DS Lab Assignment 03

Name : Pradyumna Gayake

Roll No.:160

PRN : 0120160498

```
#include<iostream>
#include<cstring>
using namespace std;
class node
{
public:
int id;
char name[20];
char branch[20];
float cgpa;
node *next;
};
class linkedlist
{
private:
node *head;
public:
linkedlist()
{
head=NULL;
}
void create_list();
//class variable
//head node declare
void display();
void insert_begining();
void insert_between();
void insert_last();
void insert_inbetween();
//Function Declaration
void delete_begining();
void delete_between();
void delete_last();
void delete_Byvalue();void reverse();
void update();
void search();
void sort();
void swap(node*p1,node*p2);
};
void linkedlist::create_list()
{
node *temp=NULL;
node *p=NULL;
int no,choice=1;
//create list function
while(choice!=0)
{
if(head==NULL)
```

```
 49  {
 50  head=new(node);
 51  cout<<"\nEnter the id no:";
 52  cin>>head->id;
 53  cout<<"Enter the Name:";
 54  cin>>head->name;
 55  cout<<"Enter the Branch:";
 56  cin>>head->branch;
 57  cout<<"Enter the cgpa:";
 58  cin>>head->cgpa;
 59  head->next=NULL;
 60  p=head;
 61  }
 62  else
 63  {
 64  temp=new(node);
 65  cout<<"\nEnter the id no:";
 66  cin>>temp->id;
 67  cout<<"Enter the Name:";
 68  cin>>temp->name;
 69  cout<<"Enter the Branch:";
 70  cin>>temp->branch;
 71  cout<<"Enter the cgpa:";
 72  cin>>temp->cgpa;
 73  temp->next=NULL;
 74  p->next=NULL;
 75  p->next=temp;
 76  //if first node
 77  //if not first nodep=temp;
 78  }
 79  cout<<"\nTo Continue press 1 else press 0 :";
 80  cin>>choice;
 81  }
 82  }
 83  void linkedlist::display()
 84  //Display Function
 85  {
 86  cout<<"\nDisplaying list :\n";
 87  node *temp1=NULL;
 88  temp1=head;
 89  while(temp1!=NULL)
 90  {
 91  cout<<"Student id:"<<temp1->id<<"\nStudent name:"<<temp1->name<<"\
nStudent
 92  branch:"<<temp1->branch<<"\nStudent Cgpa:"<<temp1->cgpa<<"\n\n";
 93  temp1=temp1->next;
 94  }
 95  }
 96  void linkedlist::insert_begining()
 97  {
 98  //Insert Begining
 99  node *temp=NULL;
100  temp=new(node);
101  cout<<"Enter id :";
102  cin>>temp->id;
103  cout<<"Enter Name :";
104  cin>>temp->name;
105  cout<<"Enter Branch :";
106  cin>>temp->branch;
107  cout<<"Enter Cgpa :";
```

```
cin>>temp->cgpa;
temp->next=NULL;
temp->next=head;
head=temp;
}
void linkedlist::insert_between()
//Insert between
{
int pos;
cout<<"Enter the Position You Want to Enter";
cin>>pos;
node*p=NULL;
p=head;for(int i=1;i<pos-1;i++)
{
p=p->next;
}
node*temp=NULL;
temp=new(node);
cout<<"Enter the id";
cin>>temp->id;
cout<<"Enter Name :";
cin>>temp->name;
cout<<"Enter Branch :";
cin>>temp->branch;
cout<<"Enter Cgpa :";
cin>>temp->cgpa;
temp->next=p->next;
p->next=temp;
}
void linkedlist::insert_last()
{
node*p=NULL;
p=head;
while(p->next!=NULL)
{
p=p->next;
}
//for loop to bring pointer to specific node
//insert at last
//while loop to bring pointer to specific node
node*temp=NULL;
temp=new(node);
cout<<"Enter the id:";
cin>>temp->id;
cout<<"Enter Name :";
cin>>temp->name;
cout<<"Enter Branch :";
cin>>temp->branch;
cout<<"Enter Cgpa :";
cin>>temp->cgpa;
temp->next=NULL;
p->next=temp;
}
void linkedlist::insert_inbetween()
{
node*temp1=NULL;
temp1=head;
//insert by valuecout<<head->id;
int flag=0;
int no;
```

```
168  cout<<"Enter the value after which you want to enter the no:";
