Comsats University Islamabad, Vehari Campus

Assignment NO:02

Submitted By:

Muhammad Usman Azhar

Submitted To:

Mam Yasmeen Jana

Subject:

Data Structure and Algorithm

Registration No:

SP22-BCS-092(B)

Date of submission: **09/10/2023**

Activity 1:

Code

```
#include<iostream>
using namespace std;
class Node{
        private:
                int data;
                Node *next;
        public:
                Node *head;

                Node(){
                        head==NULL;
                }

                void insert_beg(int n){
                        if(head==NULL){
                                head=new Node();
                                head->data=n;
                                head->next=NULL;
                        }
                        else{
                                Node *ptr;
                                ptr=new Node();
                                ptr->next=head;
                                ptr->data=n;
                                head=ptr;
                        }
```

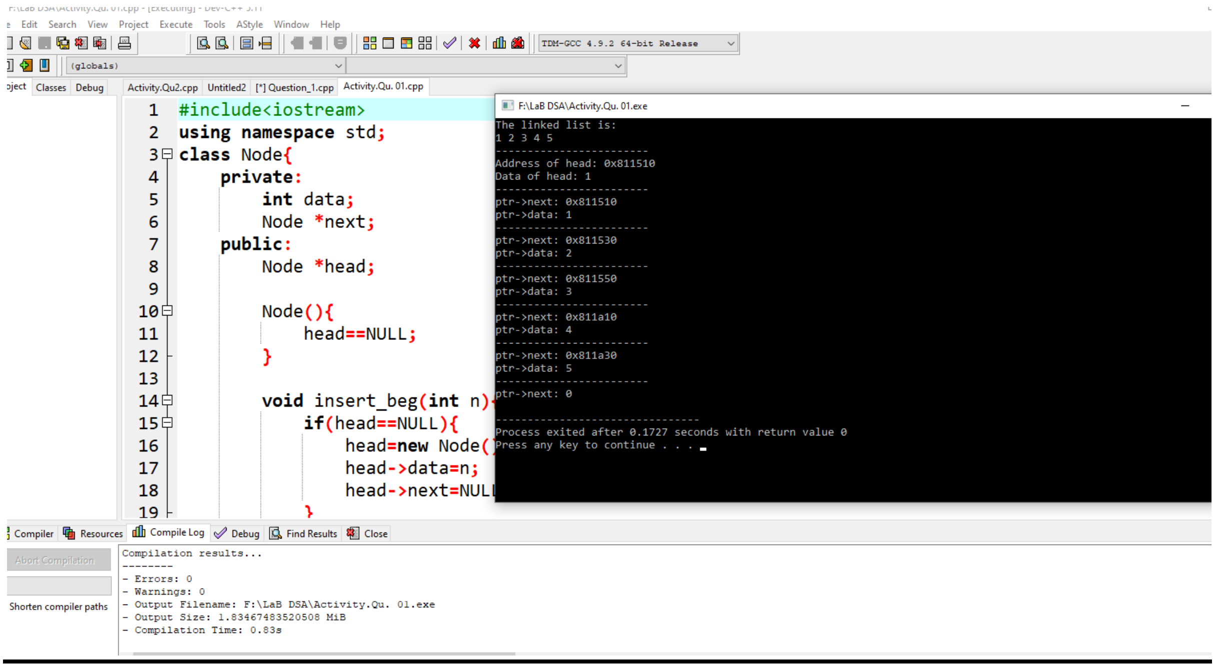
```
}

void insert_specific(int pos, int n){
    if(head==NULL){
        head = new Node();
        head->data=n;
        head->next=NULL;
    }
    else{
        Node *ptr;
        ptr=head;
        while(ptr->data!=pos){
            ptr=ptr->next;
        }
        Node *p;
        p=new Node();
        p->data=n;
        p->next=ptr->next;
        ptr->next=p;
    }
}

void insert_end(int n){
    if(head==NULL){
        head=new Node;
        head->data=n;
        head->next=NULL;
    }
    else{
```

```
                Node *ptr, *p;
                ptr=head;
                while(ptr->next!=NULL){
                        ptr=ptr->next;
                }
                p= new Node();
                p->data=n;
                p->next=NULL;
                ptr->next=p;
        }
}

void display(){
        if(head==NULL){
                cout<<"There is no list "<<endl;
        }
        else{
                Node *ptr;
                ptr=head;
                cout<<"The linked list is: "<<endl;
                while(ptr!=NULL){
                        cout<<ptr->data<<" ";
                        ptr=ptr->next;
                }
                cout<<endl;
        }
        cout<<"------------------------"<<endl;
}
```

```
        void display1(){
            Node *ptr;
            ptr=head;
            cout << "Address of head: " << head <<endl;
            cout<< "Data of head: " << head->data << endl;
            cout<<"------------------------"<<endl;
            if(head==NULL){
                cout<<"There is no list"<<endl;
                cout << "Address of head: " << head << ", Data of
head: " << head->data << endl;
            }

            else{
     while (ptr != NULL) {


      cout << "ptr->next: " << ptr <<endl;
      cout<<"ptr->data: "<<ptr->data<<endl;
      cout<<"------------------------"<<endl;
      ptr = ptr->next;
     }
     cout << "ptr->next: " << ptr <<endl;
   }


            }

};
int main(){
    Node n;
    n.insert_beg(1);
```

```
        n.insert_end(2);
        n.insert_end(3);
        n.insert_end(4);
        n.insert_end(5);
        n.display();
        n.display1();
        return 0;
}
```

