1 **// A Non-Generic Queue. It can only queue SuudentV4 Objects**

2

3 **public** **class** Queue

4 {

5 **private** **int** size;

6 **private** **int** numOfNodes = 0;

7 **private** **int** front = 0;

8 **private** **int** rear = 0;

9 **private** StudentV4[] data;

10

11 **public** Queue(**int** n)

12 {

13 size = n;

14 data = **new** StudentV4[n];

15 }

16

17 **public** **boolean** enQueue(StudentV4 newItem) **//add a StudentV4 object**

18 {

19 **if**(numOfNodes == size) **//The queue is full**

20 {

21 **return** **false**;

22 }

23 **else** **//add the object to the structure**

24 {

25 numOfNodes = numOfNodes + 1;

26 data[rear] = newItem;

27 rear = (rear + 1) % size;

28 **return** **true**;

29 }

30 }

31

32 **public** StudentV4 deQueue( ) **//fetch and delete a StudentV4 object**

33 {

34 **int** frontLocation;

35 **if**(numOfNodes == 0) **//The queue is empty**

36 {

37 **return** **null**;

38 }

39 **else** **//return an object from the structure**

40 {

41 frontLocation = front;

42 front = (front + 1) % size;

43 numOfNodes = numOfNodes - 1;

44 **return** data[frontLocation];

45 }

46 }

47 }

**Figure 13.19 The class Queue.**