

Assignment 12

Class - SE IV

Roll NO - 21630

Batch - F4

DOS - 2/12/2020

Problem Statement:-

Write a program to implement priority queue in c++ using an inorder list to store the items in the queue. Create class that includes the data items (which should be template) and the priority (which should be int). The inorder list should contain these objects with operator \leq overloaded so that the item with the highest priority appear at the start of the list (which will make it relatively easy to retrieve the higher item).

Objectives:-

To understand insertion and deletion of nodes according to priorities.

Theory:-

Priority Queue:-

Priority queue is an extension of the queue that is similar to the queue in the certain aspects and yet it differs from the ordinary queue in the following points.

1. Each item in the priority queue is associated with a priority.

2. The item with the highest priority is the first item to be removed from the queue.
3. If more than one item has the same priority then that order in the queue is considered.

Operations on priority queue:-
(Implementation using array)

1. Initialize() = Make the queue empty()
2. Empty() - Determine if queue is empty or not
3. Full() - Determine if queue is full or not
4. enqueue() - Insert an element as per its priority.
5. Dequeue() - Delete the front element (front element will have high priority)
6. Print() = prints all elements in the queue.

* Application:-

1. operating system load balancing and interrupt handles.
2. Routing.
3. Hospital Emergency
4. Dijkstra's shortest path algorithm.

* Algorithm/Pseudocode-1

* ADT representation of class Node:-

```

template <class T>
class Node
{
    T data;    // data field
    int priority; // priority of item
    Node<T> next; // next pointer.

    template <class T1>
    friend class PriorityQueue;
    bool operator <= (const Node & n1)
    {
        return this->priority <= n1.priority;
    }
};

```

* ADT representation of class PriorityQueue

```

template <class T>
class PriorityQueue
{
    Node<T> * front; // front pointer of queue
    enqueuePri(T); // insert element by priority
    dequeuePri(); // delete element from front
    show();
};

```

* Procedure enqueuePr@ (Td int p)

node<T> *temp = new Node<T>

if (front == NULL)

front = temp;

front->next = NULL;

else

node<T> *r = front;

if (front == temp)

{

temp->next = front;

front = temp;

}

else

{

node<T> *q = new Node<T>;

q = r->next;

while (q != NULL && temp != q)

{

r = r->next;

q = r->next;

}

temp->next = r->next;

r->next = temp;

}

end else

END procedure

* Procedure dequeuePrQ()

```

Node<T> * temp = front
if (front == NULL)
    Display "Queue is empty";
front = front->next;
delete temp;

```

END procedure

* Procedure show()

```

node<T> * P = front;
if (front == NULL)
    Display "Queue is empty";
while (P != NULL)
    Display P->data, P->priority;
    P = P->next;
END while

```

END Procedure

* Complexity:-

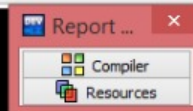
Function	Time Complexity
1. EnqueuePrQ()	$O(\log n)$
2. dequeuePrQ()	$O(\log n)$

* Test Cases:-

No.	Description	Input	Expected o/p	Actual o/p	Result
1.	1. Insert	ch = 1	item priority	item priority	
	2. Delete	item = 1	2 10	2 10	
	3. Show	priority = 5	1 5	1 5	Pass
	4. Exit	item = 2	3 2	3 2	
		priority = 10			
		item = 3			
		priority = 2			
		ch = 3			
2.	1. Insert	ch = 2	2 Deleted	2 Deleted	
	2. Delete	ch = 3	item priority	item priority	Pass
	3. Show		1 5	1 5	
	4. Exit		3 2	3 2	

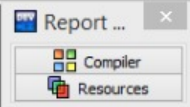
Conclusion:-

We learnt to implement insertion and deletion of nodes by priorities.



```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4  template < class T >
5  class Node
6  {
7  template < class K >
8  friend class Priority_Queue;
9  private:
10 T JobId;
11 int priority;
12 Node < T >* next;
13 public:
14 Node()
15 {
16 next = NULL;
17 priority = -1;
18 }
19 Node( T id , int prio)
20 {
21 next = NULL;
22 JobId = id;
23 priority = prio;
24 }
25 bool operator <= ( Node &obj)
26 {
27 return (this->priority <= obj.priority);
28 }
29 };
30 template < class T >
31 class Priority_Queue
32 {
33 private:
34 Node < T >* front;
35 public:
36 Priority_Queue()
```

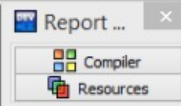
Activate Windows
Go to PC settings to activate Windows.



```
36 Priority_Queue()
37 {
38     front = NULL;
39 }
40 void insert( T id, int priority )
41 {
42     Node < T > *tmp = new Node<T>(id , priority);
43     if(front == NULL)
44     {
45         front = tmp;
46     }
47     else
48     {
49         Node< T > *current = front;
50         Node< T > *prev = NULL;
51         if (priority > current->priority)
52         {
53             tmp->next = current;
54             front = tmp;
55         }
56         else
57         {
58             while((current!= NULL) && ((*tmp) <= (*current)))
59             {
60                 prev = current;
61                 current = current->next;
62             }
63             tmp->next = current;
64             prev->next = tmp;
65         }
66     }
67 }
68 void del()
69 {
70     Node < T > *tmp;
71     if( front == NULL )
```

