

# Data Structures and Fundamentals of Programming

## Problem #1

In C++ implement a **generic** class, called `Stack<T>`, that uses a **single-linked list** implementation. This should implement the stack ADT. It should be generic on the type of the data to be stored. Give all class definitions and implement the following for `Stack`:

- Default constructor
- Destructor
- Copy-constructor
- Assignment operator – using standard copy semantics
- `push(T)` – takes an parameter of type T and adds it to the stack
- `T pop()` – removes a node from the stack

Note: Your implementation can **NOT** use STL or any other libraries (standard or otherwise).

## Problem #2

In C++ implement a **binary search tree** ADT that uses **dynamic memory allocation**. Make it a simple tree of integers. Along with the class definition(s), you **must** implement the following methods for the class:

- Default constructor
- Destructor
- Copy-constructor
- `insert` which takes a parameter of type integer and creates a new node that is added to the tree in the correct position based on the rules of a binary search tree.

Note: Your implementation can **NOT** use STL or any other libraries (standard or otherwise).

## Problem #3

In C++ implement a **generic function** that removes all duplicates in a `std::vector` passed in as reference. For example, if the vector passed in was `[1, 2, 3, 4, 2, 3, 3]` it would be modified to be `[1, 2, 3, 4]`. It should be generic on the type of the vector. What constraints are on the type that this function can be instantiated with?