CSC 2430 Final Study Guide

The Final Exam will be **open book** and **comprehensive**, covering all topics from the quarter.

Texts: Malik, C++ Programming, Program Design Including Data Structures, 7th Edition

Chapters 1-8: covered in CSC 1230, carefully reviewed in CSC 2430.

Chapter 8

Arrays and Strings (slight coverage CSC 1230, heavy coverage in CSC 2430 including c-strings)

Arrays. Homogeneous aggregates.

C-Strings; the C-standard <cstring> library functions (e.g., strcpy, strcpy_s, strcat_s, strlen, etc.)

Chapter 9 Records (structs)

Structs**.** Heterogeneous aggregates.

Chapter 10 Classes and Data Abstraction

OOP: Object-oriented programming. Encapsulation, Inheritance, Polymorphism

ADT: Abstract Data Type (set of values, set of corresponding operations)

Classes, Encapsulation, public vs. private.

Constructors, destructors

Methods, function overloading. Use of **const** keyword for parameters, or for entire methods.

Chapter 11 Inheritance and Composition ***not covered in 2430***

Inheritance and Polymorphism: ***not covered in 2430***

Chapter 12 Pointers, Classes, Virtual Functions, Abstract Classes and Lists

Pointers, addresses, pointer variables, pointer operations.

Dynamic storage allocation, new and delete

Dereferencing operations (*) and ->; & Address-of operator

Dynamic arrays: new [ ] allocation, delete [ ] deletion, array usage.

Shallow vs. Deep copying

copy constructors, operator=

Inheritance, Virtual Functions, Abstract Classes: ***not covered in 2430***

Chapter 13 Overloading and Templates

Operator overloading, Templates: ***not covered in 2430***

Chapter 14 Exception Handling

Runtime exceptions overview.

try – catch

Creating your own exception classes: ***briefly covered in 2430***

Chapter 15 Recursion *** not covered this quarter in 2430 ***

Problem solving using recursion. Iteration vs. Recursion.

Characteristics of recursive functions (calls itself, solves identical smaller problem, has base case, smaller problem must be base case).

Direct vs. Indirect recursion.

Examples, e.g., Factorial, Fibonacci, Choose K of N items, Towers of Hanoi

Chapter 16 Linked Lists

List ADT. By position lists. By value (sorted) lists.

Array-based lists

Node-based linked lists. Traversing; Node insertion and deletion operations.

Singly-linked lists. Head-pointer (NULL if empty list), single links, NULL link at end.

Variants of the singly-linked list (head/tail pointers, circular)

Doubly-linked list (dummy header node, left/right links)

Full Design: circular doubly-linked with header node

Chapter 17 Stacks and Queues

Intro. to stacks. Stack implementation using arrays or lists.

Intro. to queues. Queue implementation using lists.

Here are a few (briefly stated) examples of questions and topics relevant for the CSC 2430 final exam:

- What is a data type?
- What does the following short program output? (general pointers and pointer arithmetic review)

```
int f(char *s)
{
   char *p = s;
   while( *p != 0 ) p = p + 1;
   return(p - s);
}

int main( )
{
   char buff[25];
                  //            111111111     // character string
                  //  0123456789012345678     //   position offsets from 0-18
   strcpy_s(buff, 25, "Computers Are Great");
   char *p;
   p = buff;
   cout << *p ;               // Output: _____________________
   cout << p ;                // Output: _____________________
   cout << p + 10 ;           // Output: _____________________
   cout << *(p + 10) ;        // Output: _____________________
   cout << &p[10] ;           // Output: _____________________

   int x = f(buff);
   cout << x ;                // Output: _____________________
   x = f(buff + 10);
   cout << x;                 // Output: _____________________
   x = f( &buff[14] );
   cout << x;                 // Output: _____________________

   buff[13] = 0;
   cout << &buff[10];         // Output: _____________________
   return(0);
}
```

- In what situations are copy constructors and assignment operator=( ) methods necessary to be implemented. Why?
- Write a code snippet where a copy constructor is invoked.
  Write a code snippet where an assignment operator=( ) is invoked.

- Assume a dynamic array of size 20 has been allocated and filled up with values,
  ```
  int  *array = new int[20];
  for(int i=0; i<20; ++i)   array[i] = rand();  // fill up with random values
  ```
  and now it is necessary to re-allocate this as an array that is twice as large, but preserving copies of the initial data values in the corresponding locations in the new array. Write code to accomplish this task.

- Implement a class that represents a Rectangle, with 2 float side lengths (length, width), and methods to compute the perimeter length and the area.
- Implement a class that represents a Stack of float values using a fixed-size array of size 25.
  Include the typical methods.
- Implement a class that represents a Queue of integer values, using a singly-linked list.
  Include the typical methods.
- Write a code snippet to insert (or remove) a data value from a linked list.
- Write a code snippet to insert (or remove) a data value from a doubly-linked list.
- Given a linked list of floats, write a function to compute the average of the list of values.