**Priority Queue ..** Artful

# Priority queues are a type of container adapters, specifically designed such that the first element of the queue is the greatest of all elements in the queue and elements are in non decreasing order(hence we can see that each element of the queue has a priority{fixed order}).

Functions:  
empty() – Returns whether the queue is empty  
size() – Returns the size of the queue  
top() – Returns a reference to the top most element of the queue  
push(g) – Adds the element ‘g’ at the end of the queue  
pop() – Deletes the first element of the queue

Swap() – It is used to swap the contents of one pQueue with another pQueue.

void showpq(priority\_queue <int,vector<int>,greater<int>> gq)

{

priority\_queue <int,vector<int>,greater<int>> g = gq;

while (!g.empty())

{

cout << " " << g.top();

g.pop();

}

cout <<endl;

}

int main ()

{

priority\_queue <int,vector<int>,greater<int>> gquiz;

gquiz.push(10);

gquiz.push(30);

gquiz.push(20);

gquiz.push(1);

gquiz.push(5);

cout << "The priority queue gquiz is : ";

showpq(gquiz);

cout << "\ngquiz.size() : " << gquiz.size();

cout << "\ngquiz.top() : " << gquiz.top();

cout << "\ngquiz.pop() : ";

gquiz.pop();

showpq(gquiz);

priority\_queue<int> foo,bar;

foo.push (15);

foo.push(30);

foo.push(10);

bar.push (101);

bar.push(202);

foo.swap(bar);

//bar.swap(foo); //same as previous

cout << "size of foo: " << foo.size() <<endl;

cout << "size of bar: " << bar.size() <<endl;

return 0;

}

Output

The priority queue gquiz is : 1 5 10 20 30

gquiz.size() : 5

gquiz.top() : 1

gquiz.pop() : 5 10 20 30

size of foo: 2

size of bar: 3