Static Variables and Methods

**Class** - Defines the data and behavior for a class of objects. A cookie cutter. A type. A form.

**Object** -- an instance of a class. A cookie. A variable.

**Static** -- data or method associated with a class, not an instance.

*Static methods* are generally used when you want to collect some functionality but there is no data associated with that functionality.

- Math functions, e.g., `Math.sqrt(x)`
- Character class functions
- public static void main (String args[])

*Static data*-- Data associated with a class, not each instance

In java, often use 'static' to define constants. Constants are values that never change. It is good programming practice to use a symbolic constant instead of referring to literal values in your program.

- What literal values did you refer to in Mastermind?
- What literal values did you refer to in ParkingLot?

The keyword 'final' is used along with static to define a constant:

    static final int LOTSIZE = 20; // define a constant for parking lot size

Since the variable is a static (class) variable, you refer to it with:

    ClassName.VarName e.g.,
    ParkingLot.LOTSIZE

*Instance methods* -- a method associated with an instance (this is the norm):

    car.move()
    person.older()
    student.getGPA()
Instance data -- Data associated with an instance.

name, age in Person

init, current, final in Car

Instructor Sample: create a class and define statics/instances.

In-Class Problems

1. Create a new project with a class Char. Write a static method `isLetter` that accepts a char and returns a boolean.

2. Modify your intArray or ArrList class so that it uses a static constant `BLOCKSIZE` instead of 10 to create the initial array.