Assignment . 11

#include <iostream>

using namespace std;

#define MAX 10

class JobQueue {

int front, rear;

int queue[MAX];

public:

JobQueue() {

front = -1;

rear = -1;

}

bool isFull() {

return (rear == MAX - 1);

}

bool isEmpty() {

return (front == -1 || front > rear);

}

void addJob(int job) {

if (isFull()) {

cout << "Queue is full. Cannot add more jobs." << endl;

return;

}

if (front == -1) front = 0;

queue[++rear] = job;

cout << "Job " << job << " added to the queue." << endl;

}

void deleteJob() {

if (isEmpty()) {

cout << "Queue is empty. No jobs to delete." << endl;

return;

}

cout << "Job " << queue[front++] << " deleted from the queue." << endl;

if (front > rear) {

front = rear = -1;

}

}

void displayQueue() {

if (isEmpty()) {

cout << "Queue is empty." << endl;

return;

}

cout << "Jobs in the queue: ";

for (int i = front; i <= rear; i++) {

cout << queue[i] << " ";

}

cout << endl;

}

};

int main() {

JobQueue jobQueue;

int choice, job;

do {

cout << "1. Add Job\n2. Delete Job\n3. Display Jobs\n4. Exit\n";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter job number to add: ";

cin >> job;

jobQueue.addJob(job);

break;

case 2:

jobQueue.deleteJob();

break;

case 3:

jobQueue.displayQueue();

break;

}

} while (choice != 4);

return 0;

}

1. Add Job

2. Delete Job

3. Display Jobs

4. Exit

Enter job number to add: 101

Job 101 added to the queue.

Enter job number to add: 102

Job 102 added to the queue.

Jobs in the queue: 101 102

Job 101 deleted from the queue.

Jobs in the queue: 102

1. Add Job

2. Delete Job

3. Display Jobs

4. Exit

Enter job number to add: 103

Job 103 added to the queue.

1. Add Job

2. Delete Job

3. Display Jobs

4. Exit

Enter job number to add: 104

Job 104 added to the queue.

1. Add Job

2. Delete Job

3. Display Jobs

4. Exit

Enter job number to add: 110

Queue is full. Cannot add more jobs.

1. Add Job

2. Delete Job

3. Display Jobs

4. Exit