**Muhammad Abdullah**

**SE (3A) | 19F-0916**

DS Course Assign 2

management System

**QUESTION # 1**

**PROGRAM**

#include <iostream>

#include <string>

#include <fstream>

using namespace std; // BY M.ABDULLAH

struct Node

{

string first\_name;

string last\_name;

string rollno;

int id;

char blood\_group;

char gender;

int age;

string course1;

string course2;

Node \*NextNode, \*PreviousNode;

};

class Fun

{

public:

Fun()

{

Head = NULL;

}

Node \*Head;

void Add\_Node\_First(int val);

void Add\_Node\_Last(int val);

void Add\_Node\_Random(int val);

void Delete\_Node\_First();

void Delete\_Node\_Last();

void Delete\_Node\_Random();

void Search\_Node\_Data(int val);

void Update\_Node\_Data(int val);

void Count\_List();

void Display(int);

void Write\_in\_File(int val);

};

void Fun::Add\_Node\_First(int val)

{

Node \*temp = NULL, \*current = Head;

if (Head == NULL)

{

temp = new Node;

if (val == 1)

{

cout << " Enter ID of Teacher : ";

cin >> temp->id;

cout << " Enter First Name of Teacher : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Teacher: ";

cin >> temp->last\_name;

cout << " Enter Age of Teacher: ";

cin >> temp->age;

cout << " Enter Blood Group of Teacher: ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Teacher : ";

cin >> temp->gender;

cout << " Enter 1st Course of the Teacher : ";

cin >> temp->course1;

cout << " Enter 2nd Course of the Teacher : ";

cin >> temp->course2;

}

else if (val == 2)

{

cout << endl;

cout << " Enter ID of Student : ";

cin >> temp->id;

cout << " Enter First Name of Student : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Student : ";

cin >> temp->last\_name;

cout << " Enter Roll No. of the Student : ";

cin >> temp->rollno;

cout << " Enter Age of Student : ";

cin >> temp->age;

cout << " Enter Blood Group of Student : ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Student : ";

cin >> temp->gender;

}

else

cout << endl << "Invalid Entry !!" << endl;

temp->NextNode = NULL;

temp->PreviousNode = NULL;

Head = temp;

}

else

{

temp = new Node;

if (val == 1)

{

cout << " Enter ID of Teacher : ";

cin >> temp->id;

cout << " Enter First Name of Teacher : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Teacher: ";

cin >> temp->last\_name;

cout << " Enter Age of Teacher: ";

cin >> temp->age;

cout << " Enter Blood Group of Teacher: ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Teacher : ";

cin >> temp->gender;

cout << " Enter 1st Course of the Teacher : ";

cin >> temp->course1;

cout << " Enter 2nd Course of the Teacher : ";

cin >> temp->course2;

}

else if (val == 2)

{

cout << endl;

cout << " Enter ID of Student : ";

cin >> temp->id;

cout << " Enter First Name of Student : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Student : ";

cin >> temp->last\_name;

cout << " Enter Roll No. of the Student : ";

cin >> temp->rollno;

cout << " Enter Age of Student : ";

cin >> temp->age;

cout << " Enter Blood Group of Student : ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Student : ";

cin >> temp->gender;

}

else

cout << endl << "Invalid Entry !!" << endl;

temp->NextNode = current;

current->PreviousNode = temp;

temp->PreviousNode = NULL;

Head = temp;

}

}

void Fun::Add\_Node\_Last(int val)

{

Node \*temp = NULL, \*current = Head;

if (Head != NULL)

{

while (current->NextNode != NULL)

{

current = current->NextNode;

}

temp = new Node;

if (val == 1)

{

cout << " Enter ID of Teacher : ";

cin >> temp->id;

cout << " Enter First Name of Teacher : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Teacher: ";

cin >> temp->last\_name;

cout << " Enter Age of Teacher: ";

cin >> temp->age;

cout << " Enter Blood Group of Teacher: ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Teacher : ";

cin >> temp->gender;

cout << " Enter 1st Course of the Teacher : ";

cin >> temp->course1;

cout << " Enter 2nd Course of the Teacher : ";

cin >> temp->course2;

}

else if (val == 2)

{

cout << endl;

cout << " Enter ID of Student : ";

cin >> temp->id;

cout << " Enter First Name of Student : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Student : ";

cin >> temp->last\_name;

cout << " Enter Roll No. of the Student : ";

cin >> temp->rollno;

cout << " Enter Age of Student : ";

cin >> temp->age;

cout << " Enter Blood Group of Student : ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Student : ";

cin >> temp->gender;

