# Generics and data structures continued

Members Present: Return to:

#### Task 1 – Any questions about the lab from yesterday?

#### Task 2 – Create a simple Stack of…..

Given the code on the attached worksheet (from the Stack example last week), show how you would alter the code to make it generic…what changes would you need to make to accept a type parameter.

Write the statement to construct a Stack of String objects.

#### Task 3 – How do linked lists work….conceptual models of action.

Recall our memory models from prior in the semester. To instantiate an object (such as a String in variable text) we would have something like:

abc

“My important message”

So the variable, abc is holding the reference to where the actual object with the actual data is.

Part 1 – Using the code for IntNode, Node, and Pile, draw the following memory diagrams.

1. Draw a model of an IntNode that contains the value 5.
2. Draw a model of a Node that contains a reference to a String containing "JMU".
3. Draw a model of what memory looks like after the following is executed:

 Pile pile;

 pile = new Pile();

1. Continuing with this same example, now draw a model of what memory looks like after the following is executed:

 pile.push(10);

1. Continuing with this same example, now draw a model of what memory looks like after the following is executed:

 pile.push(20);

1. Continuing with this same example, now draw a model of what memory looks like after the following is executed:

 pile.push(30);

Part 2 - Using the code for Node, Quack, and QuackDriver, draw the following memory diagrams.

1. Draw a model showing the memory state after the following is executed.

 Quack quack;

 quack = new Quack();

1. Continuing with this same example, now draw a model of what memory looks like after the following is executed:

 quack.push("I");

1. Continuing with this same example, now draw a model of what memory looks like after the following is executed:

 quack.push("like");

1. Continuing with this same example, now draw a model of what memory looks like after the following is executed:

 quack.push("ducks");

1. Continuing with this same example, now draw a model of what memory looks like after the following is executed:

 quack.pep();

1. Continuing with this same example, now draw a model of what memory looks like after the following is executed:

 quack.pep();

1. Continuing with this same example, now draw a model of what memory looks like after the following is executed:

 quack.pep();

 Part 3 - Trace the execution of the main() method in the QuackDriver class (using memory models as above). [Note: This example uses the pip() method in the Quack class, not the pep() method.