Programming 2

Tutorial 11

Exercise 1: (Required)

Create a file named `ArrayListOperations.java` using ArrayList and perform the following tasks:

- Add an integer to the ArrayList
- Add a floating-point number to the ArrayList
- Add a boolean value to the ArrayList
- Add a string to the ArrayList
- Print out the 4 values from the ArrayList to the screen

Exercise 2: (Required):

Create file `ArrayListIntegerLoop.java` using `ArrayList<>` and perform the following tasks:

- Use a Generic ArrayList of type Integer (`ArrayList<Integer> myArray = new ArrayList<Integer>();`)
- Use a loop to input numbers from 1 to 10 into the myArray array
- Use a loop to display numbers from 1 to 10 from the myArray array.

Exercise 3: (Required)

In this exercise, your objective is to design and implement a Java class named **Product**. The Product class will contain attrubutes such as name and price. **Product** should implement the **Serializable** interface in Java to enable your class instances to be converted into a byte stream, which can be transmitted over a network or stored externally without losing the object's structure.

Complete the following tasks:

- Define a comprehensive table of domain constraints applicable to the attributes of the **Product** class.
- Design and implement the **Product** class within the "**tut11.product**" package.

Exercise 4 (Required)

In this exercise, your objective is to implement a Java class named **DAO** (Data Access Object) abstract class with CRUD (Create, Read, Update, Delete) methods.

- The **DAO** class should be defined as **abstract** and **generic** with a type parameter **Entity**, representing the type of object we want to operate on.
- The **DAO** class should contain a list of objects of type **Entity** to store data.
- The **DAO** class should have an **add method** to add an object of type Entity to the list.
- The **DAO** class should have a **remove method** to remove an object of type Entity from the list.
- The **DAO** class should have **two abstract methods update and find**. You need to implement these methods in specific concrete subclasses. The **update method** is used to update information of an object already in the list, while **the find method** is used to search and return an object from the list based on a unique key (**Serializable id).**
- The **DAO** class should have a getList method to return the list of objects of type Entity.

Notes:

- ✓ Students should ensure that the **update** and **find methods** are implemented appropriately to work with the specific type of Entity they are working with.
- ✓ Students need to pay attention to the use of generics to ensure flexibility and reusability of the **DAO** class.
- ✓ Students should handle and manage exceptions properly in the methods of the **DAO** class.

Exercise 5: (Required)

In this exercise, you create a class **ProductDAO** inheriting from the DAO class (in exercise 4) and write code to implement the abstract methods.

- The **ProductDAO** class should extend the DAO class with a type parameter of **Product**, indicating that it operates on objects of type **Product**.
- The **update method** should **iterate** through the list of products and update the product with the same name as the provided entity.
- The **find method** should **iterate** through the list of products and return the product with the same name **as the provided ID**. If no product is found, it should return null.
- Write a test program to validate the correctness of the implemented functionality.

Note:

- ✓ Ensure that the methods are implemented correctly to work with the Product class and handle any edge cases appropriately.