}

else

cout << endl << "Invalid Entry !!" << endl;

current->NextNode = temp;

temp->PreviousNode = current;

temp->NextNode = NULL;

}

else

cout << endl << "!!! Create Link List First !!!" << endl;

}

void Fun::Add\_Node\_Random(int val)

{

Node \*temp = NULL, \*current = Head;

int opt = 0, counter = 0;

if (Head == NULL)

{

cout << endl << "Create List First !!" << endl;

}

else

{

Node \*temp1 = Head;

while (temp1 != NULL)

{

counter++;

temp1 = temp1->NextNode;

}

if (opt == 1)

{

if (Head == NULL)

{

temp = new Node;

if (val == 1)

{

cout << " Enter ID of Teacher : ";

cin >> temp->id;

cout << " Enter First Name of Teacher : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Teacher: ";

cin >> temp->last\_name;

cout << " Enter Age of Teacher: ";

cin >> temp->age;

cout << " Enter Blood Group of Teacher: ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Teacher : ";

cin >> temp->gender;

cout << " Enter 1st Course of the Teacher : ";

cin >> temp->course1;

cout << " Enter 2nd Course of the Teacher : ";

cin >> temp->course2;

}

else if (val == 2)

{

cout << endl;

cout << " Enter ID of Student : ";

cin >> temp->id;

cout << " Enter First Name of Student : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Student : ";

cin >> temp->last\_name;

cout << " Enter Roll No. of the Student : ";

cin >> temp->rollno;

cout << " Enter Age of Student : ";

cin >> temp->age;

cout << " Enter Blood Group of Student : ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Student : ";

cin >> temp->gender;

}

else

cout << endl << "Invalid Entry !!" << endl;

temp->NextNode = NULL;

temp->PreviousNode = Head;

Head = temp;

}

else

{

temp = new Node;

if (val == 1)

{

cout << " Enter ID of Teacher : ";

cin >> temp->id;

cout << " Enter First Name of Teacher : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Teacher: ";

cin >> temp->last\_name;

cout << " Enter Age of Teacher: ";

cin >> temp->age;

cout << " Enter Blood Group of Teacher: ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Teacher : ";

cin >> temp->gender;

cout << " Enter 1st Course of the Teacher : ";

cin >> temp->course1;

cout << " Enter 2nd Course of the Teacher : ";

cin >> temp->course2;

}

else if (val == 2)

{

cout << endl;

cout << " Enter ID of Student : ";

cin >> temp->id;

cout << " Enter First Name of Student : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Student : ";

cin >> temp->last\_name;

cout << " Enter Roll No. of the Student : ";

cin >> temp->rollno;

cout << " Enter Age of Student : ";

cin >> temp->age;

cout << " Enter Blood Group of Student : ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Student : ";

cin >> temp->gender;

}

else

cout << endl << "Invalid Entry !!" << endl;

current = Head;

current->PreviousNode = temp;

temp->NextNode = current;

temp->PreviousNode = Head;

Head = temp;

}

}

else if (opt == counter)

{

while (current->NextNode != NULL)

{

current = current->NextNode;

}

temp = new Node;

if (val == 1)

{

cout << " Enter ID of Teacher : ";

cin >> temp->id;

cout << " Enter First Name of Teacher : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Teacher: ";

cin >> temp->last\_name;

cout << " Enter Age of Teacher: ";

cin >> temp->age;

cout << " Enter Blood Group of Teacher: ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Teacher : ";

cin >> temp->gender;

cout << " Enter 1st Course of the Teacher : ";

cin >> temp->course1;

cout << " Enter 2nd Course of the Teacher : ";

cin >> temp->course2;

}

else if (val == 2)

{

cout << endl;

cout << " Enter ID of Student : ";

cin >> temp->id;

cout << " Enter First Name of Student : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Student : ";

cin >> temp->last\_name;

cout << " Enter Roll No. of the Student : ";

cin >> temp->rollno;

cout << " Enter Age of Student : ";

cin >> temp->age;

cout << " Enter Blood Group of Student : ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Student : ";

cin >> temp->gender;

}

else

cout << endl << "Invalid Entry !!" << endl;

current->NextNode = temp;

temp->PreviousNode = current;

temp->NextNode = NULL;

}

else if (opt > 1 && opt < counter)

{

current = Head;

for (int i = 2; i < counter - 1; i++)

{

if (i == opt)

{

temp = new Node;

if (val == 1)