169  cin>>no;
170  while(temp1->next!=NULL)
171  //while loop to bring pointer to specific node
172  {
173  if(temp1->id==no)
174  {
175  flag=1;
176  break;
177  }
178  temp1=temp1->next;
179  //flag=1 means value found
180  }
181  cout<<head->id;
182  if(flag==0)
183  {
184  cout<<"Value not Find";
185  }
186  //if value not found
187  else
188  {
189  node*p=NULL;
190  p=new(node);
191  cout<<"Enter the no";
192  cin>>p->id;
193  p->next=NULL;
194  if(temp1->next!=NULL)
195  {
196  p->next=temp1->next;
197  temp1->next=p;
198  }
199  else
200  {
201  temp1->next=p;
202  }
203  //if not last node
204  //if last node
205  }
206  }
207  void linkedlist::delete_begining()
208  //delete at begining{
209  node*temp=NULL;
210  temp=head;
211  head=temp->next;
212  delete(temp);
213  }
214  void linkedlist::delete_between()
215  {
216  int pos;
217  node*temp=NULL;
218  temp=head;
219  cout<<"Enter the Position you want to delete the value";
220  cin>>pos;
221  for(int i=1;i<pos-1;i++)
222  //for loop to bring pointer to specific node
223  {
224  temp=temp->next;
225  }
226  node*p=NULL;
227  p=temp->next;
```

```
228  temp->next=temp->next->next;
229  delete(p);
230  //deleted using function delete
231  cout<<"Record Deleted Successfully\n";
232  }
233  void linkedlist::delete_last()
234  {
235  node*temp=NULL;
236  node*p=NULL;
237  node*temp1=NULL;
238  temp=head;
239  //delete at last
240  while(temp->next!=NULL)
241  {
242  p=temp;
243  temp=temp->next;
244  }
245  temp1=p->next;
246  p->next=NULL;
247  delete(temp1);
248  cout<<"Record Deleted Successfully\n";
249  }
250  void linkedlist::delete_Byvalue()
251  //delete by value{
252  int no,flag=0;int c=0;
253  node*temp=NULL;
254  temp=head;
255  node*p=NULL;
256  p=head;
257  cout<<"Enter the value you want to delete";
258  cin>>no;
259  while(temp!=NULL)
260  {
261  if(temp->id==no)
262  {
263  flag=1;
264  //flag=1 means value found
265  break;
266  }
267  temp=temp->next;
268  c++;
269  }
270  cout<<c;
271  for(int i=1;i<c;i++)
272  {
273  p=p->next;
274  }
275  if(flag==0)
276  {
277  cout<<"not present";
278  //if value not found
279  }
280  else
281  {
282  if(temp==head)
283  {
284  head=head->next;
285  //for first node
286  delete(temp);
287  }
```

```
288  else if(temp->next==NULL)
289  {
290  p->next=NULL;
291  delete(temp);
292  //for last node
293  }
294  else
295  {
296  p->next=temp->next;
297  delete(temp);
298  //in-between node
299  }
300  }
301  cout<<"Record Deleted Successfully\n";}
302  void linkedlist::reverse()
303  {
304  //reverse the sequence
305  node*p=NULL;
306  node*q=NULL;
307  node*r=NULL;
308  q=head;
309  r=head->next;
310  while(q!=NULL)
311  {
312  q->next=p;
313  p=q;
314  q=r;
315  if(q!=NULL)
316  {
317  r=q->next;
318  }
319  }
320  head=p;
321  cout<<"Record Deleted Successfully\n";
322  }
323  void linkedlist::update()
324  // updation of value
325  {
326  node*temp=NULL;
327  temp=head;
328  int no;
329  cout<<"Enter the position for updation :";
330  cin>>no;
331  for(int i=1;i<no;i++)
332  {
333  temp=temp->next;
334  //for loop to go to specific node
335  }
336  cout<<"Enter the id";
337  cin>>temp->id;
338  cout<<"Enter Name :";
339  cin>>temp->name;
340  cout<<"Enter Branch :";
341  cin>>temp->branch;
342  cout<<"Enter Cgpa :";
