

# National University of Computer and Emerging Sciences



## Lab Manual 06 Object Oriented Programming

Course Instructor	Mr. Usama Hassan
Lab Instructor (s)	Ms. Fariha Maqbool Mr. Sohaib Ahmad
Section	BSE-2C
Semester	Spring 2023

Department of Computer Science  
FAST-NU, Lahore, Pakistan

## Objectives

After performing this lab, students shall be able to understand:

- ✓ Develop better understanding of classes and objects.
- ✓ Dynamic memory allocation inside the class.
- ✓ Static data members and static member functions.

### **TASK-1:**

In this task, you have to create a 'Set' ADT, which will implement the basic operations of Set concept in mathematics. The data members and operations needed for this ADT are given below.

#### **Private Data Members:**

- int \*data; /\* pointer to an array of type integers which will be treated as set \*/
- int capacity; /\* maximum possible number of elements that can be stored in the Set \*/
- int noOfElements; /\* number of elements in the Set \*/
- static int count; // keep the count of objects of the class created in the main function. Initialize this data member outside the class with 0 and update its value by 1 in the constructors and similarly decrease the value by 1 in the destructor.

#### **Supported public Operations:**

The class 'Set' should support the following operations:

1. Set( int cap = 0 ); /\* default parameterized constructor \*/

Sets 'cap' to 'capacity' and initializes rest of the data members accordingly. If user skips the value in the parameters list then sets the cap to default value. Initially noOfElements in the set were 0 you need to update this value after every insertion in the set.

2. Set( const Set & ref) /\* copy constructor to implement deep copy \*/
3. void reSize (int newcapacity ); /\*resize the set to new capacity. Make sure that elements in old set should be preserved in the new set. Since there is a possibility that the newcapacity is either smaller or greater than the original capacity so you need to handle both the cases i.e., shrink and growth. \*/
4. static int getObjCount() const /\*this function should return the static data member "count" \*/
5. ~Set() /\* destructor to handle the issue of memory leak and dangling pointer \*/
6. void insert (int element); /\* since we are maintaining a set which is a collection of distinct elements. So, before insertion you need to check whether the value is distinct, or it is already in the Set. Store the value only if it is distinct and update the count of **noOfElements**.
7. void remove (int element); /\* search whether the provided element exist in the set. If yes, then you need to remove the received element from the Set. You are already maintaining noOfElements so swap the element with the last element of the Set and then store -1 on that index and don't forget to decrease the count of noOfElements. \*/

If the following data is there in the Set (capacity= 5, noOfElements = 4)

4	7	13	8	
---	---	----	---	--

Assume that element to be deleted is 7 and the element is present in the Set then (**swap with last element of the set**)

4	8	13	7	
---	---	----	---	--

To remove the element you need to replace the element with (-1) as shown below and decrease the noOfElements so that (capacity = 5 and noOfElements = 3)

4	8	13	-1	
---	---	----	----	--

8. int getCardinality() const; /\*returns the count of elements in the set \*/.
9. Set calcUnion ( const Set & s2 ) const; /\*returns a new Set object which contains the union of 's2' set and calling object set.\*/
10. Set calcSymmetricDifference ( const Set & s2 ) const; /\*returns a new Set object which contains the symmetric Difference of 's2' set and calling object set. Where symmetric difference is:

$$\text{Symmetric Difference} = (A \cup B) - (A \cap B)$$

11. Set calcDifference(const Set & s2) const; /\*returns a new Set object which contains the intersection of 's2' set and calling object set.
12. bool isMember ( int val ) const; /\*returns true if 'val' is member of the set otherwise return false.
13. int isSubSet ( const Set & s2 ) const; /\*returns "1" if s2 is proper subset of calling object, return "2" if s2 is improper subset of calling object otherwise return 0.
14. void print() const /\* This function should print the elements of the set on console. \*/

**Write the main function to test the functionality of each member function of the class. You need to create at-least 2 to 3 objects of the class.**