Midterm 2 Review
Arrays

- Declaring and populating two-dimensional arrays
- Find the lengths of arrays in a two-dimensional array
- Looping through a two-dimensional array (both by rows and columns)
- Passing arrays to methods
- Multi-dimensional arrays
Processing

• How to set up a processing sketch

• Standard methods (main, settings, setup, draw) and what they do

• Drawing basic shapes, changing the colors, changing the background

• keyboard and mouse interaction
Objects

- Identity, state and behavior of objects
- Creating and using classes
- Constructors, including multiple constructors, private ones, etc
- Instance variables (and null initialization)
- Static methods and data fields Section 9.7 (!)
- . (dot) operator to access data fields and methods
Static modifier

- Static methods and variables belong to the class, not to the instance

- They are the common property shared by all objects of a class, and are not unique to any one instance.

- Static methods can only access static data

- Static methods can only call other static methods

- Static methods cannot use `this()`
Static modifier

• Examples of when you might use static variables or methods:

  • Keeping a count of all objects created by a class (static int totalCount)

  • creating a helper class with useful methods that you pass data to (Math class - Math.min(3,2))

  • “Does it make sense to call this method or access this variable, even if no objects have been created from it yet?”
Objects

- Variable scope
- Hidden variables and the “this” keyword
- Class abstraction
Objects

- Relationships between classes (association, aggregation, composition)
- Basic UML of classes and relationships between classes
Objects

- Date class
- Random class
- Visibility modifiers
- Data field encapsulation (getters and setters)
- Passing objects to methods
- Array of objects
- Immutability
Strings and things

• Strings, StringBuilder, StringBuffer

• methods available in those classes

• Converting between Strings and numeric data types and between numeric data types

• Wrapper classes for numeric data types, boxing and unboxing (Primitive vs reference variables)