Virtual University	**Assignment No. 1** **Semester: Fall 2016** **CS301 – Data Structures**	**Total Marks: 20** **Due Date:** **November 16, 2016**

Instructions

Please read the following instructions carefully before submitting assignment:

It should be clear that your assignment will not get any credit if:

- **Assignment is submitted after due date.**
- **Submitted assignment does not open or file is corrupt.**
- **Assignment is copied (From internet/ to from students).**

For clarity and simplicity, You are required to Upload/Submit only ONE **.CPP** file if needed.
Keep all your files (word and .cpp) in a folder and make its zipped file to upload on LMS.

Note: Use ONLY Dev-C++ IDE for writing and executing your code.

Objectives:

The purpose of this assignment is to make you familiar with following topics:

- Singly Linked List, Double Linked List and Circular Linked List
- Infix to Postfix
- C++ Templates
- Implementation of Stack
- Function Call Stack

Assignment Submission Instructions

You have to submit only word file on the Assignments interface of CS301 at VULMS. ***Assignment submitted in any other format will not be accepted and will be graded zero marks.***

Assignment

Question 1: [10 marks]

Given is a code of abstract class declaration for a list (ADT) in Figure 1(a).

```
template <typename Elem> class List { // List ADT
private:
  void operator =(const List&) {}      // Protect assignment
  List(const List&) {}           // Protect copy constructor

public:
  List() {}          // Default constructor
  virtual ~List() {} // Base destructor

  // Reinitialize the list.  The client is responsible for
  // reclaiming the storage used by the list elements.
  virtual void clear() = 0;

  // Insert an element at the front of the right partition.
  // Return true if the insert was successful, false otherwise.
  virtual bool insert(const Elem&) = 0;

  // Append an element at the end of the right partition.
  // Return true if the append was successful, false otherwise.
  virtual bool append(const Elem&) = 0;

  // Remove and return the first element of right partition.
  virtual Elem remove() = 0;

  // Place fence at list start, making left partition empty.
  virtual void movetoStart() = 0;

  // Place fence at list end, making right partition empty.
  virtual void movetoEnd() = 0;

  // Move fence one step left; no change if at beginning.
  virtual void prev() = 0;

  // Move fence one step right; no change if already at end.
  virtual void next() = 0;

  // Return length of left or right partition, respectively.
  virtual int leftLength() const = 0;
  virtual int rightLength() const = 0;

  // Set fence so that left partition has "pos" elements.
  virtual void movetoPos(int pos) = 0;

  // Return the first element of the right partition.
  virtual const Elem& getValue() const = 0;
};
```

Figure 1(a)

a) Assume a list has the following configuration:

< | 2, 23, 15, 5, 9 >.

Write only Method calls (Name of function) using the List ADT of Figure 1(a) to delete the element with value 15.
e.g To move the pointer to end of the list We will use function **movetoEnd( ).** No need to write function implementation.

Note: Here the '|' symbol represents 'fence' like pointer that indicates the current position of pointer in list.

b) Show the list configuration resulting from each series of list operations using the List ADT of Figure 1(a) Assume that list L1 is empty at the beginning of each series. Show where the fence is in the list.

```
L1.append(10);
L1.append(20);
L1.append(15);
```

Question 2: [10 marks]

Given is an infix expression:

A + B * C / (D-B)

Write a complete program in C++ to convert the above infix expression into postfix notation.

Deadline:

Your assignment must be uploaded on or before November 16, 2016. We shall not accept your solution through email after the due date.