## 25 Polymorphic Derivation (cont.) [lecture 10]

- Uses of polymorphism
- Multiple representations
- Heterogeneous containers
- Run-time variability
- Pure virtual functions
- Abstract classes

## 26 Name Visibility [lecture 10]

- Private derivation (default)
- Private derivation example
- Public derivation
- Public derivation example
- The `protected` keyword
- Protected derivation
- Privacy summary

## 27 Name Visibility Revisited [lecture 10]

- Surprising example 1
- Surprising example 2: contrast the following
- Surprising example 3

## 28 Name Visibility Revisited [lecture 11]

- Names, Members, and Contexts
- Declaration and reference contexts
- Declaration context example
- Reference context example
- Inside and outside class references
- Examples
- Inherited names
- Inheritance example
- Inaccessible base class

## 29 Interacting Classes and UML [lecture 11]

- What is a Class: Syntax
- Class Relationships
- Class Relationship Between Two Classes
- Class B appears in Definition of Class A
- B as Data Members in A
- B as Data Members in A
- Creation and Deletion
- Example: `BarGraph` Class Interaction