

Reviews for Exam 1

Chapter 1-4

CSc 212 Data Structures, Sec FG

CCNY, Fall 2016

Chapter 1 (Lecture 1)

- Course Objectives
 - WHAT (Topics – ADTs, classes)
 - WHY (Importance – not only for credits)
 - WHERE (Goals – data structure experts)
 - HOW (Lectures, Self-Test Exercises, Assignments, Quizzes and Exams)
- The Phase of Software Development
 - Basic design strategy
 - four steps- S, D, I, T
 - Pre-conditions and post-conditions
 - assert
 - Running time analysis
 - big-O notation

Self-Test Exercises: 3-6, 11-15, 17-20

Chapter 2

A Review of C++ Classes (Lecture 2)

- OOP, ADTs and Classes
- Class Definition, Implementation and Use
- Constructors and Value Semantics

More on Classes (Lecture 3)

- Namespace and Documentation
 - three ways to use namespace; pre- /post-conditions
- Classes and Parameters
 - value, reference, const reference
- Operator Overloading
 - nonmember, member and friend function

Self-Test Exercises: 1, 4 ,513,15,17,21,23, 25,28,31

Chapter 3

- A container class is a class that can hold a collection of items.
- Container classes can be implemented with a C++ class.
- The class is implemented with
 - a header file (containing documentation and the class definition) [bag1.h](#) and
 - an implementation file (containing the implementations of the member functions) [bag1.cxx](#).
- Other details are given in Section 3.1, which you should read, especially the real [bag code](#)

Time Analysis of the Bag Class

- `count` – the number of occurrence
- `erase_one` – remove one from the bag
- `erase` – remove all
- `+=` - append
- `b1+b2` - union
- `insert` – add one item
- `size` – number of items in the bag

The Invariant of a Class

- Two rules for our bag implementation
 - The number of items in the bag is stored in the member variable `used`;
 - For an empty bag, we don't care what is stored in any of `data`; for a non-empty bag, the items are stored in `data[0]` through `data[used-1]`, and we don't care what are stored in the rest of `data`.
- The rules that dictate how the member variables of a (bag) class are used to represent a value (such as a bag of items) are called the invariant of the class

What's the most important, then?

- Concept of Container Classes
 - the bag class is not particularly important
- Other kinds of container classes
 - Other types of bags using **typedef**
 - **sequence** – similar to a bag, both contain a bunch of items. But unlike a bag, the items in a sequence is arranged in order. (**assignment 2**)
- Self-Test Exercises: 1,3, 5,10,11,14,18-24

Chapter 4

- **Pointers**
 - *(asterisk) and &(ampersand) operators
- **Dynamic Variables and new Operator**
 - Dynamic Arrays and Dynamic Objects
 - Stack (local) vs. heap (dynamic) memory
- **Garbage Collection and delete Operator**
- **Parameters revisited**
 - Pointers and Arrays as Parameters

Why Dynamic Classes

- Limitation of our bag class
 - `bag::CAPACITY` constant determines the capacity of every bag
 - wasteful and hard to reuse
- Solution:
 - provide control over size in running time, by
 - pointers and dynamic memory
 - => dynamic arrays
 - => dynamic classes

Dynamic Classes (Ch 4.3–4)

- Pointers Member Variables
- Dynamic Memory Allocation
 - where and how
- Value Semantics (with dynamic memory)
 - assignment operator overloading
 - your own copy constructor
- Destructor
- the Law of the Big Three

Self-Test Exercises: 1-4, **16 - 23**, 26- 32

Time Analysis of the Bag Class

- `count` – the number of occurrence
- `erase_one` – remove one from the bag
- `erase` – remove all
- `+=` - append
- `b1+b2` - union
- `insert` – add one item
- `size` – number of items in the bag

Exam 1

- Wed, Sep. 28, 4:00 – 5:30 pm
- Two parts (90 minutes, 30 questions)
 - Short Answers (10), e.g.

What is an automatic default constructor, and what does it do?

- Multiple Choices (20), e.g.

Suppose that the `foo` class does not have an overloaded assignment operator. What happens when an assignment `a=b;` is given for two `foo` objects?

- A. The automatic assignment operator is used
- B. The copy constructor is used
- C. Compiler error
- D. Run-time error