Mathematics 2170 : Exam 2 Topics

General comments

You are responsible for the topics covered in Labs 1 through 9, slides, lectures, worksheets, and handouts for weeks 1 through 9, and all assigned readings in the textbook. It is impossible to have an exam exclusively on the most recent topics, since this class builds on prior knowledge and skills. However, to the extent possible, the emphasis will be on material covered since the first exam.

To do well on this exam, you should be familiar with the concepts for all lab exercises for weeks 1 through 9. One thing often overlooked when preparing for exams in this class: you need to truly **understand** the solutions to lab exercises, textbook examples, and lecture material. Some students find that they can make their lab programs eventually do what they want, but it might involve lots of trial and error. You don't have the luxury of trial and error in an exam setting: you must be able to demonstrate that you understand the concepts and can solve similar problems with a well-reasoned approach.

The format is similar to the first exam. Many questions are short-answer that ask about concepts, require understanding of brief excerpts of Java code (including tracing), or ask you to supply Java statements to solve some portion of a problem. Other problems will expect you to provide a larger amount of Java code, such as providing a complete method and an invocation for it.

Important textbook material

- Section 3.5: Boolean expressions
- Section 4.3: The `if` statement
- Section 4.4: The `switch` statement
- Section 4.5: The `while` statement
- Section 4.6: The `for` statement
- Section 5.1: A quick overview of methods
- Section 5.2: Writing your own methods
- Section 5.3: Mechanics of the method-call process
- Section 5.4: Decomposition
- Section 5.5: Algorithmic methods
- Section 6.1: `RandomGenerator` class

Lab Exercises

- Lab 5: Double Curve Stitch, Two–tone triangle, Nested Ovals, and Pyramid
- Lab 6: Four–tone Checkerboard, Solving Quadratic Equations, Checkers, and Bouncing Ball
- Lab 7: Counting Digits, Perfect Numbers, and Bread Machine
- Lab 8: Bullseye, Finding Prime Numbers, Numeric Palindromes and the Julia Set
- Lab 9: Chaos Game, Approximating π, and Sine & Cosine waves

Provided materials

The exam is closed-book, although you may bring and use notes on one sheet of $8\frac{1}{2} \times 11$ inch paper, and these summaries will be provided for your use during the exam:

- Table of `Color` names (page 43)
- Summary of selected methods from `acm.graphics` (page 45)
- Summary of selected methods from the `Math` class (page 137)

Continues on next page.

Topics and skills you should know

- Relational operators: `<` `<=` `>` `>=` `==` `!=`
- Logical operators: `!` `&&` `||`
- Truth tables
- DeMorgan's laws: `!(p && q) == !p || !q` `!(p || q) == !p && !q`
- Use of named constants
- Simple, compound, and control statements
- The repeat-N-times pattern
- The read-until-sentinel pattern
- Forms of the `if` statement:
  - single-line `if`
  - multiline `if`
  - `if-else`
  - cascading `if` statements
- The `while` statement
- The `for` statement
- Using loops to compute summations and products
- Nested `for` statements (example: checkerboard)
- Simple graphical animation: loops, `move` and `pause`
- Using methods others have written ("consumer" view)
- Designing our own methods ("designer" view)
- Methods from the `Math` class
- Method interface parts: scope, return type, method name, parameters
- Method implementation: header + body
- Variety of method implementations:
  - no significant control structures (example: temperature conversion)
  - involve internal control structures (example: factorial)
  - can return primitive types and non-numeric types, or be `void`
  - predicate methods
  - methods which invoke other methods (example: combinations)
- Parameter passing (especially the 5 steps on page 149)
- Decomposition
- Algorithmic methods (example: Euclid's gcd)
- Utilizing the `RandomGenerator` class