Java Notes Chapter 03 – Methods, Classes, & Objects

**METHODS:**

\*\*Main() method is always executed first & is the starting point for any application

* Header:
* Access specifier: public, static/non-static
* Return type
* Identifier: lowercase/camel casing
* Parentheses ()
* Body:
* Statements: carry out the work
* Return statements: end the method

\*Instance methods are non-static methods used with object instantiations\*

* Static: used for class wide methods, not for methods that “belong” to objects
* Non-static: associate methods with individual objects; behaves uniquely for each -- Returns a different value for each object

**PARAMETERS:**

* Within the method parentheses
* Parameter type &
* Identifier (local name)

\*A **parameter** accepted by a method can be any type- Primitive, built-in, or class type

\***Arguments** are used in method calls, they are info passed to parameters in method headers

\*Parameters can be in any order but the arguments must follow the same order

\*Chaining Method calls\* & \*Fully Qualified Identifier

**WITH OBJECT ORIENTD PROGRAMMING, YOU CREATE CLASSES TO:**

* -Simply instantiate objects 🡪 no main() method (used by other classes/apps)
* Run as applications 🡪 has main() method

**CLASSES:**

* Header:
* Optional access specifier: public- accessible by all objects
* Keyword ‘class’
* Identifier: uppercase

Extended Classes: public classes used as the bases for other classes

\*Classes become new data types when you create them

* Body:
* Methods
* **Data** **fields:** (data components)
* Variables you declare within a class, but outside any method
* Usually private & non-static
* Static field: one value shared by all objects
* Non-static: each object has its own value

**OBJECTS:**

* Declare object 🡪 use class name as the object type

Empoyee someEmployee;

* Allocate memory for the object 🡪 use the new operator to create object

someEmployee = new Employee();

\*The name of the object (identifier) becomes a reference to the object

* Method Access:
* Object.Method

**DATA HIDING USING ENCAPSULATION**

* Data fields should usually be private and a client application should only be able to access them through the public interfaces (class’s public methods)

**CONSTRUCTORS:**

* Establishes objects
* Write your own, or Java writes one automatically for you
* Constructor name = Class name
* Cannot have a return type, not even void; Usually public

**Default Constructor:**

* Employee() – a method that constructs an Employee object

Employee employee = new Employee(); 🡪 calls the default class constructor

* Provides default initial values for object’s data fields:
* Numeric fields: 0
* Character fields: Unicode ‘\u0000’
* Boolean fields: false
* Object Reference fields (i.e. String): null