

# Object Oriented Programming

## Home Work 09

**Marks 10**

### Instructions

Work on this home work individually. **Absolutely NO collaboration is allowed. Any traces of plagiarism would result in a ZERO marks in this homework and possible disciplinary action.** Tasks should be coded in **C++**.

### Due Date

Paste the solution of the problem (source code .cpp file only) labeled with your complete **roll number** in **SEM – HW 09** and **SEA – HW 09** folders for **SE Morning** and **SE Afternoon** sections respectively on **Friday, May 13, 2016** before **05:00 PM**. These folders are available at **\\printsrv\Teacher Data\Umair Babar\Students**.

### ADT: Generic Collection

Write a generic class named **Collection** for which each object can hold **elements of any valid data type** and **zero** as default value.

- The class should have following **two private data members**.
  - A **pointer** named **data** that holds an **array of any valid data type** allocated dynamically according to the specified **size**.
  - An **integer** named **size** that holds the **size of the array** (amount of memory allocated to data).
- Provide the implementation of following **constructors** and a **destructor**
  - A **default constructor** which creates an array of **five elements** and initializes it to the so-called "empty collection," i.e., a collection whose array representation **contains all zeroes**.
  - A **constructor** which accepts an **integer** as argument to represent the **size of an array** and initializes it to the so-called "empty collection," i.e., a collection whose array representation **contains all zeroes**.
  - A **copy constructor** to initialize a collection object with already existing object.
  - A **destructor** to free any memory resources occupied by the collection object.
- Provide following **member functions** for the common operations
  - getSize** returns the **size** of collection.
  - setElement** that **inserts** a new element **k** at index **i** (both passed as argument) into a **collection**, if possible, otherwise give an appropriate error message.
  - findElement** accepts an element **key** as argument and return **true**, if the key element exist in the collection, **false** otherwise if the key does not exist.
  - countElement** accepts an element **key** as argument and **count and return** the **total occurrences** of it in a collection, **-1** otherwise, if the key does not exist.
- Provide the implementation of following **overloaded operators**
  - Assignment (=)** which copies the data of one object to another. The assignment should be done even if the sizes of both objects are not same and avoid self-assignment.
  - Stream insertion (<<)** to display the contents of **data** on the screen of a **collection**.
  - Stream extraction (>>)** to take input from user for the **data** of a **collection**.
  - Arithmetic assignment (+) binary** which perform the addition of two collections (left hand side and right hand side) if possible and **return** the result.
  - Comparison (==)** that determines whether **two collections are equal or not**. The operator should returns **true** if both the collections are equal, **false** otherwise.
  - Subscript ([])** for both **lvalue** and **rvalue** of **non-const** objects
  - Subscript ([])** for **rvalue** of **const** objects
  - Unary minus (-)** return **true** if all the elements of a collection are **non-zeroes**, **false** otherwise.
  - Unary Not (!)** assigns zero to all the elements of the object i.e. convert the collection into its "empty" form.
  - Function (())** receives two parameters as argument **start\_index** and **end\_index** and **return** the **new sub collection** which contains all the values exist in the left hand side object from **start\_index** to **end\_index** both inclusive, if possible.
- Once you have written the class, write **main** function and test its functionality by creating some objects of **Collection** for **integer**, **float** and **char** data types.

**NOTE:** - No submission will be accepted after the due date and time.

**B E S T O F U C X**