

Answer on Question #38377, Programming, C++

```
#include <cstdlib>
#include <iostream>
#include <string>
#include <iomanip>

using namespace std;
class Person{
public:
    string name;
    string contactnumber;
    Person* next;
};

Person* Person_List=NULL;//create list of Persons using linked list
void AddPerson(string name,string contactnumber);
void PrintPersons();
void Delete(string contactnumber);
void updateTelephoneNumber(string contactnumber);

int main(int argc, char *argv[])
{
    string name;//variable for name
    string contactnumber;//variable for
    int ch;//user choice
    do{
        ch=-1;
        while(ch<1 || ch>4){
            //display menu
            cout<<"1 - Add new Person to list\n";
            cout<<"2 - Print all persons\n";
            cout<<"3 - Update the telephone number\n";
            cout<<"4 - Exit\n";
            cout<<"select one menu item: ";
            cin>>ch;

            if(ch<0){
                cout<<"Enter correct choice!!!";
                exit(0);
            }
            //system("PAUSE");//delay
        }

        switch(ch){
            case 1:
                //add new person
                cout<<"Enter name: ";
                cin>>name;
                cout<<"Enter contact number: ";
                cin>>contactnumber;
                AddPerson(name,contactnumber);
                break;
            case 2:
                //Print all persons
                PrintPersons();
                break;
            case 3:
                //update Telephone Number
                cout<<"Enter contact number: ";
                cin>>contactnumber;
```

```

        updateTelephoneNumber(contactnumber);
        break;
    }
    }while(ch!=4);//exit loop
    system("PAUSE");//delay
    return EXIT_SUCCESS;
}
//Add new Person to list
void AddPerson(string name,string contactnumber){
    Person* new_Node;
    Person* previous_Node;
    Person* current_Node;
    new_Node = new Person();

    if (new_Node != NULL ) {

        new_Node->name=name;//set name to list
        new_Node->contactnumber=contactnumber;//set contact number to list

        new_Node->next = NULL; //set next to NULL
        previous_Node = NULL; //set previous_Node to NULL
        current_Node = Person_List;//set current_Node to Person_List
        //sort list by name
        while (current_Node != NULL && strcmp (name.c_str(),current_Node-
>name.c_str())>0) {
            //swap
            previous_Node = current_Node;
            current_Node = current_Node->next;
        }
        if (previous_Node == NULL ) {
            new_Node->next = Person_List;
            Person_List = new_Node;
        }
        else {
            previous_Node->next = new_Node;
            new_Node->next = current_Node;
        }
    }
}

//Print all Person
void PrintPersons(){
    Person* show_Node;
    if (Person_List == NULL ) {
        cout<<"Link list is empty.\n\n";
    }else {
        cout<<"\n\nName\t\tContact number\n";
        show_Node = Person_List;
        while(show_Node != NULL ) {
            cout<<left<<setw(15)<<show_Node->name<<"\t\t"<<show_Node-
>contactnumber<<"\n";
            show_Node = show_Node->next;
        }
        cout<<"\n\n";
    } // end else
}

//update the telephone number
void updateTelephoneNumber(string contactnumber){
    Person* display_Node;
    string tempname;
    bool flag=false;
    if (Person_List == NULL ) {

```

```

        printf( "List is empty.\n\n" );
    }else {
        display_Node=Person_List;
        //found if such contac number is in list
        while(display_Node != NULL ) {
            if(contactnumber==display_Node->contactnumber){
                cout<<left<<setw(15)<<display_Node->name<<"\t\t"<<display_Node-
>contactnumber<<"\n";
                flag=true;
                tempname=display_Node->name;
            }
            display_Node = display_Node->next;
        }
        if(flag==false){
            cout<<"No such Person!!!\n\n";
        }else{
            //display_Node=Person_List;
            string newcontactnumber;
            //input new contact name
            cout<<"Enter new contact number: ";
            cin>>newcontactnumber;
            Delete(contactnumber);
            AddPerson(tempname,newcontactnumber);
        }
    } // end while
} // end else
}

void Delete(string contactnumber){
    Person* current_Node;
    Person* previous_Node;
    Person* tempNode;

    if (contactnumber == Person_List->contactnumber) {
        tempNode = Person_List;
        Person_List = Person_List->next;
        free( tempNode ); // free node
    }else {
        previous_Node = Person_List;
        current_Node = Person_List->next;
        while ( current_Node != NULL && current_Node->contactnumber !=
contactnumber ) {
            previous_Node = current_Node;
            current_Node = current_Node->next;
        }
        if ( current_Node != NULL ) {
            tempNode = current_Node;
            previous_Node->next = current_Node->next;
            free( tempNode );
        }
    }
}
}
}

```