343  cin>>temp->cgpa;
344  cout<<"Record Updated Successfully\n";}
345  void linkedlist::search()
346  //searching any value
347  {
```

```
348  node*temp=NULL;
349  temp=head;
350  int no;
351  cout<<"Enter the position for Search :";
352  cin>>no;
353  for(int i=1;i<no-1;i++)
354  {
355  temp=temp->next;
356  }
357  cout<<"Search result :\n";
358  cout<<"id :"<<temp->next->id<<endl;
359  cout<<"Name :"<<temp->next->name<<endl;
360  cout<<"Branch :"<<temp->next->branch<<endl;
361  cout<<"Cgpa :"<<temp->next->cgpa<<endl;
362  }
363  void linkedlist :: swap(node*p1,node*p2)
364  {
365  int temp=p1->id;
366  p1->id=p2->id;
367  p2->id=temp;
368  char temp1[20];
369  strcpy(temp1,p1->name);
370  strcpy(p1->name,p2->name);
371  strcpy(p2->name,temp1);
372  char temp2[20];
373  strcpy(temp2,p1->branch);
374  strcpy(p1->branch,p2->branch);
375  strcpy(p2->branch,temp2);
376  float temp3=p1->cgpa;
377  p1->cgpa=p2->cgpa;
378  p2->cgpa=temp3;
379  }
380  void linkedlist::sort()
381  {
382  node*temp=NULL;
383  node*temp1=NULL;
384  node*p=NULL;
385  temp=head;
386  while(temp->next)
387  {
388  //swap function use for sorting
389  //swapping values from one node to another
390  //sort Function
391  //till temp->next becomes NULLp=temp;
392  temp1=temp->next;
393  while(temp1)
394  {
395  if(p->id > temp1->id)
396  {
397  p=temp1;
398  }
399  temp1=temp1->next;
400  }
401  swap(temp,p);
402  temp=temp->next;
403  //till temp1 becomes NULL
404  //swap function called
405  }
406  }
407  int main()
```

```
408  //main Function
409  {
410  int choice;
411  linkedlist l1;
412  cout<<"Placement Record List\n\n";
413  cout<<"Enter Student Detail Here :";
414  l1.create_list();
415  do
416  {
417  cout<<"\nYou can Perform Following operation\n";
418  cout<<"1.Insert Operation\n";
419  cout<<"2.Display\n";
420  cout<<"3.Delete Operation\n";
421  cout<<"4.Reverse Operation\n5.Update\n6.search\n7.Sort\nEnter you
choice :";
422  cin>>choice;
423  cout<<endl;
424  switch(choice)
425  {
426  case 1:
427  {
428  cout<<"\n1.Insert At Beginning\n2.Insert At last\n3.Insert in
Between[by pos]\n4.Insert by
429  val\nEnter Choice :";
430  cin>>choice;
431  switch(choice)
432  {
433  case 1:
434  {
435  l1.insert_begining();
436  //insert case}
437  break;
438  case 2:
439  {
440  l1.insert_last();
441  }
442  break;
443  case 3:
444  {
445  l1.insert_between();
446  }
447  break;
448  case 4:
449  {
450  l1.insert_inbetween();
451  }
452  break;
453  }
454  }
455  break;
456  case 2:
457  //display case
458  {
459  l1.display();
460  }
461  break;
462  case 3:
463  //delete case
464  {
465  cout<<"\n1.Delete At Beginning\n2.Delete At last\n3.Delete in
```

```
Between[by pos]\n4.Delete by
  val\nEnter Choice :";
  cin>>choice;
  switch(choice)
  {
  case 1:
  {
  l1.delete_begining();
  }
  break;
  case 2:
  {
  l1.delete_last();
  }break;
  case 3:
  {
  l1.delete_between();
  }
  break;
  case 4:
  {
  l1.delete_Byvalue();
  }
  break;
  }
  }
  break;
  case 4:
  {
  l1.reverse();
  }
  break;
  case 5:
  {
  l1.update();
  }
  break;
  case 6:
  {
  l1.search();
  }
  break;
  case 7:
  {
  l1.sort();
  }
  break;
  default :
  cout<<"Invalid choice";
  break;
  }
  //reverse case
  //update
  //search
  //sort
  //default}while(choice!=8);
  return 0;
522
```

