

## CS-302 Midterm 1 - Terminology Review

Some definitions here are taken from the glossary at the back of the textbook (p. 547 in the second edition).

**Class** - A container for methods (*my definition--we will get into more details about classes later in the course*)

**Method** - A piece of code with a name. A method has a list of parameter variables (which may be empty), a return type (which may be void) and a body containing zero or more code statements.

**Array** - A collection of values of the same type stored in contiguous memory locations on the heap, each of which can be accessed by an integer index.

**Variable** (also called a "local variable") - A symbol in a program that identifies a storage location that can hold different values.

**Flag** - A variable of type `boolean`, which can have one of two values: `true` or `false`.

**Stack** - A data structure with "last-in, first-out" retrieval. Elements can be added (pushed) and removed (popped) only at one position, called the top of the stack. Java uses a stack to store the activation records (ARs) of all the methods that are currently executing. Each method's AR stores its local variables and parameters.

**Heap** - The area in memory where a Java program can allocate arbitrary amounts of memory using the `new` keyword.

**Caller** - A method that calls another method.

**Callee** - A method that is being called.

**Call site** - The place in code where a method call happens.

**Primitive type** - A numeric type (`int`, `double`, etc.), `char`, or `boolean`. Primitive types are stored without reference. The actual value of a primitive variable is copied when it is assigned or passed to a method.

**Reference type** - Any non-primitive type. Includes arrays and classes. Reference types are stored in the heap and accessed by reference. Only the reference is copied when a reference variable is assigned or passed to a method.

**Object** - An instance of a class, stored in the heap and created with the `new` keyword. Each object provides a set of methods that may be called on that object.

Also, make sure you are familiar with the conventions for drawing flowchart diagrams. We use diamonds to represent nodes that evaluate a condition and make a decision (this includes `if` and `while` statements). We use boxes to represent other code.