OutPut


F:\LaB DSA\Activity.Qu. 01.cpp - [Executing] - Dev-C++ 5.11
e Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
oject Classes Debug
Activity.Qu2.cpp Untitled2 [*] Question_1.cpp Activity.Qu. 01.cpp
F:\LaB DSA\Activity.Qu. 01.exe
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths

```
#include<iostream>
using namespace std;
class Node{
    private:
        int data;
        Node *next;
    public:
        Node *head;

        Node(){
            head==NULL;
        }

        void insert_beg(int n)
            if(head==NULL){
                head=new Node(
                head->data=n;
                head->next=NUL
            }
```

```
The linked list is:
1 2 3 4 5
------------------------
Address of head: 0x811510
Data of head: 1
------------------------
ptr->next: 0x811510
ptr->data: 1
------------------------
ptr->next: 0x811530
ptr->data: 2
------------------------
ptr->next: 0x811550
ptr->data: 3
------------------------
ptr->next: 0x811a10
ptr->data: 4
------------------------
ptr->next: 0x811a30
ptr->data: 5
------------------------
ptr->next: 0

--------------------------------
Process exited after 0.1727 seconds with return value 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: F:\LaB DSA\Activity.Qu. 01.exe
- Output Size: 1.83467483520508 MiB
- Compilation Time: 0.83s
```

Activity 2:

