Midterm Exam Review

1 Format and Timing

- This will be administered as two Canvas Quizzes.
- Part 1 will become available at 6pm and close at 7:10pm.
  - Matching questions are self-grading; others require my review.
- Part 2 will become available at 7:20pm and close at 8:30pm.
  - You must not only complete the exam but also upload any files during that time.
  - Save code in pristine (untouched) form until the exam is graded.

2 Question Types and Percentages

Part	Type	Description	Count	Points	Percentage
Part 1	Essay	Answer questions concisely but thoroughly	2	10	20%
Part 1	Matching	Match the data conversion concept to a code sample that demonstrates it	4	5	10%
Part 1	Matching	Match a description of a to the corresponding primitive data type	8	5	10%
Part 1	Matching	Match a description to the corresponding OOP term	10	10	20%
Part 2	File Upload	Given narrative text describing the problem, draw a UML class diagram	1	5	10%
Part 2	Upload	Using the narrative text and the UML you drew above, write OOP code to meet the requirements	1	15	30%
Totals				**50**	**100%**

3 Topics

Be prepared to describe, compare, contrast, etc. Prepare to write code where applicable.

- Classes: create from description and/or UML class diagram; includes private member data, constructors (calling each other, if necessary, to avoid code duplication), accessors, mutators, and other methods (public or private).
- Compiled vs. interpreted languages and Java's place among them.
- Conditional execution (aka selection control structures) including nesting.
- Console input and output; common output escape characters; data type and range checking.
- Data types: primitives, immutable types, numeric types (and their differences), Booleans.
- Expressions: operators, precedence, data type implications, Boolean expressions, logical operators, widening (automatic) and narrowing (casting).

- Keywords: this, static, final.
- Loops: definite, indefinite (both types), specifying simple and/or complex conditions, exit sentinels.
- Math library: sqrt, pow, min, max, round.
- Methods: creating static methods including main; creating non-static (instance) methods.
- Object-oriented languages: what they are.
- Objects: create references, instantiate objects.
- Preconditions: testing for them and throwing exceptions if they are violated.
- Scanner: instantiate and use to collect user input; use nextInt, nextDouble, next, nextLine.
- String: data type, simple methods (e.g., change case, find length), testing for equality.
- ~~Testing: creating tests for constructors and other methods; testing preconditions.~~
- UML class diagram: three sections for each class, marking access modifiers (public vs. private), showing defaults, using data types, showing methods. Prepare to draw one given a problem description.
- Variable declarations and assignments (including shorthand assignments).