Use the following outline to guide your self-assessment and notetaking

Week 8 – Sort, Search & Exceptions

Sorting (Ch 10.4)

Selection Sort

- Successively places specific values in final sorted positions
  1. Scan the entire list to find the min value
  2. Exchange that value with the value in the first position
  3. Scan the rest of the list to find min value (except for first 2 values)
  4. Exchange the value with the value in the second position
  5. Repeat this process for all but the last position in the list

Insertion Sort

- Sorts by repetitively inserting a particular value into a sorted subset of the list
- Each unsorted element is inserted until the entire list is sorted
  1. Begin with a sorted list with 1 value
  2. Sort the first 2 values in the list relative to each other by swapping if necessary
  3. Insert the 3rd value of the list in the appropriate position relative to the 1st 2 values
  4. Insert the 4th value of the list in the appropriate position relative to the 1st 3 values
  5. Continue until all values are inserted

Comparing Sorts

- Both are almost the same in terms of efficiency
- Both have an outer loop for each value and an inner loop to compare the value in the outer loop
- n^2 number of comparisons (approximate)
  - Some algorithms are more efficient with fewer comparisons

Searching (Ch 10.5)

Linear search

- Essentially, iterates through a list and compares the values to a target
  - If the value equals the target, the value exists and is returned. Otherwise, nothing is found

Binary search

- Works if the elements in the list are already sorted
- Begins in the middle of the sorted list
  - If the target is not found in the middle, the search continues away from the middle until it is found (branches into earlier or later parts of the list, eliminating redundant data in the process
  - First iteration cuts out 50% of the list if the value is not the middle element

Comparing searches

- Binary search is much more efficient, but it requires the data to be sorted first
- Linear search is much easier to implement since it is iterative
- Linear search may be better if long term efficiency is not an issue

Designing for Polymorphism (Ch 10.6)

How do you choose between inheritance and interfaces?

- Depends on the type of relationships involved with the objects
  - If there is a true is-a relationship, then inheritance is preferred
  - However, if each object needs to be processed in a particular way, then interfaces are better

Exception Handling (Ch 11.1)

An exception is an object that defines an unusual or erroneous situation

An error is an unrecoverable situation that should not be caught

Uncaught Exceptions (Ch 11.2)

What happens?

- The program will stop abnormally and will produce a message that describes what the exception was and where it occurred

The try-catch Statement (Ch 11.3)

The try block …

- Represents the normal execution flow and what should be executed most of the time

The catch clause …

- Handles the exception
- Can exist multiple times in the statement, each block denoting what should happen for different exceptions

The finally clause …

- This contains code that runs regardless of whether the exception occurred or not
  - Ex: closing a file after reading or attempting to read it

Exception Propagation (Ch 11.4)

Describe what happens:

- If there are multiple methods and an exception is thrown, the program will return control to the method that produced the exception
- Propagation continues until the exception is caught and handled or until it is passed out of the main method

The Exception Class Hierarchy (Ch 11.5)

Role of the Exception class

- Defines many types of exceptions that can occur in a program
- Exception classes also have many children
- Used to define new exceptions

Checked and Unchecked Exceptions

- Checked exceptions must either be caught by a method or it must be listed in the throws clause of any method that may throw or propagate it
    - Throws clause acknowledges that the method can produce or propagate an exception if it occurs
- Unchecked exceptions do not require throws clauses
    - These are Runtime Exceptions or any decendants

I/O Exceptions (Ch 11.6)

A stream is …

- An ordered sequence of bytes
    - Ex: files
- Input stream – A source where information is read
- Output stream – A source where information is written
- System.in is the standard input stream
- System.out is the standard output stream
- System.err is the standard error stream (output for error messages)