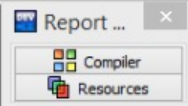
Activate Windows
Go to PC settings to activate Windows.


```
72 {
73     cout << "\t\t Queue Underflow...!!!" << endl;
74 }
75 else
76 {
77     tmp = front;
78     cout << "\t\t Deleted item is = " << tmp->JobId << endl;
79     front = front->next;
80     delete tmp;
81 }
82 }
83 void display()
84 {
85     Node < T > *ptr;
86     ptr = front;
87     if( front == NULL )
88     {
89         cout<<"\t\t Queue is empty...!!!" << endl;
90     }
91     else
92     {
93         cout<<"\t\t ===Queue ===" << endl;
94         cout << endl;
95         cout<<"\t\t" << left << setw(20)<< "Priority" << left <<
96         setw(20)<< "Item" << endl;
97         cout << endl;
98         while( ptr != NULL )
99         {
100             cout << "\t\t" << left << setw(20) << ptr->priority
101             << left << setw(20) << ptr->JobId<< endl;
102             ptr = ptr->next;
103         }
104     }
105 }
106 };
107 int main()
```



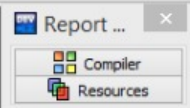
Activate Windows
Go to PC settings to activate Windows.

```
109 int choice;
110 Priority_Queue < int > pq_int;
111 Priority_Queue < float > pq_float;
112 Priority_Queue < char > pq_char;
113 do
114 {
115     cout << " === MENU ===" << endl;
116     cout << "\t 1. Integer Data Priority Queue" << endl;
117     cout << "\t 2. Float Data Priority Queue" << endl;
118     cout << "\t 3. Char Data Priority Queue" << endl;
119     cout << "\t 4. EXIT" << endl;
120     cout << endl;
121     cout << "Enter choice to continue = ";
122     cin >> choice;
123     cout << endl;
124     switch( choice )
125     {
126     case 1:
127     {
128         int choice1, item1, priority1;
129         do
130         {
131             cout << endl;
132             cout << "\t === Choices ===" << endl;
133             cout << "\t\t 1.Insert int item \n";
134             cout << "\t\t 2.Delete int item \n";
135             cout << "\t\t 3.Display int item \n";
136             cout << "\t\t 4.Quit \n";
137             cout << endl;
138             cout << "\t Enter your choice : ";
139             cin >> choice1;
140             cout << endl;
141             switch(choice1)
142             {
143             case 1:
144             {
```



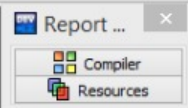
Activate Windows
Go to PC settings to activate Windows.


```
145 cout<<"\t\t Enter the int item value to insert in the queue = ";
146 cin>>item1;
147 cout << endl;
148 cout<<"\t\t Enter its priority = ";
149 cin>>priority1;
150 pq_int.insert(item1, priority1);
151 }
152 break;
153 case 2:
154 {
155 pq_int.del();
156 }
157 break;
158 case 3:
159 {
160 pq_int.display();
161 }
162 break;
163 case 4:
164 {
165 cout << "\t\t Thank You...!!!" <<endl;
166 cout << endl;
167 cout << endl;
168 }
169 break;
170 default :
171 {
172 cout << "\t\t ERROR...Entered Invalid Choice !!!" << endl;
173 cout << endl;
174 cout << endl;
175 }
176 }
177 }while(choice1 != 4);
178 }
179 break;
180 case 2:
```



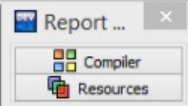
Activate Windows
Go to PC settings to activate Windows.

```
181 {
182     int choice2, priority2;
183     float item2;
184     do
185     {
186         cout << endl;
187         cout << "\t === Choices ===" << endl;
188         cout << "\t\t 1.Insert float item \n";
189         cout << "\t\t 2.Delete float item \n";
190         cout << "\t\t 3.Display float item \n";
191         cout << "\t\t 4.Quit \n";
192         cout << endl;
193         cout << "\t Enter your choice : ";
194         cin >> choice2;
195         cout << endl;
196         switch(choice2)
197         {
198             case 1:
199             {
200                 cout << "\t\t Enter the float item value to insert in the queue = ";
201                 cin >> item2;
202                 cout << endl;
203                 cout << "\t\t Enter its priority = ";
204                 cin >> priority2;
205                 pq_float.insert(item2, priority2);
206             }
207             break;
208             case 2:
209             {
210                 pq_float.del();
211             }
212             break;
213             case 3:
214             {
215                 pq_float.display();
216             }
```



