CSCI 232, Fall 2019

Data Structures
and
Algorithms

Java Program Examples

- Binary Search to show changes I am making to the book’s code
  - and get a feel for how we will learn how the code runs

Data types

- Abstract data types

- Abstraction
  - focus on what the object does instead of how it is done
    - Details inside class
    - High level outside class
    - Easier to maintain
    - Can create the classes first or last – apart from the program
    - Reuse
  - Represent the essential feature without detailing the background implementation or internal working detail
  - Focus on only those things that matter our module.
    - Modifying one independent module does not impact the other modules.
    - The only knowledge one needs to know is what a module gives you.
    - The caller of that module does not need to bother about how the task is achieved or what exactly is happening in the background.

What do we need to do with whatever data structure we study?

- add an item
- move through the entire set of data – visiting each item
- Search for an item
- delete an item

Exception Handling

- Example: ExceptionExample.java

Polymorphism

- Definition: having many forms
- 2 types
  - compile time polymorphism
  - run time polymorphism

Generics

- Can specify, with a single method declaration, a set of related methods that work on different types.
    - Allow type (Integer, String, … etc and user defined types) to be a parameter to methods, classes and interfaces. For example, classes like HashSet, ArrayList, HashMap, etc use generics very well. We can use them for any type.
- Programs that uses Generics has got many benefits over non-generic code
    - Reuse: We can write a method/class/interface once and use for any type we want.
    - Type Safety : Generics make errors to appear at compile time rather than at run time

from https://www.geeksforgeeks.org/generics-in-java/