

CMSC 132: Object-Oriented Programming II



Software Development III

Department of Computer Science
University of Maryland, College Park

Overview

■ **Software Development**

- **Choosing a software development process**
- **Algorithm and Data Structures**
- **Coding and Debugging**
- **Testing and Verification**
- **Documentation Support**
- **Maintenance**

Choosing a software development process

- Which software life cycle is appropriate when?
- For projects like 132 programming projects, just code and test probably suffices
- But the world isn't like 132 programming projects

Big questions

- **Do you understand what you are trying to build?**
- **What is the cost of change?**
- **How easy is it to get the entire thing in your head?**
- **How many people have to interact with the design?**

Do you understand the problem?

- In many cases, the things we want software to do are not well understood:
 - provide a web interface for students applying for the PhD program
 - build software that allows users to view and manipulate photographs
 - Build a better search engine
- You have to understand the real-world constraints/interactions
- May have to build prototype to understand how users can effectively use it

What is the cost of change?

- Say you've already completed much of the coding, and realize you need to change something in the design or even the requirements
 - how expensive is that?
- If it is hugely expensive, you better get the requirements and design right before you start most of the coding

Has the cost of change changed?

- **Some people have said that recent software development techniques have substantially changed the cost of change**
 - **Safe programming languages**
 - (e.g., not C/C++/assembly language)
 - **OO design programming**
 - **Test driven development**

Sometimes, change is still expensive

- Key nexus in a large system
- Stuff that interacts with hardware that is being co-designed
- Stuff that interacts with software being developed externally
 - Can't change an API once it is published

How easy is it to understand?

- When building and developing software, you need to understand it (at least, parts of it)
- For 100 lines of code, just read the code
- That doesn't work for 100,000 lines of code
- Need to have ways of documenting the requirements and design at a higher level

How many people interact with design?

- **How many people interact with the design?**
- **Part of the cost of change**
 - **If you make a change, how many people need to be aware of or consulted on the design change**
- **Design changes that interact with a lot of people are expensive and need to be minimized**
 - **Try to get these design choices right early and document them**

Algorithms and Data Structures

■ Goal

- Select algorithms and data structures to implement each component

■ Problems

■ Functionality

- Provides desired abilities

■ Efficiency

- Provides desired performance

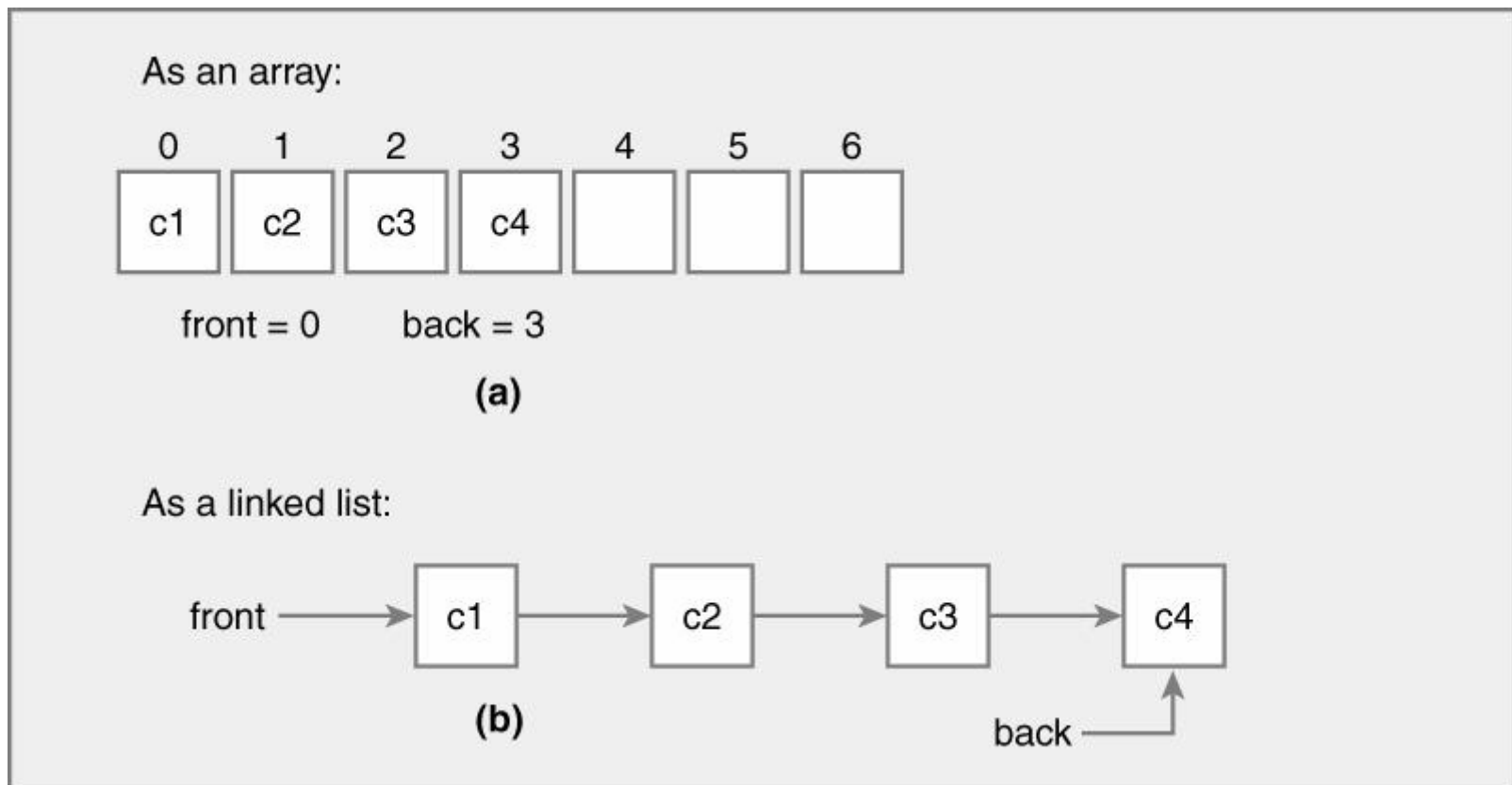
■ Correctness

- Provides desired results

Algorithms and Data Structures

■ Example

■ Implement list as array or linked list



Coding and Debugging

■ Goal

- Write actual code and ensure code works

■ Problems

- Choosing programming language

- Functional design

- Fortran, BASIC, Pascal, C

- Object-oriented design

- Smalltalk, C++, Java

- Using language features

- Exceptions, streams, threads

Testing and Verification

■ Goal

- Demonstrate software correctly match specification

■ Problem

■ Program verification

- Formal proof of correctness
- Difficult / impossible for large programs

■ Empirical testing

- Verify using test cases
 - Unit tests, integration tests, alpha / beta tests
- Used in majority of cases in practice

Documentation and Support

■ Goal

- Provide information needed by users and technical maintenance

■ Problems

■ User documentation

- Help users understand how to use software

■ Technical documentation

- Help coders understand how to modify, maintain software

Maintenance

■ Goal

- Keep software working over time

■ Problems

- Fix errors
- Improve features
- Meet changing specification
- Add new functionality

Project #1

- **Overview of the project and what students are expected to implement.**