Into to Abstraction –

Many layers of abstraction, e.g. client gets button on their interface, but behind the scenes there is programming that is needed and included. Program code may be inaccessible among two coworkers and make the code available to them, which is otherwise hidden. Logical abstraction is a programming language specific, and which tools, e.g. what to use in ADT?

\*\*Recall previously in Chapter 2 the StringLog example, which involved the concept of abstraction to a lesser degree. Strings in Java are fairly nice and easy. Example StringLog show the imprtance of several O(N) and O(1) operations to build the various pieces (see data levels below).

**Data Levels:** application, logical and implementation (how it gets done).

Application Level – Performs some simple tasks

Logical Level – Programmable abstraction such as interface or contract

Implementation Level – Based on the ADT a specific fulfillment of contract

More Java Concepts Discussed:

Arrays of polymorphic objects:

1. Objects of the same superclass (sharing the same parent class)

2. Variable is declared of the super type

Generics or parameterized type:

1. Interface list or ArrayList implementation of generic interfaces

2. Generics are useful to create a stack (in implementation)

***UMLdiagramming***

1. + is for public

2. # (comment)

3. Consult key for different arrows for implements (interface), and extends, etc.

***Review of Classes and Interfaces (Java programming modules)***

Serializable

Compareable

User Built Interfaces

***Stack***

Stacks start with no elements. Push/Pop are the operations. Top views the top element.

Originally the stack is empty. Push block to populate it.

*Look for posting on canvas of Naive vs. Smart stack implementation – will be on canvas.*

For implementing stacks of generic class make sure to specify the Type with a single letter <E>, <T>,...

***Selected Code from Class***

public interface StackInterface<T> {

public void pop();

public void push(T obj);

public T top();

}

public ArrayBoundedStack() {

elements = (T[]) new Object[100]; }

public class TestClass() {

//inside main method

StackInterface<String> stack = new ArrayBoundedStack<String>();

//still inside main method...time to operate on stack

stack.push }

**For future lesson: Recursive funtions causing StackOverflow.**

Additional Notes from Slides/Textbook

(3.2 Collection Elements)

Collections – objects that hold other objects

Collection ADT – when building the elements it holds, the operations to export, and how exceptional situations are handled.