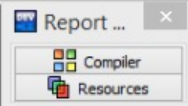
Activate Windows
Go to PC settings to activate Windows.


```
218 case 4:
219 {
220 cout << "\t\t Thank You...!!!" << endl;
221 cout << endl;
222 cout << endl;
223 }
224 break;
225 default :
226 {
227 cout << "\t\t ERROR...Entered Invalid Choice !!!" << endl;
228 cout << endl;
229 cout << endl;
230 }
231 }
232 }while(choice2 != 4);
233 }
234 break;
235 case 3:
236 {
237 int choice3, priority3;
238 char item3;
239 do
240 {
241 cout << endl;
242 cout << "\t === Choices ===" << endl;
243 cout<<"\t\t 1.Insert char item \n";
244 cout<<"\t\t 2.Delete char item \n";
245 cout<<"\t\t 3.Display char item \n";
246 cout<<"\t\t 4.Quit \n";
247 cout << endl;
248 cout<<"\tEnter your choice : ";
249 cin>>choice3;
250 cout << endl;
251 switch(choice3)
252 {
253 case 1:
```



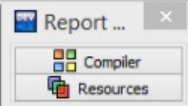
Activate Windows
Go to PC settings to activate Windows.

```
254 {
255     cout<<"\t\t Enter the char item value to insert in the queue = ";
256     cin >> item3;
257     cout << endl;
258     cout<<"\t\t Enter its priority = ";
259     cin>>priority3;
260     pq_char.insert(item3, priority3);
261 }
262 break;
263 case 2:
264 {
265     pq_char.del();
266 }
267 break;
268 case 3:
269 {
270     pq_char.display();
271 }
272 break;
273 case 4:
274 {
275     cout << "\t\t Thank You...!!!" <<
276     endl;
277     cout << endl;
278     cout << endl;
279 }
280 break;
281 default :
282 {
283     cout << "\t\t ERROR...Entered Invalid Choice !!!" << endl;
284     cout << endl;
285     cout << endl;
286 }
287 }
288 }while( choice3 != 4 );
289 }
```



Activate Windows
Go to PC settings to activate Windows.


```
273     case 4:
274     {
275     cout << "\t\t Thank You...!!!" <<
276     endl;
277     cout << endl;
278     cout << endl;
279     }
280     break;
281     default :
282     {
283     cout << "\t\t ERROR...Entered Invalid Choice !!!" << endl;
284     cout << endl;
285     cout << endl;
286     }
287     }
288     }while( choice3 != 4 );
289     }
290     break;
291     case 4:
292     {
293     cout << "\t\t Thank You...!!!" << endl;
294     cout << endl;
295     exit ( 0 );
296     }
297     break;
298     default:
299     {
300     cout << "\t\t ERROR...Entered Invalid Choice !!!" <<
301     endl;
302     cout << endl;
303     }
304     }
305     }while(choice != 4);
306     return 0;
307 }
```



Activate Windows
Go to PC settings to activate Windows.

```
=== MENU ===
```

1. Integer Data Priority Queue
2. Float Data Priority Queue
3. Char Data Priority Queue
4. EXIT

```
Enter choice to continue = 1
```

```
=== Choices ===
```

- 1.Insert int item
- 2.Delete int item
- 3.Display int item
- 4.Quit

```
Enter your choice : 1
```

```
Enter the int item value to insert in the queue = 124
```

```
Enter its priority = 2
```

```
=== Choices ===
```

- 1.Insert int item
- 2.Delete int item
- 3.Display int item
- 4.Quit

```
Enter your choice : 3
```

```
===Queue ===
```

Priority	Item
2	124

```
=== Choices ===
```

- 1.Insert int item
- 2.Delete int item
- 3.Display int item
- 4.Quit

```
Enter your choice : 1
```

```
Enter the int item value to insert in the queue = 45
```

```
Enter its priority = 3
```

```
=== Choices ===
```

- 1.Insert int item
- 2.Delete int item
- 3.Display int item
- 4.Quit

```
Enter your choice : 3
```

```
===Queue ===
```

```
3      45
2      124
```

```
=== Choices ===
```

- 1.Insert int item
- 2.Delete int item
- 3.Display int item
- 4.Quit

```
Enter your choice : 1
```

```
Enter the int item value to insert in the queue = 84
```

```
Enter its priority = 1
```

```
=== Choices ===
```

- 1.Insert int item
- 2.Delete int item
- 3.Display int item
- 4.Quit

```
Enter your choice : 3
```

```
===Queue ===
```

Priority	Item
3	45
2	124
1	84

```
=== Choices ===
```

- 1.Insert int item
- 2.Delete int item
- 3.Display int item
- 4.Quit

```
Enter your choice : 4
```

```
Thank You...!!!
```

```
=== MENU ===
```

- 1. Integer Data Priority Queue
- 2. Float Data Priority Queue
- 3. Char Data Priority Queue
- 4. EXIT

```
Enter choice to continue = 4
```

```
Thank You...!!!
```

```
Process exited after 2928 seconds with return value 0
```

```
Press any key to continue . . .
```