DESIGNING METHODS

Parts of a Method Declaration

- A **method declaration** specifies all the information needed to call a method.
- Example of a method declaration:

```
public final void nap(int minutes) throws Exception {
    //statements
}
```

- The parts of a method declaration are as follows:

Element	Value in nap() example	Required?
Access modifier	`public`	No
Optional specifier	`final`	No
Return type	`void`	Yes
Method name	`nap`	Yes
Parameter list	`(int minutes)`	Yes, but can be empty ( )
Optional exception list	`throws Exception`	No

Method body	`{`
	`//statements`
	`}`
		Yes, but can be empty { }

  To call the `nap()` method, just type its name followed by a single `int` value in parentheses. Ex. **`nap(15);`**
- An **access modifier** specifies whether other classes can use a particular field or invoke a particular method. The access modifiers in Java are `public`, `protected`, package-private (no modifier), and `private`.
- The most common optional specifiers are the following:
  - `static` is used for class methods.
  - `abstract` is used when a method does not have a body.
  - `final` is used when a method is not allowed to be overridden by a subclass.
- **Return type** is the data type of the value returned by the method, or `void` if the method does not return a value.
- The **method name** should follow the naming convention rules below:
  - Should be a verb in lowercase (ex. run); or
  - A multi-word name that begins with a verb in lowercase, followed by adjectives, nouns, etc., that starts with capital letters (ex. runFast)
- The **parameter list** is a comma-delimited list of input parameters, preceded by their data types and enclosed by parentheses. Empty parentheses are allowed if there are no parameters specified.
- The **optional exception list** indicates that the method might throw one of the listed exception types.
- The **method body** is everything that is enclosed between braces. It contains the method's code including the declaration of local variables.

Access Modifiers

- Two (2) levels of access control:
  - Top level – `public` or package-private
  - Member level – `public`, `private`, `protected`, or package-private
- The **`public`** modifier allows the method to be visible to all classes everywhere.
- The **`private`** modifier specifies that the method can only be accessed in its own class.
- The **`protected`** modifier specifies that the member can only be accessed within its own package and by a subclass of its class in another package.
- The **package-private** modifier (no modifier) specifies that the member can only be accessed within its own package.

References:

Baesens, B., Backiel, A., & Broucke, S. (2015). *Beginning Java programming: The object-oriented approach*. Indiana: John Wiley & Sons, Inc.

Farrell, J. (2014). *Java programming* (7th ed.). Boston: Course Technology, Cengage Learning.

Oracle Docs (n.d.). *Citing sources.* Retrieved from https://docs.oracle.com/javase/tutorial/java/javaOO/index.html