राष्ट्रीय प्रौद्योगिकी संस्थान दिल्ली
National Institute of Technology Delhi
(An autonomous Institute under the aegis of Ministry of Education, Govt. of India)

# *Object Oriented Programming*

## Assignment 5


*Name : Ganta Sowmya Kranthi*

*Roll no : 201210019*

*Year : 2nd year*

*Semester :4th Sem*

*Group : 1*

Question 1: Write a C++ program to create a linked list using structure and class and implement following methods into class.

1.Print (to print the elements of linked list)

2.Head(to print the head of linked list)

3.Tail(to print the tail of linked list).

4.Reverse the elements of linked list.

5.Sort the linked list in increasing order.

Code:

```
#include <iostream>

using namespace std;

struct node
{
    int data;
    struct node *link;
};
class linkedlist
{
public:
   node *head,*tail;

public:
    linkedlist()
    {
        head = NULL;
        tail = NULL;
    }

  void insertnode(int data);
  void printlist();
  void reverselist();
  void sorting();

};

void linkedlist::insertnode(int data)
```

```
{

    struct node *tmp=new node;
    tmp->data=data;
    tmp->link=NULL;
    if(head == NULL)
        {
            head = tmp;
            tail = tmp;
        }
        else
        {
            tail->link = tmp;
            tail = tail->link;
        }

}


void linkedlist::printlist()
{
    struct node* temp = head;
    if(head ==NULL)
    {
        cout<<"Linkedlist is empty"<<endl;
        return;
    }
 else{
     cout<<"elements are"<<endl;
    while(temp!=NULL)
    {
        cout<<temp->data<<"\t ";;
        temp = temp->link;
    }
  cout <<endl;
 }
}

void linkedlist::reverselist()
{
    struct node* temp = NULL;
    struct node* prev = NULL;

    struct node *current = head;
    while(current!=NULL)
    {
        temp = current->link;
        current->link = prev;
        prev = current;
```

```
        current = temp;

    }
    head = prev;
    cout<<"After Reversing :";

    while(head!= NULL)
    {
        cout<<head->data<<"\t";
        head = head->link;
    }
}


void linkedlist::sorting()
{
    struct node *current,*bcurrent;
    current = head;
    bcurrent = head->link;
    int count1 =0;

    for(int i = count1-1;i>=0;i--)
    {
        for(int j=0;j<count1-1;j++)
        {
           if( current->data > bcurrent->data)
            {
                swap(current->data, bcurrent->data);
            }
            current = bcurrent;
            bcurrent= bcurrent->link;
        }
    }



}



int main()
{


    int choice,data;

    linkedlist list;
    do{
```

```
    cout<<"1.Enter elements to linked list \n";
    cout<<"2. Reverse linked list \n";
    cout<<"3. Sort the linked list \n";
    cout<<"4. print the linked list \n";
    cout<<"5.exit \n";
    cin>>choice;


    switch(choice)
    {
        case 1:
        cout<<"Enter data :";
        cin>>data;
        list.insertnode(data);
        break;

        case 2:
        list.reverselist();
        break;

        case 3:
        list.sorting();
        break;

        case 4:
        list.printlist();
        break;

        case 5:
            exit(5);
             break;
        default:
        cout<<"Enter right choice ";
           break;
    }
}while(choice!=5);

    return 0;


}
```

## Output:

```
1
Enter data :6
1.Enter elements to linked list
2. Reverse linked list
3. Sort the linked list
4. print the linked list
5.exit
1
Enter data :7
1.Enter elements to linked list
2. Reverse linked list
3. Sort the linked list
4. print the linked list
5.exit
4
elements are
1       2       3       4       5       6       7
1.Enter elements to linked list
2. Reverse linked list
3. Sort the linked list
4. print the linked list
5.exit
```

```
4. print the linked list
5.exit
2
After Reversing :7      6       5       4       3       2       1
2. Reverse linked list
3. Sort the linked list
4. print the linked list
5.exit
```