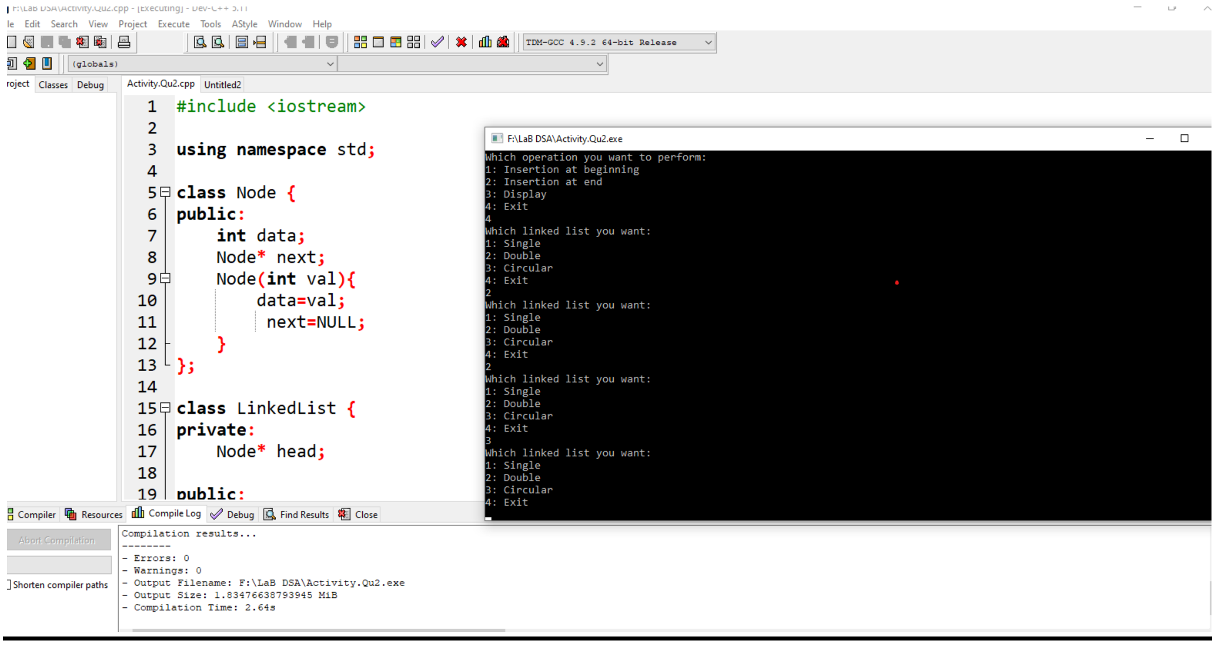
Code

```
#include <iostream>

using namespace std;

class Node {
public:
    int data;
    Node* next;
    Node(int val){
          data=val;
                    next=NULL;
          }
};

class LinkedList {
private:
    Node* head;

public:
    LinkedList() {
          head=NULL;
          }

    // Function to insert a node at the beginning
    void insertAtBeginning(int val) {
        Node* newNode = new Node(val);
        newNode->next = head;
        head = newNode;
    }

    // Function to insert a node at the end
    void insertAtEnd(int val) {
        Node* newNode = new Node(val);
        if (!head) {
            head = newNode;
        } else {
            Node* current = head;
            while (current->next) {
                current = current->next;
            }
            current->next = newNode;
        }
    }

    // Function to display the linked list
    void display() {
        Node* current = head;
        while (current) {
            cout << current->data << " -> ";
            current = current->next;
        }
        cout << "nullptr" << endl;
    }
};
```

```
int main() {
    LinkedList sll;
    int choice, data;

    do {
        cout << "Which linked list you want:" << endl;
        cout << "1: Single" << endl;
        cout << "2: Double" << endl;
        cout << "3: Circular" << endl;
        cout << "4: Exit" << endl;
        cin >> choice;

        switch (choice) {
            case 1:
                int subChoice;
                do {
                    cout << "Which operation you want to perform:" << endl;
                    cout << "1: Insertion at beginning" << endl;
                    cout << "2: Insertion at end" << endl;
                    cout << "3: Display" << endl;
                    cout << "4: Exit" << endl;
                    cin >> subChoice;

                    switch (subChoice) {
                        case 1:
                            cout << "Enter data to insert at the beginning: ";
                            cin >> data;
                            sll.insertAtBeginning(data);
                            break;
                case 2:
                    cout << "Enter data to insert at the end: ";
                    cin >> data;
                    sll.insertAtEnd(data);
                    break;
                case 3:
                    cout << "Linked List: ";
                    sll.display();
                    break;
                case 4:
                    break;
                default:
                    cout << "Invalid option" << endl;
            }
        } while (subChoice != 4);
        break;
    case 2:
        // Implement similar options for doubly linked list
        break;
    case 3:
        // Implement similar options for circular linked list
        break;
    case 4:
        cout << "Exiting program" << endl;
        break;
    default:
        cout << "Invalid option" << endl;
  }
} while (choice != 4);
```

```
    return 0;
}
```

Output


F:\LaB DSA\Activity.Qu2.cpp - [Executing] - Dev-C++ 5.11
le Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
roject Classes Debug
Activity.Qu2.cpp Untitled2
F:\LaB DSA\Activity.Qu2.exe
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths

```
#include <iostream>

using namespace std;

class Node {
public:
    int data;
    Node* next;
    Node(int val){
        data=val;
         next=NULL;
    }
};

class LinkedList {
private:
    Node* head;

public:
```

```
Which operation you want to perform:
1: Insertion at beginning
2: Insertion at end
3: Display
4: Exit
4
Which linked list you want:
1: Single
2: Double
3: Circular
4: Exit
2
Which linked list you want:
1: Single
2: Double
3: Circular
4: Exit
2
Which linked list you want:
1: Single
2: Double
3: Circular
4: Exit
3
Which linked list you want:
1: Single
2: Double
3: Circular
4: Exit
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: F:\LaB DSA\Activity.Qu2.exe
- Output Size: 1.83476638793945 MiB
- Compilation Time: 2.64s
```