# Object Oriented Programming : I

**Problem 1:**

Create a class ‘item’ that contains the following data fields:

1. item_id(integer)
2. Quantity(integer)
3. item Label(character array of size 10)

and a suitable constructor .

Create another class ‘Stack’ which contains an array of such items and data member ‘top’ and ‘size’. In the constructor, initialize the top to -1.

Create methods to push( ), pop( ) , is_empty( ) and display( ) the stack. Note that the push method will call the constructor of the ‘item’ class.

**Problem 2:**

Augment the above program to include the following two methods in the stack class:

a. A method to sort the elements in the stack only using the class ‘Stack’ methods defined previously (push(),pop(), is_empty()) such that the item with the largest quantity appears at the top.

b. A method that removes the middle element in the stack (using only the standard stack methods defined in question 1).

For both questions, create a main method. Create an object for Stack and verify the correctness of the methods you’ve defined by invoking them appropriately.

**Problem 3:**

Write an object oriented program for manipulating strings by creating your own definitions of operators +, <=, = =, != and subscript operator [ ] using friend functions as follows.

i) The declaration for the String class looks as follows:

```
Class String{
char *value;
int length;
public:
Methods:-
Constructors:
string ( ) { length=0; value=0);
string (const char *s) ;
```

```
 string ((const string &s);
 //operators
 friend string operator + (const string &s, const string &t);
 friend int operator >= (const string &s, const string &t);
 friend int operator== (const string &s, const string &t);
 friend int operator != (const string &s, const string &t);
 friend char operator [ ] (int value); //(Unary)
 friend void display(const string s);
};
```

ii) Write definition of each function and use given operators for manipulating strings. Take different user inputs for string.

**Problem 4:** Define a custom class LinkedList with appropriate constructors, destructors, data members and functions which contain ComplexInteger objects as elements. Also 2 additional friend functions to display the contents of the LinkedList and Search for the Specified object and returning boolean value (if present or not). Also perform the following operations via its Objects by overloading appropriate operators.

i) + : Adds a ComplexInteger object at the end of the list.( when called with a ComplexInteger Object).
And
Appends another list at the end (when called and passed another List Object).
ii) - : Deletes a ComplexInteger object from the end of the list.
iii) * : Deletes the element at the specified no. from the beginning. (if it exists)
Say list is :
2+3i → 5+2i → 7+9i
After calling :
list*2;
The list becomes :
2+3i → 7+9i
(element no.2 deleted )
iv) = : Assigns a list Object to another list reference (when called with 2 LinkedList objects).
And
Creates a fresh List from an array of ComplexInteger Objects. (when called from a list object and passed an array of ComplexInteger objects).
v) << : Left Shifts the contents of the list specified no. of times.
Ex: list<<2;
vi) >> : Right Shifts the contents of the list specified no. of times.
Ex: list>>3;