B – Starting Classes

# Overview

This topic introduces some of the basic syntax and grammar surrounding Object-Oriented Programming in Java. The following keywords are introduced. (Note that additional keywords from previous topics may also be present.)

* new
* char
* int
* double

This topic will also introduce the following grammars, syntax and language constructs. (Note that additional concepts from previous topics may also be present.)

* Method declaration syntax (instance methods)
* Object instantiation
* Instance method call syntax (with and without arguments)
* Class Field declaration syntax (member variables)
* Assignment statement syntax
* Accessing public fields in objects (storing & retrieving values)

Note that this package introduces the distinction between **static** and **non-static** methods. Static methods, also known as "class methods", are identified by the keyword static and must be qualified by the class name; for example, in the following method call Salutation is the name of the class and Greeting is a static method of that class.

Salutation.Greeting()

Non-static methods, also known as "instance methods", are identified by the **absence** of the static keyword and must be called by using an instance of the class (that is, an ***object***). An example of calling an instance method would be

app.Greeting()

where the app is an instance of the Salutation class.

**Daily LOGs**

The following daily LOGs are covered in this package.

**OOP Basics**

##### 

* Define the term "object" as used in OOP
* Describe the syntax of static and non-static method declarations
* Compare and contrast the syntax for method declarations (implementations) vs. method calls (for both static and non-static methods)
* Define the term "field" as used in OOP and give an example
* Compare and contrast instance and class members, for both methods and fields
* Identify the "member access operator" and explain it's purpose
* Create (instantiate) objects
* Distinguish between classes and objects in OOP
* Explain what is meant by the phrase “type extensibility” and how classes represent “complex” data types in OOP

**General Programming Concepts and Terms**

* List the three categories of simple information
* Define the term "intrinsic data type"
* Explain what an assignment statement is and how it works
* Compare and contrast the three categories of simple information with the complex types of information which classes represent
* Explain the advantages of using classes and objects instead of only using intrinsic data types