**Muhammad Ahmed Fraz bhatti**

**332779**

**BSCS-10C**

**Data Structures and Algorithms**

**LAB #3**

**Submission**

Write a C++ program that can

1. Create a simple linked list using function, by inserting nodes at head.

Code:

**Code:**

#include<iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

};

class List {

private:

Node\* head = NULL;

public:

void add\_node(int data\_value) {

Node\* temp = new Node();

temp->data = data\_value;

temp->next = head;

head = temp;

}

void displayList() {

Node\* temp = new Node();

temp = head;

while (temp != NULL) {

cout << temp->data << " ";

temp = temp->next;

}

}

};

int main() {

//A Link List has been created with four nodes

List list;

//Adding Nodes into the list

list.add\_node(1);

list.add\_node(2);

list.add\_node(3);

list.add\_node(4);

list.add\_node(5);

list.displayList();

}

**Output:**

A screenshot of a computer

Description automatically generated with medium confidence

2. Make a function that can insert another node at 3rd location.

**Code:**

#include<iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

};

class List {

private:

Node\* head = NULL;

public:

void add\_node(int data\_value) {

Node\* temp = new Node();

temp->data = data\_value;

temp->next = head;

head = temp;

}

void displayList() {

Node\* temp = new Node();

temp = head;

while (temp != NULL) {

cout << temp->data << " ";

temp = temp->next;

}

}

int lengthList() {

int len=0;

Node\* temp = new Node();

temp = head;

while (temp != NULL) {

temp = temp->next;

len++;

}

return len;

}

void insertAt(int position, int value) {

Node\* temp = new Node();

int count = 1;

temp->data = value;

if (position == 1) {

temp->next = head;

head = temp;

}

if (position>1) {

Node\* insert = new Node();

insert->next = head;

while (count != position) {

insert = insert->next;

count++;

}

temp->next = insert->next;

insert->next = temp;

}

else {

cout << "Please Enter some valid number";

}

}

};

int main() {

//A Link List has been created with four nodes

List list;

//Adding Nodes into the list

list.add\_node(1);

list.add\_node(2);

list.add\_node(3);

list.add\_node(4);

list.add\_node(5);

list.displayList();

cout << endl;

list.insertAt(3, 10);

list.displayList();

}

**Output:**

**Graphical user interface, text

Description automatically generated**

3. Make a function that can display the lists made in 1 and 2.

**Code:**

void insertAt(int position, int value) {

Node\* temp = new Node();

int count = 1;

temp->data = value;

if (position == 1) {

temp->next = head;

head = temp;

}

if (position>1) {

Node\* insert = new Node();

insert->next = head;

while (count != position) {

insert = insert->next;

count++;

}

temp->next = insert->next;

insert->next = temp;

}

else {

cout << "Please Enter some valid number";

}

}

**Output:**

**Text

Description automatically generated**

4. Write a function that can delete node from the linked list selected by the user. Display it as well.

**Code:**

void deleteAt(int position){

Node\* temp = new Node();

Node\* prev = new Node();

temp = head;

int count = 1;

if (position == 1) {

temp = head;

head = head->next;

delete temp;

}

if (position >1) {

while (count != position) {

prev = temp;

temp = temp->next;

count++;

}

prev->next = temp->next;

delete temp;

}

else {

cout << "Please Enter some valid number";

}

}

5. Write a function that can count the number of nodes present in list.

int lengthList() {

int len=0;

Node\* temp = new Node();

temp = head;

while (temp != NULL) {

temp = temp->next;

len++;

}

return len;

}

**Output:**

Graphical user interface, text, application

Description automatically generated\

6. Create menu in main function to give call to all of the above functions depending upon user’s input.

#include<iostream>

using namespace std;

class Node {

public:

int data;

Node\* next;

};

class List {

private:

Node\* head = NULL;

public:

void add\_node(int data\_value) {

Node\* temp = new Node();

temp->data = data\_value;

temp->next = head;

head = temp;

}

void displayList() {

Node\* temp = new Node();

temp = head;

while (temp != NULL) {

cout << temp->data << " ";

temp = temp->next;

}

}

int lengthList() {

int len=0;

Node\* temp = new Node();

temp = head;

while (temp != NULL) {

temp = temp->next;

len++;

}

return len;

}

void insert\_at\_loc(int position, int value) {

Node\* temp = new Node();

int count = 1;

temp->data = value;

if (position > lengthList()) {

cout << endl;

cout << "The list doesn't have these numbers!!!" << endl;

}

else if (position == 1) {

temp->next = head;

head = temp;

}

else if (position > 1) {

Node\* insert = new Node();

insert->next = head;

while (count != position) {

insert = insert->next;

count++;

}

temp->next = insert->next;

insert->next = temp;

}

else {

cout << "Please Enter some valid number";

}

}

void del(int position){

Node\* temp = new Node();

Node\* prev = new Node();

temp = head;

int count = 1;

if (position > lengthList()) {

cout << endl;

cout << "The list doesn't have these elements" << endl;

}

else if (position == 1) {

temp = head;

head = head->next;

delete temp;

}

else if (position >1) {

while (count != position) {

prev = temp;

temp = temp->next;

count++;

}

prev->next = temp->next;

delete temp;

}

else {

cout << "Please Enter some valid number";

}

}

void insert\_at\_beginning(int new\_value) {

Node\* temp = new Node();

temp->data = new\_value;

temp->next = head;

head = temp;

}

};

int main() {

List list;

int choice=0;

int input,pos;

list.add\_node(6);

list.add\_node(2);

list.add\_node(3);

list.add\_node(9);

list.add\_node(4);

cout << "\*\*\*\*\*\*\*\*\*\* Functions Available \*\*\*\*\*\*\*\*\*\*" << endl;

cout << "1. Insert At the Beginning of List" << endl;

cout << "2. Insert At a specific Location in the list" << endl;

cout << "3. Delete any particular node in the list" << endl;

cout << "4. Display the list " << endl;

cout << "5. Number of nodes in the list" << endl;

cout << endl << endl;

cout << "Enter your choice number: " << endl;

cin >> choice;

switch (choice)

{

case 1:

cout << endl;

cout << "Enter the number you want to insert in the beginning of the list";

cout << endl;

cin >> input;

list.insert\_at\_beginning(input);

cout << endl;

cout << "The list after the insertion is : " << endl;

list.displayList();

break;

case 2:

cout << endl;

cout << "Enter the position at which you want to insert!";

cout << endl;

cin >> pos;

cout << endl;

cout << endl;

cout << "Enter the value you want to insert at position "<<pos<<" :"<<endl;

cin >> input;

list.insert\_at\_loc(pos, input);

cout << "The list after the insertion is : " << endl;

list.displayList();

break;

case 3:

cout << endl;

cout << "Enter the number you want to delete from the list" << endl;

cin >> input;

list.del(input);

cout << endl;

cout << "The list after the insertion is : " << endl;

list.displayList();

break;

case 4:

cout << endl;

cout << "The list is: " << endl;

list.displayList();

break;

case 5:

cout << endl;

cout << "The number of elements in the list are : "<<list.lengthList()<< endl;

break;

default:

cout << "Please Enter the number in between 1 and 5!!";

break;

}

}

**Text

Description automatically generated**