**Software Design Document**

**For**

**Programming Assignment 1**

**October 13, 2015**

Prepared By

Poornima Byre Gowda

[pg0018@uah.edu](mailto:pg0018@uah.edu)

**Table of Contents**

1. System overview............................................................................................................ 3
2. Referenced Documents …………………………………………………………………………………………… 3
3. Architectural Design……………………………………………………………………………………………….. 3

3.1 Concepts of execution ………………………………………………………..…………………. 3

3.2 Code outline………………………………………………………………………………………….. 4

4.0 Detailed Design

4.1 Source file: ……………………………………………………………………………………………. 4

**Software Design Document**

**Programming Assignment 2**

**1.0 System Overview**  
  
 This program will demonstrate a decoder using a special implementation of a stack as a linked structure.

**2.0 Referenced Documents**

* Programming Assignment 1 Statement of Work.
* Dale, Nell and Teague, David, *C++ Plus Data Structures* 2nd ed. 2001.

**3.0 Architectural Design**

**3.1 Concept of Execution**

This program will demonstrate a decoder using a special implementation of a stack as a linked structure. An encoded message will be parsed and decoded using the stack.  
  
**3.2 Abstract Data Type**

This program will be implemented using the abstract datatype which a Structure named as StackNode. This abstract data type is accessed by the interface class called Decoder.

The purpose of using structure StackNode here is to create the character datatype ‘m\_cCh’ and a Structure pointer m\_pNext and utilize the variables using Class.

**3.3 Code Outline**

The classes and headers used in this program are listed below:

**Class: Decoder**

**Files: Decoder.h and Decoder.cpp**

**Decoder.cpp**:

This file will create the instance of the class to access the private Structure arrays declared in Class. The input values are provided to the structure arrays and printed using functions. The functions used in the program are listed below.

**Constructor: Decoder ();**

**Destructor: ~ Decoder ();**

**Functions:**

void Push(char ch)

char Pop()

void Decode(char \*encMsg, char \*decMsg);

**4.0 Detailed Design**

**4.1 Source File: Decoder.h and Decoder.cpp**

    **4.1.1 Constructor: Decoder ()**

**4.1.1.1 Purpose**  
The constructor shall set the structure pointer to NULL.

**4.1.1.2 Arguments**  
No arguments.  
**4.1.1.3 Return value**  
There is no return value.   
**4.1.1.4 Outline in pseudocode**

It initializes the structure pointer to NULL.

**4.1.1.5 Traceability**

It fulfils the requirement of 2.2.2.

**4.1.2 Destructor: ~Decoder()**

**4.1.2.1 Purpose**  
This will deallocate memory of the stack by deleting the instances of StackNode.

**4.1.2.2 Arguments**  
No arguments.  
**4.1.2.3 Return value**  
There is no return value.   
**4.1.2.4 Outline in pseudocode**  
No

**4.1.2.5 Traceability**  
The destructor shall remove and delete any instances of StackNode left in the stack. It fulfils the requirement of 2.2.2.

**4.1.3 Function: void Push(char ch)**

**4.1.3.1 Purpose**  
This function dynamically create a structure of type **StackNode** store the character in the structure and "Push" the structure onto the stack.

**4.1.3.2 Arguments**  
The arguments received are the following:

ch (character value that has )

**4.1.3.3 Return value**  
There is no return value.   
**4.1.3.4 Function outline in pseudocode**  
dynamically create the Structure through struct pointer.

Store the character in the struct character element.

Assign the address of next structure pointer to the Top stack pointer.

Assign the newly pushed structure as the Top of the stack

**4.1.3.5 Traceability**  
It fulfils the requirement of 2.2.3.

**4.1.4 Function: Char Pop()**

**4.1.4.1 Purpose**  
Function will Remove the top node from the stack, copy the character, from this node, delete the node and return the character. This function shall be private.

**4.1.4.2 Arguments**  
no arguements

**4.1.4.3 Return value**  
Returns character value.  
**4.1.5.4 Function outline in pseudocode** Create the Temporary Struct Pointer.

Check the Top of the Stack is NULL

Assign the m\_pTop structure values to the Temporary struct pointer.

Assign the m\_pTop to the m\_pTop next pointer.

Delete the tempoarary pointer

**4.1.4.5 Traceability**  
It fulfils the 2.2.4 requirement in the requirement sheet. This function pop the top element in stack.

**4.1.5 Function: void Decode(char \*encMsg, char \*decMsg)**

**4.1.5.1 Purpose**  
This Function will parse and decode the message stored in the character array encMsg using the stack functions and return the decoded message in the character array decMsg.

**4.1.5.2 Arguments**

The arguments received are the following:

Char \*encMsg.

Char \*decMsg.  
**4.1.5.3 Return value**  
There is no return value.  
**4.1.5.4 Function outline in pseudocode**

If EncMsg first array pointer value has ‘1’

Push the next array element

Increament the array pointer by ‘2’

If else EncMsg array value has ‘2’

Pop the Stack ‘n’ times, ‘n’ is next array element value.

Increament the array pointer by ‘2’

Finally Pop out the decoded data stored in Stack.

**4.1.5.5 Traceability**

2.2.5 Requirement in requirement sheet is fulfilled. This decodes the encoded data correctly.