

Student Name _____

- A method provides a way to encapsulate code in one place so that it can be reused. Its general form is

Method Heading
Method Body

where the heading has the general form

```
modifiers returnType methodName(parameterDeclarations)
```

and the body is a sequence of statements enclosed in curly braces ({ and }):

```
{  
    statements  
}
```

- Values (called arguments) can be passed into a method via parameters.
- Execution transfers from a method back to the caller when a `return` statement is encountered or the end of the method is reached.
- A `return` statement is not required for methods whose `return` type is `void`.
- A specification for a method should include descriptions of values it receives – parameters and their types; values input to the method; the value it returns; values it outputs; preconditions; and postconditions.
- A method's documentation should include a comment that describes its specifications.
- Locals exist only while a method is executing and thus can be accessed only within the method. This means that other methods may reuse the name of a local for some other purpose without causing a conflict.
- When a method is called within the class in which it is defined, its name can be used without qualification.
- When a method is called, the arguments are associated with the parameters from left to right - the first argument with the first parameter, the second argument with the next parameter, and so on, until the matching is complete. There must be the same number of arguments as parameters and each argument's type must be compatible with the type of the corresponding parameter.