{

cout << " Enter ID of Teacher : ";

cin >> temp->id;

cout << " Enter First Name of Teacher : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Teacher: ";

cin >> temp->last\_name;

cout << " Enter Age of Teacher: ";

cin >> temp->age;

cout << " Enter Blood Group of Teacher: ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Teacher : ";

cin >> temp->gender;

cout << " Enter 1st Course of the Teacher : ";

cin >> temp->course1;

cout << " Enter 2nd Course of the Teacher : ";

cin >> temp->course2;

}

else if (val == 2)

{

cout << endl;

cout << " Enter ID of Student : ";

cin >> temp->id;

cout << " Enter First Name of Student : ";

cin >> temp->first\_name;

cout << " Enter Last Name of Student : ";

cin >> temp->last\_name;

cout << " Enter Roll No. of the Student : ";

cin >> temp->rollno;

cout << " Enter Age of Student : ";

cin >> temp->age;

cout << " Enter Blood Group of Student : ";

cin >> temp->blood\_group;

cout << " Enter Gender of the Student : ";

cin >> temp->gender;

}

else

cout << endl << "Invalid Entry !!" << endl;

temp->NextNode = current->NextNode;

temp->PreviousNode = current;

current->NextNode->PreviousNode = temp;

current->NextNode = temp;

break;

}

current = current->NextNode;

}

}

}

}

void Fun::Delete\_Node\_First()

{

Node \*temp = NULL;

if (Head != NULL)

{

temp = new Node;

temp = Head;

Head = Head->NextNode;

Head->NextNode->PreviousNode = Head;

free(temp);

cout << endl << "First Node has been deleted !" << endl;

}

else

cout << "Link List is Empty ! " << endl;

}

void Fun::Delete\_Node\_Last()

{

Node \*temp = NULL, \*current = Head;

if (Head != NULL)

{

while (current->NextNode->NextNode != NULL)

{

current = current->NextNode;

}

temp = new Node;

temp = current->NextNode->NextNode;

current->NextNode = NULL;

free(temp);

cout << endl << "Last Node has been deleted !" << endl;

}

else

cout << "Link List is Empty ! " << endl;

}

void Fun::Delete\_Node\_Random()

{

Node \*temp = NULL, \*current = Head;

if (Head == NULL)

{

cout << endl << "Link List is Empty !" << endl;

}

else

{

int opt = 0, counter = 0;

cout << "Enter the Position of Node to Delete it : ";

cin >> opt;

while (current != NULL)

{

counter++;

current = current->NextNode;

}

if (opt == 1)

{

temp = new Node;

temp = Head;

Head = Head->NextNode;

Head->NextNode->PreviousNode = Head;

free(temp);

cout << endl << opt << " Node has been deleted !" << endl;

}

else if (opt == counter)

{

current = Head;

while (current->NextNode->NextNode != NULL)

{

current = current->NextNode;

}

temp = new Node;

temp = current->NextNode->NextNode;

current->NextNode = NULL;

free(temp);

cout << endl << opt << " Node has been deleted !" << endl;

}

else if (opt > 1 && opt < counter)

{

current = Head;

for (int i = 2; i < counter - 1; i++)

{

if (i == opt)

{

temp = new Node;

temp = current->NextNode;

current->NextNode = temp->NextNode;

temp->NextNode->PreviousNode = current;

free(temp);

cout << endl << opt << " Node has been deleted !" << endl;

break;

}

current = current->NextNode;

}

}

}

}

void Fun::Search\_Node\_Data(int val)

{

if (Head == NULL)

{

cout << endl << "Link List is Empty !" << endl;

}

else

{

Node \*current = Head;

int counter = 1;

while (current != NULL)

{

if (current->id == val)

{

cout << endl << "Value found at " << counter << " Node" << endl;

}

current = current->NextNode;

counter++;

}

}

}

void Fun::Update\_Node\_Data(int val)

{

if (Head == NULL)

{

cout << endl << "Link List is Empty !" << endl;

}

else

{

Node \*current = Head;

int counter = 1, id = 0;

if (val == 1)

{

cout << endl << "Enter Teacher ID to update it : ";

cin >> id;

while (current != NULL)

{

if (current->id == id)

