JASKIRAT SINGH (2020CSC1008)

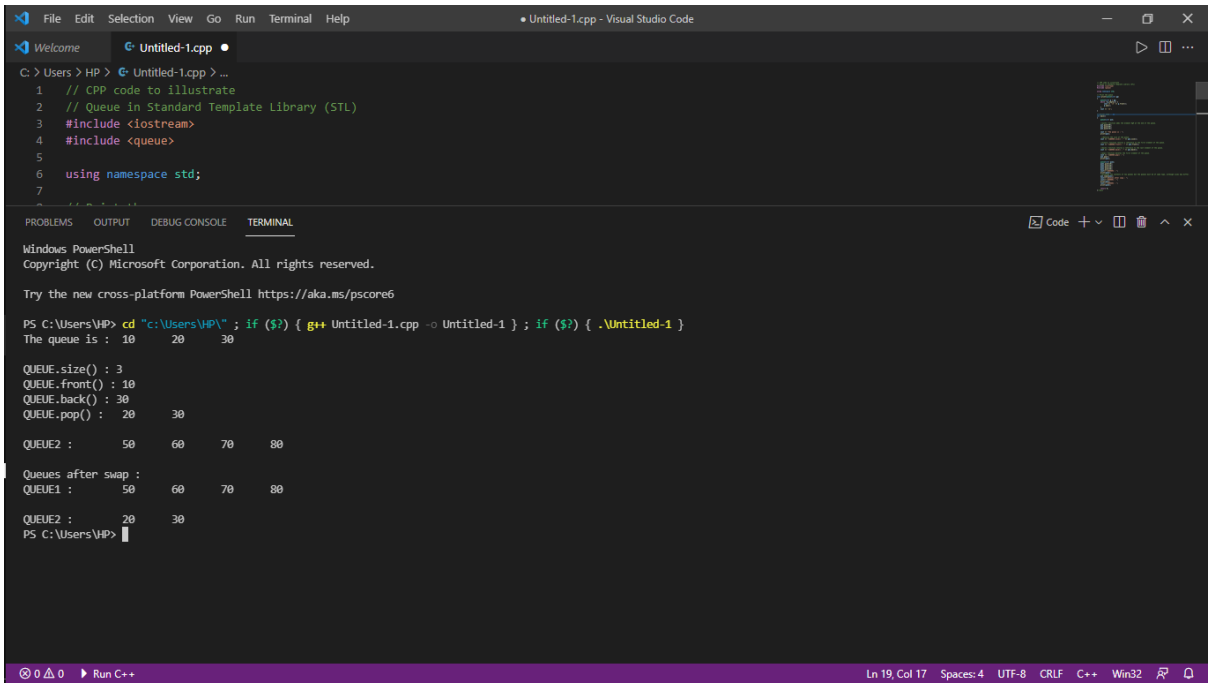
CPP code to illustrate Queue in Standard Template Library (STL)

```
#include <iostream>
#include <queue>

using namespace std;

// Print the queue
void print(queue<int> gq)
{
	queue<int> g = gq;
	while (!g.empty()) {
		cout << '\t' << g.front();
		g.pop();
	}
	cout << '\n';
}

// Driver Code
int main()
{
	queue<int> que;

   //push() function adds the element ‘g’ at the end of the queue.
	que.push(50);
	que.push(60);
```

```
        que.push(70);

        cout << "The queue is : ";
        print(que);

 //Returns the size of the queue.
        cout << "\nQUEUE.size() : " << que.size();

//front() function returns a reference to the first element of the queue.
        cout << "\nQUEUE.front() : " << que.front();

//back() function returns a reference to the last element of the queue.
        cout << "\nQUEUE.back() : " << que.back();

//pop() function deletes the first element of the queue.
        cout << "\nQUEUE.pop() : ";
        que.pop();
        print(que);

queue<int> que1;
que1.push(10);
        que1.push(20);
        que1.push(30);
que1.push(40);
cout<<"\nQUEUE2 : ";
print(que1);
//Exchange the contents of two queues but the queues must be of same type, although sizes may differ.
que.swap(que1);
cout<<"\nQueues after swap : ";
cout<<"\nQUEUE1 : ";
```

```
    print(que);
    cout<<"\nQUEUE2 : ";
    print(que1);

        return 0;
}
```

output



Implement a templated Queue using Array.

```
#include <iostream>
using namespace std;

// A class to represent a queue
```

```
template <class X>
class queue
{
   X *arr;      // array to store queue elements
   int capacity; // maximum capacity of the queue
   int front;    // front points to the front element in the queue (if any)
   int rear;     // rear points to the last element in the queue

public:
   queue(int size);    // constructor
   ~queue() { delete[] arr; } //destructor

   void dequeue();
   void enqueue(X x);
   X peek();
   int size();
   bool isEmpty();
   bool isFull();
};

// Constructor to initialize a queue
template <class X>
queue<X>::queue(int size)
{
   arr = new X[size];
   capacity = size;
   front = -1;
   rear = -1;
}

// Template function to dequeue the front element
```

```
template <class X>
void queue<X>::dequeue()
{
   // check for queue underflow
   if (isEmpty())
   {
      cout << "Underflow";
      exit(EXIT_FAILURE);
   }

   front++;
   cout << "Removing " << arr[front] << endl;
}

// Template function to add an item to the queue
template <class X>
void queue<X>::enqueue(X item)
{
   // check for queue overflow
   if (isFull())
   {
      cout << "Overflow";
      exit(EXIT_FAILURE);
   }

   cout << "Inserting " << item << endl;

   rear++;
   arr[rear] = item;
}
```

```
// Template function to check if the queue is empty or not
template <class X>
bool queue<X>::isEmpty()
{
   if (rear == front)
      return true;
   else
      return false;
}

// Template function to check if the queue is full or not
template <class X>
bool queue<X>::isFull()
{
   if (rear == capacity - 1)
      return true;
   else
      return false;
}

int main()
{
   // create a queue of capacity 4
   queue<string> q(4);

   q.enqueue("p");
   q.enqueue("q");
   q.enqueue("r");

   q.dequeue();
```

```
    q.enqueue("s");

    if(q.isFull())
    {
        cout << "The queue is full\n";
    }
    else
    {
        cout << "The queue is not full\n";
    }

    q.dequeue();
    q.dequeue();
    q.dequeue();

    if (q.isEmpty())
    {
        cout << "The queue is empty\n";
    }
    else
    {
        cout << "The queue is not empty\n";
    }

    return 0;
}
```

output

File Edit Selection View Go Run Terminal Help — Code.cpp - Visual Studio Code

Welcome | Untitled-1.cpp | Code.cpp

C: > Users > HP > Code.cpp > main()

```cpp
}

// Template function to check if the queue is full or not
template <class X>
bool queue<X>::isFull()
{
    if (rear == capacity - 1)
        return true;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\HP> cd "c:\Users\HP\" ; if ($?) { g++ Code.cpp -o Code } ; if ($?) { .\Code }
Inserting p
Inserting q
Inserting r
Removing p
Inserting s
The queue is full
Removing q
Removing r
Removing s
The queue is empty
PS C:\Users\HP>
```

0 0 Run C++ Ln 97, Col 17 Spaces: 4 UTF-8 CRLF C++ Win32