

Data Structures and Fundamentals of Programming

Problem 1

In C++ implement a **generic** class, called `Stack<T>`, that uses a **single-linked list** implementation. It must implement the stack ADT. It must 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
- Swap – constant time swap (i.e., run time is not dependent on size of stack)
- Overload the assignment operator – using standard copy semantics
- `push(T)` – takes a parameter of type T and adds it to the stack
- `T pop()` – removes a node from the stack

You can **NOT** use STL or any other predefined library or built in types (such as `std::string`).

Problem 2

In C++ implement a **generic** class, called `Queue<T>`, that uses a **fixed sized array** implementation. This must implement the queue ADT. Implement the queue using a **circular array**. It must be generic on the type of the data to be stored. The implementation **must** utilize the entire fixed sized array in storing items. Give all class definitions and implement the following for `Queue`:

- Default constructor
- `push(T)` – takes an parameter of type T and adds it to the queue
- `T pop()` – removes a item from the queue
- `isEmpty()` – returns true when the queue is empty.
- `isFull()` – returns true when the queue is full.

You can **NOT** use STL or any other predefined library or built in types (such as `std::string`).

Problem 3

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

- Default constructor
- Destructor
- Copy-constructor – **must** be recursive or use a recursive method to copy the nodes.
- `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.

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