CSC 20 Program Concept + Method II – Spring 2018

Final Exam Review Document (5/4/18)

The final covers the following topics:

- **Data and Expressions**
  - **Character Strings:** be able to print strings using print and println, use escape sequences correctly, and understand string concatenation.
  - **Variables and Assignment**: declare and initialize variables and constants.
  - **Primitive Data Types**: understand and use the different primitive data types (byte, short, int, long, double, float, char, boolean), understand the differences between them,
  - **Expressions**: write and evaluate arithmetic expressions, understand operator precedence, increment and decrement (postfix and prefix), assignment operators.
  - **Data Conversion**: understand and use promotion and casting.
  - **Interactive Programs**: use the Scanner class to input data from the user.

- **Using Classes and Objects**
  - **Creating Objects**: understand how to create objects, assign them, invoke methods on them and re-assign references.
  - **The String Class**: understanding and using strings and String methods
  - **Packages:** use import statement when needed
  - **Wrapper Classes**: use Wrapper classes to parse strings.

- **Writing classes**
  - **Anatomy of a Class**: understand the concept of a class, define its instance variables, constructor and methods.
  - **Anatomy of a Method**: understand class methods, be able to write methods.
  - **Encapsulation:** understand visibility modifiers, write accessors and mutators, write the toString method.

- **Conditionals and Loops**
  - **The if Statement and Conditions**: use if statements, understand and use boolean expressions and logical operators.
  - **Other Conditional Statements**: understand and use if-else statements, the conditional operator, nested if-else statements and switch statements.
  - **Comparing Data**: be able to compare primitive data types, objects and strings properly.
  - **The while Statement**: understand and use while loops, nested loops. Recognize infinite loops
  - **Iterators**: use the Scanner class to read files and divide sentences into words.
  - **Other Repetition Statements**: understand and use for loops and do-while loops.

- **Object-Oriented Design**
  - **Identifying Classes and Objects**
  - **Class Relationships:** understand and use dependencies and aggregation relationships to communicate between classes. Understand and use the "this" reference. Understand how to use **Abstract** classes.
  - **Interfaces**: understand the concept of interfaces and be able to implement them in classes.

  - **Method Design**: be able to decompose methods into smaller easier to manage methods**.** Understand parameters passing in Java. Use method overloading and constructor overloading correctly when needed.

- **Arrays**
  - **Declaring and Using Arrays:** understand the concept and structure of an array, declare an array, use its elements, access its contents, check its bounds using the length variable, understand ArrayIndexOutOfBoundsException, understand and use initializer lists, and pass arrays as parameters to methods
  - **Arrays of Objects:** understand and use arrays of objects (i.e CSUS Students), understand and use command-line arguments.

- **Inheritance**
  - **Creating Subclasses:** understand the concept of inheritance, create subclasses, understand and use the protected modifier, the super reference.
  - **Overriding Methods:** understand and use the concept of overriding methods, differentiate between overriding and overloading methods.
  - **Class Hierarchies:** understand the concept of class hierarchies, the position and importance of the Object class, and define abstract classes.
  - **Inheritance and Visibility:** understand the visibility of superclass members in subclasses.

- **Polymorphism**
  - **Polymorphic References:** understand the concept of Binding and Polymorphism
  - **Polymorphism via Inheritance:** understand and use polymorphism via inheritance.
  - **Polymorphism via Interfaces:** understand and use polymorphism via interfaces.
    - Study your final prelab closely.

- **Recursion (study slides set on Recursive Lecture closely)**
  - **Recursive Thinking:** understand recursive definitions, be able to exercise recursive thinking and understand and recognize infinite recursions. Review binary search algorithm and quicksort.
  - **Recursive Programming:** understand and use recursive programming, be able to write recursive methods as well as tracing them.
    - **Study the f**irst practice problem from Code Step by Step (**`collapseSequences`**)
    - **Study the printStar** examples from lecture.

- **Collections**
  - Understand the concept of Linked List, Stacks, and Queues Abstract Data Types (ADT). Know how to apply and use their operations (see Palindrome lab).

- **Analysis of algorithms**
  - Understand Big O notation. Know how to analyze simple programs using primitive operations.

- **Sorting:** understand and demonstrate the usage of Bubble, Selection Sort, Quicksort, and Insertion Sort. Analyze the algorithms under worst case scenarios.
- **Searching:** understand and demonstrate the use of Linear Search and Binary Search (non recursive and recursive versions). Know their Big-O.