CSCI 145 -- Review for Final

Textbook

- Go over examples in the book
- Go over self-review questions
- Do some chapter exercises
- Try programming projects 9.3 (class diagram only), 10.2, 11.2, and 12.1

Review

- HW 1 to 3, PAs 1 to 12, in-class exercises and pop quizzes, exams 1 and 2
- Java Programming Language -- basic syntax, compiler and interpreter, types of errors, Java virtual machine (JVM)
- Primitive Data Types -- integers, floating-points, characters, boolean
- Common Java classes – Math, Random, String
- Input/Output -- System.out and Scanner, reading and writing text files
- Expressions -- formation of expressions, order of evaluation, and conversions (widening, narrowing, and casting)
- Methods -- parameters, return type, method headers
- GUI – common components such as panels, frames, text fields, etc.; events and listeners; layout managers
- UML -- what is it? class diagram for one class and multiple classes
- Arrays -- declaration, process an array, initializer lists, bounds checking, two-dimensional arrays, sorting and searching
- References -- references vs primitives, null, "this", alias
- Class Relationships -- dependency, aggregation, and inheritance; UML class diagram
- Applets/Graphics -- drawing strings and shapes; polygons and polylines
- Software Engineering – SW development models, testing: white box vs. black box testing, testing vs. debugging
- Control Structures -- selection (if, switch, operator ? :); repetition (while, do/while, and for), counting loops, sentinel loops, yes/no loops; nested selection and nested loops
- Conditions -- true/false, relational operators, logical operators, short-circuit evaluation
- Classes -- define a class, methods, public vs. private, constructors, this keyword
- Interfaces -- implement an interface, ***Comparable*** interface, abstract methods
- Inheritance -- base classes and derived classes, protected members, super reference, overriding vs. overloading, abstract base classes (can have abstract methods and no objects can be created), Object class
- Polymorphism – define, polymorphism via inheritance and via interface
- Exceptions -- why, keywords: try, catch, finally, throw, and throws; unchecked exceptions vs. checked exceptions; common exceptions
- Threads – no code, thread states, class ***Thread*** and interface ***Runnable***
- Recursion -- base case and recursive case, recursion vs. iteration, n!, x^n, reverse an array, print array in reverse order

- Data Structures – familiar with linked lists, familiar with well-known data structures: Stacks and Queues
- Program Development -- analysis, design, code, and test; tracing code
- Problems – set up classes, use existing classes, set up methods, use arrays effectively

Some Review Questions/Exercises

1. Generate a random number between 5 and 20
2. Differences between primitive type and reference type, draw diagrams
3. Determine output for a segment of code

```
for (int i = 1; i <= n; i++)
{
        for (int j = i; j > 0; j--)
        System.out.print("#");
        System.out.println();
}
```

4. Translate pseudocode to Java code
   a. Creating an object of type X
   b. Invoke method m1 of this object sending an int value
   c. Print the object
5. What is a constructor and how do you set one up?
6. Set up a method that accepts some parameters and performs a task – determine if n is prime or power of 2, reverse an array
7. Perform some work with a string using various string operations – given a string, convert the first letter of each word to uppercase; given a string, count vowels
8. Complete the code for a class – constructors, getters, setters, equals, toString
9. What are some levels of testing from earliest to latest?
10. Best loop to implement a sentinel loop, a counting loop, and a y/n loop?
11. Testing vs. debugging; black box vs. white box testing; when should you stop testing?
12. Perform a simple if-else with the conditional operator – min of two values
13. Use a nested loop to generate a pattern like a rectangle or a triangle
14. Draw a simple UML class diagram showing relationships between some classes
15. Use the Coin or Die class to perform a simulation; roll two dice 100 times and count the number of times they have the same value
16. Use an array to keep track of the counts for n items
17. Describe three different relationships between classes
18. How is method overloading different from method overriding?
19. How try-catch works; checked exceptions vs. unchecked exceptions; what happens when an exception is thrown?
20. Set up a simple recursive method to print a string in reverse order; trace a recursive method such as factorial or summing values from 1 to n
21. Explain polymorphism and how to achieve polymorphism in Java

---