2. A method declaration is always the first line of a method, and it contains important information about the method such as its name, access level, return type, and parameters. The method body on the other hand is made up of the statements that are executed when the method is called.

3. The keyword used is an access modifier.

4. Another word that can be used is visibility.

5. The variables var3 and var4 are within the scope of the method method1. Method1 is also in the scope of the class ScopeExample, so therefore var 3 and 4 are also within that scope. All variables (var1, var2, var3, var4) are within the scope of the main method because 1 and 2 are declared within that method and then the main method also calls method1 where the other two variables are located.

6a. `public static int getVowels(String vowels)`
6b. `public static int extractDigit(int digit)`
6c. `public static String insertString(String iString, int position)`

7a. Either by their unique method names or by method overloading.
7b. Yes they can. If you use a technique called method overloading (using two methods of the same name but giving them different parameters) you can make two methods in the same class have the same name.

8a. It is used to return a value back to the calling method.
8b. One value.
8c. When you make a method declaration return a value, it means that you are able to send that value back to the statement that called the method. If the declaration does not return a value, then that method will not be able to use a return statement to send information back to the calling method.

9. The issue with the code is that the line doSomething(); is completely useless. You don't need to call the method there because you end up just calling it again in the next line, and you only need to call it once to assign the value of 5 to the variable num.

11a. True.
11b. False, a method call consists of the method's name with parenthesis next to it.
11c. False, a void method returns no values.
11d. False, an access modifier actually tells other methods whether or not they can access that method.
11e. True.
11f. False, they are enclosed by parentheses.
11g. False, local variables can only be used by the method they are in.
11h. True.

11i. False, method overloading is a technique that allows a program to have multiple methods of the same name.
11j. True.
11k. True.
11l. True.