

## ITEC324 - Exam 2 Study Guide

- Each of these topics is described in more detail below:
  - Pattern
  - Adapter Class, Abstract Class
  - Multithreading
  - Recursion
  - Binary Search
  - Sorting
  - Performance (Time Complexity)
- Class Notes:
  - Chapter 5, 6, and 9
  - Recursion and Time Complexity
  - Advanced Sorting and Time Complexity

### Patterns

- ✚ Concept
- ✚ For each of 5 patterns in Chapter 5:
  - ✓ Intent
  - ✓ Context and Solution
  - ✓ Diagram
  - ✓ Examples

### Adapter Class, Abstract Class

- ✚ Adapter Class Programming
- ✚ Abstract Class Programming

### Multithreading

- ✚ Multithread Programming
- ✚ Thread Synchronization
  - Race Condition
  - Deadlock

### Recursion

- ✚ Describe the performance penalties incurred by using recursion and why they occur
- ✚ Describe and illustrate the stack model of execution of recursive routines
- ✚ Give an argument that a recursive routine is correct
- ✚ Write simple recursive routines (including mathematical routines and list routines)
- ✚ Describe the value of recursive routines
- ✚ Contrast the power of recursive routines with non recursive code wrt speed, ease of programming, and what can be programmed recursively vs non-recursively

## Binary Search

- ✚ Know the advantages and limitations of binary search
- ✚ Explain why the performance of binary searching is  $O(\log n)$
- ✚ Be able to write a binary search routine, recursively and non-recursively

## Sorting

- ✚ Merge Sort
  - Hand execute and describe the operation of merge sort
  - Write the routines, recursive as appropriate, to perform merge and merge sort
  - Explain why the performance of mergesort routine is  $O(n * \log n)$
  - State the best/worst case performance of mergesort and explain why each is what it is
- ✚ Quick Sort
  - Hand execute and describe the operation of quick sort
  - Write the routines, recursive as appropriate, to perform partition and quick sort
  - Explain why the performance of quicksort is  $O(n * \log n)$
  - Explain when the performance of quicksort is degenerated to  $O(n^2)$  and prove it.
- ✚ Shell Sort
  - Hand execute and describe the operation of shell sort
  - Discuss the performance of shell sort.
- ✚ Radix sort
  - Hand execute and describe the operation of radix sort
  - Give the performance of radix sort

## Performance (Time Complexity)

- ✚ Understand the summary table of sorting algorithms
  - State the worst case performance of common sorting algorithms: bubble, insertion, selection
  - State best/worst case performance of merge, quick, and radix sort
  - Give basic operation(s) of each sorts