{

cout << endl << "Enter the New Values to Assign to this Node !!" << endl;

cout << " Enter ID of Teacher : ";

cin >> current->id;

cout << " Enter First Name of Teacher : ";

cin >> current->first\_name;

cout << " Enter Last Name of Teacher: ";

cin >> current->last\_name;

cout << " Enter Age of Teacher: ";

cin >> current->age;

cout << " Enter Blood Group of Teacher: ";

cin >> current->blood\_group;

cout << " Enter Gender of the Teacher : ";

cin >> current->gender;

cout << " Enter 1st Course of the Teacher : ";

cin >> current->course1;

cout << " Enter 2nd Course of the Teacher : ";

cin >> current->course2;

}

current = current->NextNode;

counter++;

}

}

else if (val == 2)

{

cout << endl << "Enter Student ID to update it : ";

cin >> id;

while (current != NULL)

{

if (current->id == val)

{

cout << endl << "Enter the New Values to Assign to this Node !!" << endl;

cout << endl;

cout << " Enter ID of Student : ";

cin >> current->id;

cout << " Enter First Name of Student : ";

cin >> current->first\_name;

cout << " Enter Last Name of Student : ";

cin >> current->last\_name;

cout << " Enter Roll No. of the Student : ";

cin >> current->rollno;

cout << " Enter Age of Student : ";

cin >> current->age;

cout << " Enter Blood Group of Student : ";

cin >> current->blood\_group;

cout << " Enter Gender of the Student : ";

cin >> current->gender;

}

current = current->NextNode;

counter++;

}

}

else

cout << endl << "Invalid Entry !!" << endl;

}

}

void Fun::Count\_List()

{

Node\*temp = Head;

int counter = 0;

if (Head == NULL)

{

cout << endl << "Empty Link List !!" << endl;

}

else

{

while (temp != NULL)

{

counter++;

temp = temp->NextNode;

}

cout << endl << "Total Number of List is/are = " << counter << endl;

}

}

void Fun::Display(int val)

{

Node \*temp = Head;

cout << endl;

if (val == 1)

{

while (temp != NULL)

{

cout << " ID of Teacher is : " << temp->id << endl;

cout << " First Name of Teacher is : " << temp->first\_name << endl;

cout << " Last Name of Teacher is : " << temp->last\_name << endl;

cout << " Age of Teacher is : " << temp->age << endl;

cout << " Blood Group Name of Teacher is : " << temp->blood\_group << endl;

cout << " Gender of Teacher is : " << temp->gender << endl;

cout << " Course 1st of Teacher is : " << temp->course1 << endl;

cout << " Course 2nd of Teacher is : " << temp->course2 << endl;

temp = temp->NextNode;

}

}

else if (val == 2)

{

while (temp != NULL)

{

cout << " ID of Student is : " << temp->id << endl;

cout << " First Name of Student is : " << temp->first\_name << endl;

cout << " Last Name of Student is : " << temp->last\_name << endl;

cout << " Roll NO. of Teacher is : " << temp->rollno << endl;

cout << " Age of Student is : " << temp->age << endl;

cout << " Blood Group Name of Student is : " << temp->blood\_group << endl;

cout << " Gender of Student is : " << temp->gender << endl;

temp = temp->NextNode;

}

}

else

cout << endl << "Invalid Entry !!" << endl;

cout << endl;

}

void Fun::Write\_in\_File(int val)

{

Node \*temp = Head;

cout << endl;

ofstream file;

if (val == 1)

{

while (temp != NULL)

{

file.open("Teachers.txt");

if (file.is\_open())

{

file << temp->id << " " << temp->first\_name << " " << temp->last\_name << " "<< temp->age

<<" " << temp->blood\_group << " " << temp->gender << " " << temp->course1 << " " << temp->course2 << endl;

}

else

cout << endl << "Error Opening File !!" << endl;

temp = temp->NextNode;

}

}

else if (val == 2)

{

file.open("Students.txt");

if (file.is\_open())

{

file << temp->id << " " << temp->first\_name << " " << temp->last\_name << " " << temp->rollno << " " << temp->age

<< " " << temp->blood\_group << " " << temp->gender << endl;

}

else

cout << endl << "Error Opening File !!" << endl;

temp = temp->NextNode;

}

else

cout << endl << "Invalid Entry !!" << endl;

cout << endl;

}

int main()

{

Fun Student, Teacher;

int opt = 1, val = 0, choice = 0;

while (opt != 0)

