

```

1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1C
5  * SECTION    : MW 5pm
6  * Assign #8  : Templates
7  * DUE DATE   : 25 March 2020
8  *****/
9
10 #include "MyHeader.h"
11
12 int main(){
13     /*****
14      * Queuing and deleting the objects to the stacks
15      *
16      * using the templates and the functions, the user can queue or
17      * delete objects from the head of the stacks and delete from the
18      * end of the stack. Also, using the class function, the program
19      * determines if the stack is empty or full
20      *
21      * INPUT: N/A
22      *
23      * OUTPUT: outputs step by step of adding the objects the objects
24      *          or deleting the objects from the stack, also outputs
25      *          if the
26      *
27      *****/
28
29     cout << "/*****\n"
30         << " * Queuing and deleting the objects to the stacks\n"
31         << " * _____\n"
32         << " * using the templates and the functions, the user can queue or\n"
33         << " * delete objects from the head of the stacks and delete from the\n"
34         << " * end of the stack. Also, using the class function, the program\n"
35         << " * determines if the stack is empty or full\n"
36         << " * _____\n"
37         << " * INPUT: N/A\n"
38         << " * \n"
39         << " * OUTPUT: outputs step by step of adding the objects the objects\n"
40         << " *          or deleting the objects from the stack, also outputs\n"
41         << " *          if the\n"
42         << " * \n"
43         << " *****/\n\n";
44
45     //creating a Character queue
46     Queue<string> q(10);
47
48     //Queuing the string characters to the head of the stack
49     q.enqueue("a");
50     q.enqueue("b");
51     q.enqueue("c");
52     q.enqueue("d");
53     q.enqueue("e");
54     q.enqueue("f");
55
56     //deleting the string characters from bottom of the stack
57     q.dequeue();
58     q.dequeue();
59     q.dequeue();

```

```

60 //Queuing the string characters to the head of the stack
61 q.enqueue("g");
62 q.enqueue("h");
63 q.enqueue("i");
64 q.enqueue("j");
65 //deleting the string characters from bottom of the stack
66 q.dequeue();
67 q.dequeue();
68 q.dequeue();
69 q.dequeue();
70 q.dequeue();
71 q.dequeue();
72 q.dequeue();
73 q.dequeue();
74 cout<<"Front of queue: "<<q.front()<<"\n";
75
76
77 //creating an integer queue
78 Queue<int> d(10);
79
80 //Queuing the integers to the head of the stack
81 d.enqueue(1);
82 d.enqueue(2);
83 d.enqueue(3);
84 d.enqueue(4);
85 d.enqueue(5);
86 d.enqueue(6);
87 //deleting the integers from bottom of the stack
88 d.dequeue();
89 d.dequeue();
90 //Queuing the integers to the head of the stack
91 d.enqueue(7);
92 d.enqueue(8);
93 d.enqueue(9);
94 //deleting the integers from bottom of the stack
95 d.dequeue();
96 d.dequeue();
97 cout<<"Front of queue: "<<d.front()<<"\n\n";
98
99
100 //creating an Double queue
101 Queue<double> i(10);
102
103 //Queuing the doubles to the head of the stack
104 i.enqueue(1.1);
105 i.enqueue(2.1);
106 i.enqueue(3.3);
107 i.enqueue(4.4);
108 i.enqueue(5.5);
109 i.enqueue(6.6);
110 //deleting the doubles from bottom of the stack
111 i.dequeue();
112 //Queuing the doubles to the head of the stack
113 i.enqueue(7.7);
114 i.enqueue(8.8);
115 //deleting the doubles from bottom of the stack
116 i.dequeue();
117 i.dequeue();
118 i.dequeue();

```

```
119 i.dequeue();
120 i.dequeue();
121 cout<<"Front of queue: "<<i.front()<<"\\n\\n";
122
123
124 return 0;
125 }
126
127
128
129
130
```