**PROGRAMMING PATTERNS**

420-301-VA

#### **LAB-1**

### Instructions

* Labs’ answers must be demoed to the teacher in class.
* Labs are individual works.
* You are supposed to work on this LAB during the session hours otherwise you risk falling behind.
* First, try to solve each problem without assistance except asking clarifying questions, preferably from the teacher.
* Do not look at solutions or get assistance from colleagues, online resources, or AI before spending considerable time trying to do the work from scratch yourself, some problems may require hours to solve, and you may need to look away work on another assignment and get back to the problem with a fresh eye that is normal and differs from individual to individual.
* The lab work is based on the theory sessions and a direct implementation of the concepts. Make sure you understand the concepts and can do the examples, or follow the logic, discussed in class and identify what the lab is asking you to do relative to the discussed concepts.
* Feel free to ask more questions to the teacher or look at different examples that are not the solutions to the lab from any other resource.
* Note that the due date is different for each section.
* Note that if you could not work on a lab by the deadline, you should still work on it, submit it and discuss the case with the teacher.

## Data Structure Definition

1. Using the definitions writing technique discussed in class write the definition of a data structure (research information if needed):

|  |
| --- |
| **What are a Two dimensional array’s characteristics?**   * **Nature**: it is a data structure concept that stores data in a grid like format with rows and columns * **Purpose**: organize data that naturally form a grid * **Composition/elements**: it contains rows (horizontal collection of data) and columns (vertical collection of data) * **Usage**: to store data, it could be used for image quality, spreadsheet data, database, * **Properties**: contains the same data type, it has a fixed size, * **What it is not**: it is not a list of lists * **Examples (give a specific example of a group of data that could be stored in the array)**: chessboard |

|  |
| --- |
| **Write a full comprehensive statement defining a data structure putting all the characteristics listed above together in good English writing style:**  A two-dimensional array is a data structure concept that stores data in a grid like format. It organizes data in a way that naturally forms a table. It contains a horizontal collection of data and a vertical collection of data. It is used for data storage such as spreadsheet data, database, image quality, etc. The rows and columns must contain the same data type, and it has a fixed size that cannot be dynamically stretched. Each row and column use an index to keep track of the specific element that we wish to access. A two-dimensional array is not a list of lists. For example, a good example that represents a two-dimensional array is a chessboard which contains rows and columns. |

## Lambda Expressions

Consider the following defined interface:

|  |
| --- |
| **interface Enumerator {**  **public void enumerate(ArrayList<String> arrayList);**  **}** |

We assume that the expected functionality of the ‘enumerate’ method is to print out the values of the ArrayList one by one.  
Implement the lambda expression that would be used to call the following method:

|  |
| --- |
| **public static void enumerate(ArrayList<String> arrayList, Enumerator e){**  **e.enumerate(arrayList);**  **}** |

Implement a fully working example with a sample ArrayList.

## 

## Arrays Manipulation

### Extending an Array

Consider the class below that represents a simple list of integer elements, whereas it holds a set of values in an array.

|  |
| --- |
| **class CustomArrayList{**  **int [] elements;**  **CustomArrayList(int [] data){**  **elements = data;**  **}**  **void add(int e){**    **}**  **int get(int index){**  **}**  **}** |

The constructor **CustomArrayList(int [] data)** takes an array of integers and initializes the elements arrays.

* Implement the **add(int e)** method that is supposed to take an integer and add it to the elements array.
* Implement the **get(int index)**method that is supposed to take an index value and to return the value of the element at that index.
* Implement a test code that populates the list with elements and then outputs all the elements from the list.