{

system("cls");

cout << " -----------------------------------" << endl;

cout << " | Press 1 to Add Node on Start |" << endl;

cout << " | Press 2 to Add Node on Last |" << endl;

cout << " | Press 3 to Add Node Randomly |" << endl << " |\t\t\t\t |" << endl;

cout << " | Press 4 to Delete First Node |" << endl;

cout << " | Press 5 to Delete Last Node |" << "\t\t\tManagment System" << endl;

cout << " | Press 6 to Delete Random Node |" << endl << " |\t\t\t\t |" << endl;

cout << " | Press 7 to Search Data in Node |" << endl;

cout << " | Press 8 to Update Data in Node |" << endl;

cout << " | Press 9 to Count the Total Link |" << endl << " |\t\t\t\t |" << endl;

cout << " | Press 10 to Display Link List |" << endl;

cout << " | Press 11 to Write in File |" << endl << " |\t\t\t\t |" << endl;

cout << " | Press 0 to Exit from the system |" << endl;

cout << " -----------------------------------" << endl;

cout << endl << " Option Choosen : ";

cin >> opt;

cout << endl;

switch (opt)

{

case 1:

{

cout << endl << "Press 1 to Add Teacher and 2 to Add Student : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Add\_Node\_First(choice);

}

else if (choice == 2)

{

Student.Add\_Node\_First(choice);

}

else

cout << endl << "Invalid Entry !!" << endl;

system("pause");

break;

}

case 2:

{

cout << endl << "Press 1 to Add Teacher and 2 to Add Student : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Add\_Node\_Last(choice);

}

else if (choice == 2)

{

Student.Add\_Node\_Last(choice);

}

else

cout << endl << "Invalid Entry !!" << endl;

system("pause");

break;

}

case 3:

{

cout << endl << "Press 1 to Add Teacher and 2 to Add Student : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Add\_Node\_Random(choice);

}

else if (choice == 2)

{

Student.Add\_Node\_Random(choice);

}

else

cout << endl << "Invalid Entry !!" << endl;

system("pause");

break;

}

case 4:

{

cout << endl << "Press 1 to Delete Teacher and 2 to Delete Student : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Delete\_Node\_First();

}

else if (choice == 2)

{

Student.Delete\_Node\_First();

}

else

cout << endl << "Invalid Entry !!" << endl;

system("pause");

break;

}

case 5:

{

cout << endl << "Press 1 to Delete Teacher and 2 to Delete Student : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Delete\_Node\_Last();

}

else if (choice == 2)

{

Student.Delete\_Node\_Last();

}

else

cout << endl << "Invalid Entry !!" << endl;

system("pause");

break;

}

case 6:

{

cout << endl << "Press 1 to Delete Teacher and 2 to Delete Student : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Delete\_Node\_Random();

}

else if (choice == 2)

{

Student.Delete\_Node\_Random();

}

else

cout << endl << "Invalid Entry !!" << endl;

system("pause");

break;

}

case 7:

{

cout << endl << "Press 1 to Search Teacher and 2 to Search Student : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Search\_Node\_Data(choice);

}

else if (choice == 2)

{

Student.Search\_Node\_Data(choice);

}

else

cout << endl << "Invalid Entry !!" << endl;

cout << endl;

system("pause");

break;

}

case 8:

{

cout << endl << "Press 1 to Update Teacher and 2 to Update Student : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Update\_Node\_Data(choice);

}

else if (choice == 2)

{

Student.Update\_Node\_Data(choice);

}

else

cout << endl << "Invalid Entry !!" << endl;

cout << endl;

system("pause");

break;

}

case 9:

{

cout << endl << "Press 1 to Count Teachers and 2 to Count Students : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Count\_List();

}

else if (choice == 2)

{

Student.Count\_List();

}

else

cout << endl << "Invalid Entry !!" << endl;

system("pause");

break;

}

case 10:

{

cout << endl << "Press 1 to Display Teachers and 2 to Display Students : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Display(choice);

}

else if (choice == 2)

{

Student.Display(choice);

}

else

cout << endl << "Invalid Entry !!" << endl;

system("pause");

break;

}

case 11:

{

cout << endl << "Press 1 to Write Teacher Data and 2 to Write Student Data : ";

cin >> choice;

cout << endl;

if (choice == 1)

{

Teacher.Write\_in\_File(choice);

}

else if (choice == 2)

{

Student.Write\_in\_File(choice);

}

else

cout << endl << "Invalid Entry !!" << endl;

system("pause");

break;

}

case 0:

{

opt = 0;

cout << endl << "You have exited from Link List !" << endl;

break;

}

default:

cout << "Invalid Entry, Press any key to try again !!" << endl;

system("pause");

break;

}

}

cout << endl << endl;

system("pause");

}

A screenshot of a computer screen

Description automatically generated