CSC115 Lecture 9

Pseudo-Quiz #3

Midterm Objectives & Review

Midterm: Thursday February 20

In order to successfully complete the midterm, you must be able to:

- Explain difference between primitive data types and references (to any object, including arrays), including the passing of each as parameters to and returning values from methods. Know how each are represented in memory. Know the scope of variables.
- Create and use arrays, including arrays of objects.
- Differentiate between instance methods vs. static methods—how are they different?
- Call instance methods, from within and from outside their class.
- Explain the purpose of an Abstract Data Type in a large programming project
- Describe how an ADT relates to Java's *interface*.
- Design a Java class according to a specification and/or interface, including choosing appropriate attributes and writing instance methods (ie, toString, equals, getters, setters and helper methods.)
- Instantiate objects
- Write a tester or test cases for a Java class that tests every methods in a class
- Be able to hand trace Java code, showing the contents of memory as the program executes and determine the output. These traces can and will include arrays and linked lists. (Ie, be able to draw memory trace diagrams and then show the corresponding output)
- Write instance methods that implement linked list functionality
- Box trace and write recursive functions.
- Determine and justify the worst case running time of Java methods (O(1) and O(n) only).

Recursive Box Trace Example:

```
public static void countDownBy2(int n) {
  if (n != 1) {
     System.out.println(n + " ");
    countDownBy2(n-2);
   }
}
```

a) **Show a box trace and determine the output when n=7.**
b) **Show a box trace and determine the output when n=6.**

The problem? No Base Case.

Recursive Box Trace Example:

```
public static int fun(int n) {
  System.out.print("Inside f, n = " + n);
  switch (n) {
     case 1: case 2: case 3: return n+1;
    default: return fun(n-1) * fun(n-3);
   }
}
```

a) **Show a box trace and determine the output when n=8.**

Write a Recursive method Example:

The nth Harmonic number is the sum of the reciprocals of the first n natural numbers:

H(n) = 1 + ½ + 1/3 + ¼+ 1/5 + . . . + 1/n.

Write a recursive method to compute the nth Harmonic number.

Write a class, instantiate an object:

- Write a class called RationalNumber that represents a fraction with an integer numerator and denominator. A RationalNumber object should have the following methods:
  - public RationalNumber(int numerator, int denominator)
    - Constructs a new rational number to represent (numerator/denominator.) The denominator cannot be 0 so throw an IllegalArgumentException if 0 is passed.
- public RationalNumber()
  - Constructs a new rational number to represent 0/1.
- public into getDenominator()
  - Returns this rational number's denominator value
- public int getNumerator()
  - Returns this rational number's numerator value
- public String toString()
  - Returns a String representation of this rational number. If the denominatore is 1 omit it and the / sign.
- Reading break challenge #1: Ensure that your RationalNumber objects are always stored in reduced form, avoiding rational numbers such as 3/6 in favour of 1/2 , or avoiding 2/-3 in favour of -2/3.
- Reading break challenge #2: Add methods to add subtract, multiply and divide two rational numbers.

Example: Using the RationalNumber

- Write a tester that tests all of your methods.

Example: reversing a Linked List

- Given a singly linked list class, add an instance method that reverses the order of the list. Do not move the data elements, only change the head and next pointers. For example, the list:

becomes: