**Lab Exercises 5: Stack**

**LEARNING OUTCOMES**

* Understand the problem related to stack
* Able to implement stack function using linked list

### PROBLEM : PUSH & POP IN STACK USING LINKED LIST

**LAB EXERCISE 5.1**

Estimate Time**: 2 Hour*s***

**Tasks:**

Based on the main program in Figure 5.1, write three functions *push ( ), pop ( )* and *display( )* as anillustration of stack using linked list. *push( )* is a function to add a new node at the beginning of stack. *pop( )* is a function to delete a node at the beginning of stack. The *pop ( )* function must be able to prevent any pop operations if the stack is empty. *headptr* is a pointer which contains the address of first node. *headptr* is used to locate the top element in the stack, which is the first node in the linked list.

struct node

{ int number;

struct node \*ptrnext; };

struct node \*headptr, \*newptr, \*currentptr; //pointer that can //point to a node

void main()

{ char ch;

headptr=(struct node \*)NULL;

do

{ printf("\n\n[-------Stack Menu-------]");

printf("\nA - Push Stack");

printf("\nB - Pop Stack");

printf("\nD - Display Stack Content");

printf("\nX - Exit Menu");

printf("\nPlease enter choice: ");

scanf(" %c",&ch);

ch=toupper(ch);

switch(ch)

{

case 'A':push();break;

case 'B':pop();break;

case 'D':display();break;

case 'X': printf("\nExit stack menu"); break;

default: printf("\nInvalid entry. Please select A, B, D, X\n");

}

}while (ch!='X');

getch();

return;

}

Figure 5.1: Main Program