

**IT1712**

Classes and Methods

# Class Methods and Definitions

* The ***void*** keyword denotes that the method is not used to return a value.
* **Parameter** is the list of variables in a method declaration while **argument** is the actual value that is passed when the method is invoked.
* The data type of the *return* value must match the declared *return* type. *(For instance, you cannot return an integer value from a method declared to return a Boolean.)*
* **Instance variables** are variables declared outside the method, constructor, or any block.
* A **method** is a collection of statements that are grouped together to perform an operation.
* The ***this*** keyword represents the object’s name receiving the method call within a method definition.
* **Local variables** are variables declared within a method definition. These variables are only visible to the methods in which they are declared. They are not accessible from the rest of the class.

# Information Hiding and Encapsulation

* **Information hiding** is the mechanism for restricting access to some of the object’s components.
* Advantages of Information Hiding (Bernstein, 2015):
  + Makes components easier to understand/use
  + Simplifies modification and repair
  + Facilitates re-use
* If an instance variable is **public**, there are no restriction on where you can use its name.
* If an instance variable is **private**, its name cannot be used to access it outside of the class definition.
* If the method is **public**, you can invoke it anywhere without restriction.
* If a method definition is **private**, the method cannot be invoked within the definitions of methods in its class.
* The **accessor** is a public method that returns data from a private instance variable
* The **mutator** is a public method that changes the data stored in one (1) or more private instance variables.

# Encapsulation:

* + It is the process of combining data and actions into a single item.
  + It groups instance variables and methods into a class.
  + It hides implementation details.
* The **UML Class Diagram** describes the structure of a class by displaying the class name (first row), variables (second row), and methods (third row).

# Objects and References

* A variable of a class type contains the memory address of the object named by the variable. The object itself is not stored in the variable but rather in some location in memory.
* The address of the object’s memory location is called a **reference** to the object.
* Class types are **reference types** whose variables hold references or memory addresses.
* Instead of using == for comparison, use the ***equals()*** method.

# 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 Edition*. Boston: Course Technology, Cengage Learning

Savitch, W. (2014). *Java: An introduction to problem solving and programming, 7th Edition*. California: Pearson Education, Inc.

**02 Handout 1 *\*Property of STI***

**Page 1 of 1**