**Comsats University Islamabad, Attock Campus**

**Data Structures**

**Assignment 1**

**Submitted to**

**Sir Kamran**

**By**

|  |  |
| --- | --- |
| **Haroon Ijaz** | **FA23-BSE-002** |

**Assignment Report:**

**Structure of Node:**

**Attributes:** id, description, priority, next\_ptr: point towards next Node.

**A black and white rectangular object with text

AI-generated content may be incorrect.constructor**: Node (id, description, priority).

In constructor all above attributes are

initialized.

**Creating a Link List:**

Note:1,2,3… highest priority

**Attributes:** head ptr.

**constructor:** List().

Initially, (when no node Exists) head =NLL

**Adding a new Task** (addNewTask (string description, int id, int priority))**:**

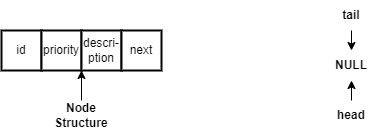
Parameters: description, id, priority.

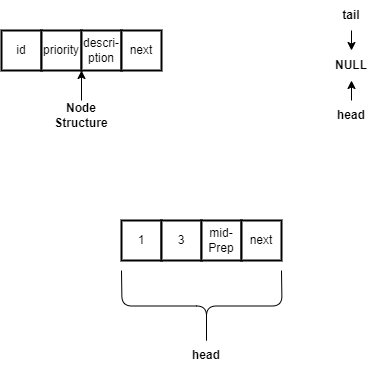
* Create newNode and pass values to the constructor of node Class

Node \*newNode = new Node(description, id, priority);

* If no Node exists:

Head :points towards the first node in linked list.

Head==NULL

When we create newNode:

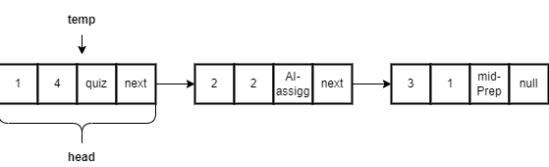
newNode->next=head;

head=newNode;

* If a node already exists, then we check priority. If new Node has highest priority then same above step will performed:

A diagram of a flowchart

AI-generated content may be incorrect.

* If newNode’s priority is less than or equal to already existing Node then:
* A temp node=head is set
* Using while we iterate over it to find our correct destination.

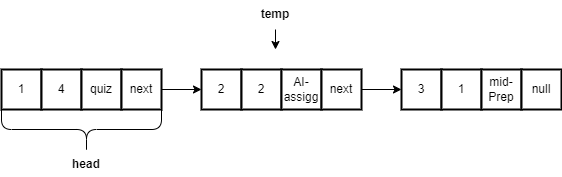
While(temp->next!=NULL && temp->next->priority<=priority{

temp=temp->next;}

newNode->next=temp->next;

temp->next=newNode;

Lets say we want to enter node with priority 1. In first iteration of loop, temp points towards firstNode(head).

Loop condition is true so it moves temp to the next node.

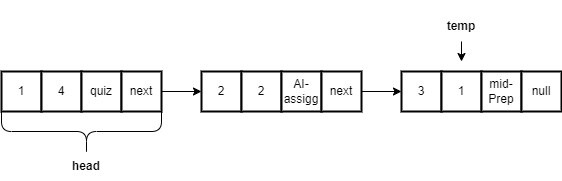
* Temp= temp->next
* Same for second iteration.
* in third iteration loop condtion is false:

Figure :after .first Iteration

so, newNode->next=

temp->next;

Figure : after second Iteration

A diagram of a block diagram

AI-generated content may be incorrect.temp->next=nextNode

Figure :linking newNode with temp

**Deleting task from list by ID:**

* If head==null or there are no nodes in list then:

Cout<<”Empty List”<<endl;

* Else:

We take temp node(to get the node we want to delete) and prv node(for track of previous node of temp).

Node\* temp=head;

Node\* prv=NULL;

A diagram of a flowchart

AI-generated content may be incorrect.

Lets say we want to delete node with id 3.

* + In first iteration, while condition checks: while(temp!=NULL) then:
  + If statement checks :if(temp->id==id)
  + **In our case temp->id!=id so, we move prv and temp to the next node.**

prv=temp;

temp=temp->

A diagram of a block diagram

AI-generated content may be incorrect.

* **In second case still temp->id!=id: so perform above step:**

A diagram of a block diagram

AI-generated content may be incorrect.**Now:**

* **In third iteration condition is if(temp->==id) is true: so**
  + Inside first if statement second if statement checks if (prv==NULL) shift head to the nextNode.

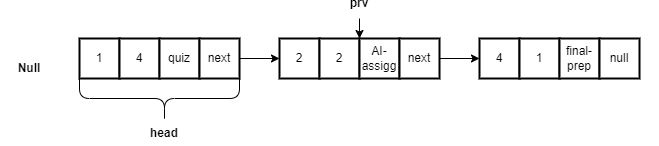
Head=temp->next;

**In this case: we skip second if statement because head prv is not NULL so, we go with the else case:**

prv->next=temp->next

* After else statement we delete temp and return from the function:

**delete temp;**

**return;**

**Deleting highest Priority Task:**

**->**check if head==NULL then cout<<”List is empty”;

->take temp=head

Shift head to next node:

A diagram of a diagram

AI-generated content may be incorrect. Head=temp->next

A diagram of a block diagram

AI-generated content may be incorrect.->delete temp;

**ViewTask():**

**->if head==NULL means empty list. Nothing to print.**

**->Else:take while loop with condition: while (temp!==null)**

**Cout data of each temp node:**

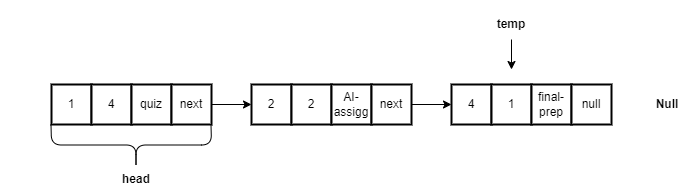
**cout<<temp->id<<temp->desp<<temp->priority<<endl;**

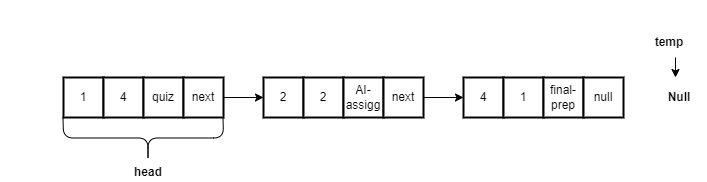
A diagram of a computer

AI-generated content may be incorrect.

A diagram of a flowchart

AI-generated content may be incorrect.





**myMenu() method:**

**->**refrenrence variable of List is created in this function**:List list;**

**->**switch case is used to get user choice from 1 to 5:

  cout << "1. Add description" << endl;

        cout << "2. Delete description" << endl;

        cout << "3. View description" << endl;

        cout << "4. Delete highest priority task" << endl;

        cout << "5. Exit" << endl;

**->each function of List is called (list.addNewTask) in switch cases according to user choice.**

**->It takes choice from user until he enters 5.**

**Main Method():**

**->myMenu() function called inside the main function**