OUTPUT :

Create list:

Inserting:

Inserting

```
To Continue press 1 else press 0 :1

Enter the id no:78
Enter the Name:Pradumna
Enter the Branch:Chemica;
Enter the cgpa:3.4

To Continue press 1 else press 0 :0

You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :1


1.Insert At Beginning
2.Insert At last
3.Insert in Between[by pos]
```

Inserting

```
4.Insert by val
Enter Choice :3
Enter the Position You Want to Enter3
Enter the id6
Enter Name :Dhiraj
Enter Branch :Mechanical
Enter Cgpa :8.8

You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :3


1.Delete At Beginning
2.Delete At last
```

Delete Operation

```
2.Insert At last
3.Insert in Between[by pos]
4.Insert by val
Enter Choice :3
Enter the Position You Want to Enter3
Enter the id6
Enter Name :Dhiraj
Enter Branch :Mechanical
Enter Cgpa :8.8

You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :3


1.Delete At Beginning
2.Delete At last
```

Revarse operation:

```
2.Delete At last
3.Delete in Between[by pos]
4.Delete by val
Enter Choice :4
Enter the value you want to delete78
3Record  Deleted Successfully

You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :5

Enter the position for updation :2
Enter the id36
Enter Name :Viraj
Enter Branch :IT
Enter Cgpa :1.11
```

```
You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :2


Displaying list :
Student id:6
Student name:Dhiraj
Student branch:Mechanical
Student Cgpa:8.8

Student id:11
Student name:Anuj
Student branch:Computer
Student Cgpa:9.4

Student id:36
Student name:Viraj
Student branch:IT
Student Cgpa:1.11

Student id:89
Student name:omkar
Student branch:Computer
Student Cgpa:9


You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :4
```

```
4.Delete by val
Enter Choice :4
Enter the value you want to delete78
3Record  Deleted Successfully

You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :5

Enter the position for updation :2
Enter the id36
Enter Name :Viraj
Enter Branch :IT
Enter Cgpa :1.11
Record Updated Successfully
```

Display :

```
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :6

Enter the position for Search :3
Search result :
id :6
Name :Dhiraj
Branch :Mechanical
Cgpa :8.8

You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
```

```
Record  Deleted Successfully

You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :2


Displaying list :
Student id:89
Student name:omkar
Student branch:Computer
Student Cgpa:9

Student id:36
Student name:Viraj
Student branch:IT
```

```
Student Cgpa:7.11

Student id:11
Student name:Anuj
Student branch:Computer
Student Cgpa:9.4

Student id:6
Student name:Dhiraj
Student branch:Mechanical
Student Cgpa:8.8


You can Perform Following operation
1.Insert Operation
2.Display
3.Delete Operation
4.Reverse Operation
5.Update
6.search
7.Sort
Enter you choice :
```