# Object Oriented Programming Lab

**Lab 09** **Marks 10**

## Instructions

Work on this homework individually. Absolutely NO collaboration is allowed. Any traces of plagiarism would result in ZERO marks in this homework and possible disciplinary action. Task should be coded in **C++**. You are strictly ***NOT ALLOWED*** to include any additional data-members/functions/constructors in your class. Write the ***main*** function first and keep testing the functionality of each function once created.

## Due Date

Upload the solution *(source code **.cpp** file only)* labeled with your complete roll number in capital letters e.g., BITF21M000 till 05:00PM Friday, April 28, 2023, in course's *Google* classroom.

## ADT: Generic Array

Write a **generic class** named **Array** that can hold an array of any valid data type *(int, float, char etc…)*.

**1.** The class should have following **two private data members**.

   **a.** A *generic pointer* called **data** that holds an array of given data type allocated dynamically according to the specified size.
   **b.** An *integer* called **size** represents the number of elements in the array.

**2.** Implement the following **constructors** and a **destructor**:

   **a.** A **default constructor** that allocates an array of **size 5** and initializes it to the so-called "empty array," i.e., an array whose array representation **contains all zeroes**.

   **b.** A **constructor** that accepts an **integer** as argument to represent the **size of an array** and initializes it to the so-called "empty array," i.e., an array whose array representation **contains all zeroes**.

   **c.** A **copy constructor** initializes an array object with an already existing object.

   **d.** A **destructor** to **free any memory resources** occupied by the **array** object.

**3.** Implement the following **non-static member functions** and **operators**:

- **getSize** returns the size of array.
- **setElement** that **inserts** a new element **k** at index **i** (both passed as argument) into an **array**, if possible, otherwise give an appropriate error message.
- **countElement** accepts a **key** as argument and **counts and returns** the **total occurrences** of it in an array.
- **Assignment (=)**: Copies the data of the right-hand-side object to the left-hand-side object. If the size of the left-hand-side object is different from the right-hand-side object, reallocate the memory for the left-hand-side object based on the size of the right-hand-side object and then copy the data. Don't forget to update the size of the left-hand-side object.
- **getSubArray** This member function takes two integer parameters **start_index** and **end_index** as arguments and returns a new Collection that contains all the values in the left-hand-side object from **start_index** to **end_index**, both inclusive. If the requested sub-collection cannot be created, it displays an appropriate error message and returns a Collection object consisting of NULL with its size set to 0.
- **Arithmetic binary (+)** that inserts contents of *right-hand-side* object at the end of *left-hand-side* object and **return** the result. The function should not make any changes in *left/right-hand-side* object.
- **Stream insertion (<<)** to take input from user for the **data** of an **array**.
- **Stream extraction (>>)** to display the contents of **data** on the screen of an **array**.
- **Comparison (==)** that determines whether **two arrays are equal or not**. The operator should return **true** if both the arrays *(left-hand-side and right-hand-side)* are equal, **false** otherwise.
- **Subscript ([])** for non-const objects. If the **index** is out of bounds, it displays an appropriate error message and returns $-1$, Without exiting the program.
- **Subscript ([])** for const objects. If the **index** is out of bounds, it displays an appropriate error message and returns $-1$, Without exiting the program.

**4.** Once you have written the class, write the **main** function and test its functionality by creating some objects of **Array**.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