**Debre Markos University**

**Department of Information Technology**

**Data Structure and Algorithm Lab Manual**

**Lab 3: Structure and singly linked list**

**Tools used:** Quincy 2005v.1.3 editor

**Objective:**

Students should be able to know:

- Structure and the difference between array and structure.

- Singly linked lists and operations like insertion and deletion done on them

- How can we implement structures and singly linked list operations

**Source code for structure declaration, creating**

**object and accessing members of structure**

#include<iostream.h>

struct student

{

char name [20];

char Idd [10];

int age;

};

int main ()

{

student stud1;

cout<<"Enter name of student: ";

cin >>stud1.name;

cout<<"Enter id number: ";

cin>>stud1.Idd;

cout<<"Enter age of student: ";

cin>>stud1.age;

cout<<"\n\t Name: "<<stud1.name;

cout<<"\n\t ID Number: "<<stud1.Idd;

cout<<"\n\t Age :"<< stud1.age;

}

**Source code for showing how structure**

**objects/variables can be initialized.**

#include<iostream.h>

struct student

{

char name [25];

char Id [10];

char Department [20];

int age;

};

int main ( )

{

// creating and initializing structure variables

student stud1 = {"Yosef", "R/1035/07", "IT", 25};

student stud2;// creating another structure variable

stud2 = stud1;

cout<<" Name: "<<stud1.name;

cout<<"\n Id :"<< stud1.Id;

cout<<"\n Department: "<<stud1.Department;

cout<<"\n Age :"<< stud1.age;

cout<<"\n \t Copied data"<<endl;

cout<<" \t Copied Name: "<<stud2.name;

cout<<"\n \t Copied Id :"<< stud2.Id;

cout<<"\n \t Copied Department: "<<stud2.Department;

cout<<"\n \t Copied Age :"<< stud2.age;

}

**Singly linked list operations implementations**

#include<iostream.h>

#include<ctype.h>

struct link

{ int data;

link \*next;

};

link \*root=NULL;

void creatlink(int d);

void display();

void InsertFRONT(int d);

void InsertEND(int d);

void deleteFRONT();

void deleteEND();

void deleteBn(int);

void creatlink(int d)

{

link \*temp;

temp = new link;

temp-> data = d;

temp -> next = root;

root = temp;

}

void InsertFRONT(int a)

{ link \*temp;

temp=new link;

if(root==NULL)

{ root=temp;

temp->data=a;

temp->next=NULL;

}

else

{ temp->data=a;

temp->next=root;

root=temp;

} }

void InsertEND(int a)

{ link \*temp,\*temp2;

temp=new link;

temp->data=a;

temp->next=NULL;

if(root==NULL)

{ root=temp;

}

else

{ temp2=root;

while(temp2->next!=NULL)

{ temp2=temp2->next;

}

temp2->next=temp;

}

}

void deleteFRONT()

{ link \*temp;

temp=new link;

if(root==NULL)

{cout<<"Linked List is empty\n";

}

else

{ temp=root;

root=root->next;

delete temp;

}

}

void deleteEND()

{ link \*temp,\*temp2=NULL;

temp=new link;

if(root==NULL)

{ cout<<" List is empty\n";

}

else

{ temp=root;

while(temp->next!=NULL)

{ temp2=temp;

temp=temp->next;

}

if(temp->next==NULL && temp2==NULL)

{ delete temp;

root=NULL;

}

else

{delete temp;

temp2->next=NULL;

temp=temp2;

}

}

}

void display()

{ link \*t;

if(root==NULL)

cout<<" List is empty.\n";

else

{ t = root;

while(t!= NULL)

{

cout<<t-> data<<" ";

t = t->next;

}}

cout<<endl;

}

int main()

{ int m, value,choice;

char ch;

do

{ cout<<"==================================="<<endl;

cout<<"== What do operation you are going to do? ~~~~"<<endl;

cout<<"~\tEnter 1 for creating list"<<endl;

cout<<"~\tEnter 2 for Inserting at Front of the list"<<endl;

cout<<"~\tEnter 3 for Inserting at the End of the list"<<endl;

cout<<"~\tEnter 4 for Deleting from the Front of the list "<<endl;

cout<<"~\tEnter 5 for Deleting the End of the list"<<endl;

cout<<"~\tEnter 6 for Displaying the list"<<endl;

cout<<"~\tEnter other number for exiting the program"<<endl;

cout<<"==================================="<<endl;

cin>>choice;

switch(choice)

{ case 1:

cout<<"How many element you want to store in the link list: ";

cin>>m;

cout<<"\nEnter elements of the linked list:"<<endl;

for(int i=1;i<=m ; i ++)

{ cout<<"\tEnter element "<<i<<" :";

cin>>value;

creatlink(value);

}

cout<<"\nThe link list elements are: "<<endl<<"\t";

display(); break;

case 2:

cout<<"\nEnter an element to insert at the FRONT:";

int a;

cin>>a;

InsertFRONT(a);

cout<<"\nAfter the element is inserted at the front:"<<endl<<"\t";

display(); break;

case 3:

cout<<"\n Enter an element to insert at the END:";

cin>>a;

InsertEND(a);

cout<<"\n After the element is inserted at the end:"<<endl<<"\t";

display(); break;

case 4:

cout<<"\n Before deleting the front element:"<<endl<<"\t";

display();

cout<<"\n After Deleting at the front of a list:"<<endl<<"\t";

deleteFRONT(); display(); break;

case 5:

cout<<"\n Before deleting the end element:"<<endl<<"\t";

display();

cout<<"\n After deleting end element of a list:"<<endl<<"\t";

deleteEND(); cout<<"\t"; display(); break;

case 6:

cout<<"The linked list looks like(if there is)"<<endl<<"\t";

display(); break;

default:

cout<<"Invalid entry, good bye"<<endl;

}

cout<<"Do you want to continue for another operation [y/n]: ";

cin>>ch;

} while(tolower(ch)!='n');

}

**Page-2**