

Name : Prem Sah
PRN : 123B1B234
Div : D

Assignment no. 9

A) Implement a restaurant waitlist system using the queue data structure. Restaurant waitlist provide following facility:

- a. Add Party to Waitlist**
- b. Seat Party**
- c. Display Waitlist**

```
#include <iostream>
#include <queue>
#include <string>
class Party {
public:
    std::string name;
    int size;

    Party(std::string partyName, int partySize) : name(partyName), size(partySize) {}

    void display() const {
        std::cout << name << " (Party Size: " << size << ")" << std::endl;
    }
};
class Waitlist {
private:
    std::queue<Party> queue;

public:
    void addParty(const std::string& name, int size) {
        Party newParty(name, size);
        queue.push(newParty);
        std::cout << "Added " << name << " to the waitlist." << std::endl;
    }
}
```

```

void seatParty() {
    if (!queue.empty()) {
        Party seatedParty = queue.front();
        queue.pop();
        std::cout << "Seated " << seatedParty.name << "." << std::endl;
    } else {
        std::cout << "No parties on the waitlist." << std::endl;
    }
}

```

```

void displayWaitlist() const {
    if (queue.empty()) {
        std::cout << "The waitlist is currently empty." << std::endl;
    } else {
        std::cout << "Current Waitlist:" << std::endl;
        std::queue<Party> tempQueue = queue; // Copy the queue to display without
modifying
        int index = 1;
        while (!tempQueue.empty()) {
            tempQueue.front().display();
            tempQueue.pop();
            index++;
        }
    }
}
};

```

```

int main() {
    Waitlist waitlist;
    waitlist.addParty("Smith", 4);
    waitlist.addParty("Johnson", 2);
    waitlist.addParty("Williams", 5);
    waitlist.displayWaitlist();
    waitlist.seatParty();
    waitlist.displayWaitlist();
}

```

```
    return 0;  
}
```

OUTPUT :

**Added Smith to the waitlist.
Added Johnson to the waitlist.
Added Williams to the waitlist.
Current Waitlist:
Smith (Party Size: 4)
Johnson (Party Size: 2)
Williams (Party Size: 5)
Seated Smith.
Current Waitlist:
Johnson (Party Size: 2)
Williams (Party Size: 5)**

main.cpp

```
1 #include <iostream>
2 #include <queue>
3 #include <string>
4 class Party {
5 public:
6     std::string name;
7     int size;
8     Party(std::string partyName, int partySize) : name(partyName), size(partySize) {}
9     void display() const {
10         std::cout << name << " (Party Size: " << size << ")" << std::endl;
11     }
12 };
13 class Waitlist {
14 private:
15     std::queue<Party> queue;
16 public:
17     void addParty(const std::string& name, int size) {
18         Party newParty(name, size);
19         queue.push(newParty);
20         std::cout << "Added " << name << " to the waitlist." << std::endl;
21     }
22     void seatParty() {
23         if (!queue.empty()) {
24             Party seatedParty = queue.front();
25             queue.pop();
26             std::cout << "Seated " << seatedParty.name << "." << std::endl;
27         } else {
28             std::cout << "No parties on the waitlist." << std::endl;
29         }
30     }
31     void displayWaitlist() const {
32         if (queue.empty()) {
33             std::cout << "The waitlist is currently empty." << std::endl;
34         } else {
35             std::cout << "Current Waitlist:" << std::endl;
36             std::queue<Party> tempQueue = queue; // Copy the queue to display without modifying
37             int index = 1;
38             while (!tempQueue.empty()) {
39                 tempQueue.front().display();
40                 tempQueue.pop();
41                 index++;
42             }
43         }
44     }
45 };
46 int main() {
47     Waitlist waitlist;
48     waitlist.addParty("Smith", 4);
49     waitlist.addParty("Johnson", 2);
50     waitlist.addParty("Williams", 5);
51     waitlist.displayWaitlist();
52     waitlist.seatParty();
53     waitlist.displayWaitlist();
54     return 0;
55 }
```

Output

```
/tmp/hKGp38na83.o
```

```
Added Smith to the waitlist.
```

```
Added Johnson to the waitlist.
```

```
Added Williams to the waitlist.
```

```
Current Waitlist:
```

```
Smith (Party Size: 4)
```

```
Johnson (Party Size: 2)
```

```
Williams (Party Size: 5)
```

```
Seated Smith.
```

```
Current Waitlist:
```

```
Johnson (Party Size: 2)
```

```
Williams (Party Size: 5)
```

```
=== Code Execution Successful ===
```