

# CS 2100 – Java Review [01]

## Data Structures and Algorithms 1

### Basics about Java

#### Main Components of a Java Program:

- **OBJECT** – Objects have states/attributes and behaviors. Example: A cat has attributes - color, name, age, breed as well as behaviors - napping, meowing, and eating. An object is an *instance* of a class.
- **CLASS** – A class can be defined as a *template/blue print* that describes the behaviors/states that object of its type support.
- **METHODS** – A method is basically a behavior. A class can contain many methods. It is in methods where the logic and activities are carried out, data is manipulated and all the actions are executed. Typically, each method is associated with one distinct behavior that the object belonging to that particular class can perform.
- **INSTANCE VARIABLES** – Each object has its unique set of instance variables. An object's *state* is created by the values assigned to these instance variables. Example: A cat's instance variables could be: orange, Ginger, 4, Persian (the values assigned to each of its attributes color, name, age, and breed.)

Note: Instance variables are also called "*non-static fields*." The word *field* is sometimes used interchangeably with the word *attribute*.

#### Basic Syntax

- **CASE SENSITIVITY** – Java is case sensitive, which means the identifier Hello and hello would have different meaning in Java.
- **CLASS NAMES** – For all class names the first letter should be in upper case. If several words are used to form a name of the class, each inner word's first letter should be in upper case. Example: `class MyFirstJavaClass`
- **METHOD NAMES** – All method names should start with a lower case letter. Like class names, if several words are used to form the name of the method, each inner word's first letter should be in upper case. Example: `public void myMethodName()`
- **PROGRAM FILE NAME** – Name of the program file should exactly match the class name. When saving the file, you should save it using the class name (remember Java is case sensitive) and append '.java' to the end of the name (if the file name and the class name do not match your program will not compile). Example: Assume 'MyFirstJavaProgram' is the class name. Then the file should be saved as 'MyFirstJavaProgram.java'

## Java Identifiers

All Java components require names. Names used for classes, variables and methods are called *identifiers*. There are several rules to remember about identifiers. They are as follows:

- All identifiers should begin with a letter (A to Z or a to z), currency character (\$) or an underscore (\_).
- After the first character identifiers can have any combination of characters.
- A *key word* cannot be used as an identifier. (See below for what the Java key words are.)
- Most importantly identifiers are case sensitive.

Examples of legal identifiers: age, \$salary, \_value, \_\_1\_value, accountNumber

Examples of illegal identifiers: 123abc, -salary

## Java Keywords/Reserved Words

Here is a list of keywords (also known as *reserved words*) in the Java programming language. You *cannot* use any of the following as constants, variables, or other identifiers in your programs. The keywords `const` and `goto` are reserved, even though they are not currently used. `true`, `false`, and `null` might seem like keywords, but they are actually literals; you cannot use them as identifiers in your programs.

<code>abstract</code>	<code>continue</code>	<code>for</code>	<code>new</code>	<code>switch</code>
<code>assert***</code>	<code>default</code>	<code>goto*</code>	<code>package</code>	<code>synchronized</code>
<code>boolean</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>this</code>
<code>break</code>	<code>double</code>	<code>implements</code>	<code>protected</code>	<code>throw</code>
<code>byte</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>throws</code>
<code>case</code>	<code>enum****</code>	<code>instanceof</code>	<code>return</code>	<code>transient</code>
<code>catch</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>try</code>
<code>char</code>	<code>final</code>	<code>interface</code>	<code>static</code>	<code>void</code>
<code>class</code>	<code>finally</code>	<code>long</code>	<code>strictfp**</code>	<code>volatile</code>
<code>const*</code>	<code>float</code>	<code>native</code>	<code>super</code>	<code>while</code>

\* not used

\*\* added in 1.2

\*\*\* added in 1.4

\*\*\*\* added in 5.0