access Type: controls whether or not the method can be accessed from outside the class.

return Type: declares what type of value is returned by the method. We use void as the return type for methods that don’t return anything.

method Name: The unique identifier for the method

parameters: allow us to pass primitive data into a method. If we are not passing data into our method, we will leave the parentheses blank. Parameters are also referred to as arguments.

static: static is an additional modifier that allows a method to be run without instantiating an object. Since no objects exist in our program before it runs, The main() method must be static in order to act as our entry point.

void: This is the return type for the method. A return type can be set to a variety of data values. However the main method is always set to a return type of void. This means that the main() method does not return a value.

parameters: String[]args is actually a placeholder for parameters or arguments. Although passing arguments into the main() method is beyond the scope of this course, we will discuss the meaning of String[]args in a later lesson.

- %[#]s creates a spacing that allows for indent for alignment, so long as all the lengths are within parameters

- %[#].[#1]f allows for the spacing to be 10 characters, but also includes "#1" amount of decimal places in it

- If "#1" does not include all the decimal places, the excessive portions would be cut off.

- If numbers exceeed given parameters, the printf function would not space correctly

- For sake of formating, "public void FUNCNAME () {}" should be used to include any change of existing variables

- Parameters are only visible to the function itself, and will work so long as the correct values were plugged in

- You would need to make a new class in order to run varNAME.FUNCNAME(vars)