# Functions/Methods: Coding to Scale

### Functions/Methods

- Way to encapsulate or wrap up a piece of functionality
- Creates a "command" that can be called
- For this course, will use the terms "function" and "method" interchangeably
- Seen many examples. e.g.:
  - ≻ draw()
  - ≻ setup()
  - ➤ rect(...)
  - ≻ fill(...)

### **Methods Provide**

- Modularity
  - Divide and conquer
  - Break code into smaller pieces
  - > Easier to understand, debug and maintain
  - Function call can replace *many* lines of code
     Better readability
- Reusability
  - Create functionality that can be reused anywhereWrite once, use often
  - New "commands"
  - > Other people could call the new function
- Can write your own

# General Form of Method

#### <ReturnType> name(<arguments>)

- { //code
- }
- Return type is required
  - > "void" means method doesn't return a value
  - > Return type could be a variable or object type
- Arguments are optional
  - > 0 or more. Must specify type (e.g. int or float)



# Example of Function Being Called

float num = 100; float result = average(num, 50); println(result);

# Simple Method to Build a Function

- 1. Write code in draw()
- 2. Test it and make sure it does what you expect
- 3. Cut and paste the block of code to outside draw()
- 4. Add the return type, method name and braces

void foo() {

//your code

- }
- 5. Replace code in draw with call to function
- 6. Add arguments
- 7. Use your function in other ways

## **Example: Drawing Stars**

## What does this code output?

void passVariable(int a) {
 a = a + 5;
}
void setup() {
 size(600, 600);
 int testInt = 5;
 println("Before call: testInt " + testInt );
 passVariable(testInt);
 println("After call: testInt " + testInt );

# Passing by Copy or Passing by Reference

- Variables are passed by value
  - "Passed by value" or "passed by copy" mean the same thing
  - > Function gets the value of the variable
  - Making changes to this copy will NOT change the value of the original variable
  - > Output to previous code is 5



# Passing by Copy or Passing by Reference

- Objects are passed by reference
  - > Function gets a reference to the actual object
  - Making changes to this reference will change the object
  - > We'll revisit this when we know what objects are!

