Exam 2 Study Guide

Preparing for Exam 2

- Review the learning objectives
- Review the lecture slides
- Review zyBook, and the activities
- Review (and possibly rewrite) the lab exercises
- If you do not fully understand a topic, read the related textbook section
- Attend office hours to ask additional questions/clarifications
- Complete the practice problems

Exam 2 Learning Objectives

- Formatting Text with printf
    - Write code that used the `System.out.printf` to output a formatted string.
- Random Numbers
    - Write and trace code that uses random numbers (`Random`).
- Equality, Relational, and Logical Operators
    - Evaluate expressions containing equality operators.
    - Evaluate expressions containing relational operators.
    - Evaluate expressions containing logical operators.
    - Write code containing equality operators.
    - Write code containing relational operators.
    - Write code containing logical operators.
    - Negate boolean expressions using De Morgan's Law.
- boolean Data Type
    - Evaluate boolean expressions, including short-circuit evaluations.
- Conditionals
    - Trace conditional (e.g., if or if-else) statements and provide output.

    - Write conditional (e.g., if or if-else) statements to perform an operation or produce specified output.
    - Distinguish between when to use each conditional structure.
- Returning Within a Conditional
    - Write code that returns within a conditional.
    - Trace code that contains multiple return statements.
- Strings
    - Construct new String objects.
    - Write and trace code that uses String methods: length, equals, equalsIgnoreCase, indexOf, charAt, substring, contains, replace, toLowerCase, trim…
- Character Operations from Character Class
    - Write and trace code that uses Character class methods: isDigit, isLetter, isLowerCase, isUpperCase, toLowerCase, toUpperCase…
- Scanner Next methods
    - Construct new Scanner object for console input.
    - Write code to read user input from the console with nextInt, nextDouble, next, and nextLine methods.
    - Trace code containing Scanner nextInt, nextDouble, next, and nextLine methods.
- Scanner hasNext Methods
    - Write robust code that checks user input using Scanner hasNext methods: hasNextInt, hasNextDouble, hasNext, and hasNextLine methods..
- while Loop
    - Trace a while loop and provide output along with number of times loop body executes.
    - Write a while loop to perform an operation or produce specified output.
- do-while Loop
    - Trace a do-while loop and provide output along with number of times loop body executes.
    - Write a do-while loop to perform an operation or produce specified output.

- For Loop
  - Explain the benefit of using a `for` loop.
  - Describe the structure of `for` loop and flow of control.
  - Trace a `for` loop and provide output.
  - Write a `for` loop to perform an operation or produce specified output.
- Nested Loops
  - Trace nested loops and provide output.
  - Write nested loops to perform an operation or produce specified output.
- Assertions
  - Identify the various assertions in code as being either always true, never true, or sometimes true/sometimes false at various points in program execution.
- Scope
  - Identify the scope of a variable.
- File Input (Token-Based Processing and Line-Based Processing)
  - Write and trace code that uses a `Scanner` to read from a file.
  - Write and trace code that uses a `Scanner` to tokenize a `String`
  - Write and trace code for file input using token-based processing.
  - Write and trace code for file input using line-based processing.
  - Describe the difference between token-based and line-based processing.
- throws Clause
  - Describe when a `throws` clause is needed.
  - Write and trace code that uses a `throws` clause.