* Within a method, you may make multiple types of statements, including:
  + Declaration of local variables
    - Ex: int myNumber;
  + Assignment
    - Ex: myNumber = 2;
  + Conditionals
    - Ex: if(myNumber == 2) {enter statement}
  + Loops
    - For loops, while loops, do while loops
  + Method calls
    - someMethod();
    - someObject.someMethod();
  + Return statements
    - Return; -> for void methods that don't return anything
    - Return someType; -> for a method that returns some type

* Within a method, if you create a variable, this variables is called a local variable
  + It is created when the method is called, then destroyed after the method is done or hits a 'return' statement
* When naming variables, keep it simple and logical
  + Cannot start with a number, cannot contain whitespace
* A+= b just means A = A +b
  + Same is true for multiplication, division, subtraction
  + a\*= b is same as a = a\*b
* Switch statements are used when a object(usually a variable) can potentially have multiple values, and you want to do different things depending on what it is
  + So an example could be an object myFruit, which could be a lot of types of fruit like so:

switch(myFruit) {

case banana:

sliceBanana();

break;

case apple:

chopApple();

break;

case pineapple:

skinPineapple();

break;

Default:

someDefaultMethod();

}

To import a jar file:

1. Save jar file somewhere
2. Create        a new project in Eclipse.
3. Right click the src folder         in the project and choose, “Import…”
4. Choose        the type General->Archive and click
5. Next Browse for and select the JAR file.
6. Click Finish

* Within a loop, you can use "continue" to proceed and "break" to escape the loop
* Ternary operator uses a question mark
  + Shorthand version of an if then else statement that results in a value
  + Some\_expression ? If\_true: if\_false
    - If true, you get the first part, if false you get the second value
    - Used to choose the appropriate action depending on the value of an expression

* Bitwise stuff looks like this: ~, <<, >>, >>>, &,|, ^
* A Byte is a value from -128 to 127
* If static is in the signature of a method, this means the method is class method
  + no need to create an object to call the method on, just call ClassName.method() rather than ClassName classInstance = \_\_\_\_; classInstance.method()