```
#include <iostream>

#include <queue>

#include <string>


using namespace std;


struct Job {

    int jobID;

    string jobDescription;


    Job(int id, string desc) : jobID(id), jobDescription(desc) {}

};


class JobQueue {

private:

    queue<Job> q;


public:

    void addJob(int id, string description) {

        Job newJob(id, description);

        q.push(newJob);

        cout << "Job added: " << description << " (ID: " << id << ")" << endl;

    }


    void deleteJob() {

        if (q.empty()) {

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

        } else {

            Job job = q.front();

            q.pop();

            cout << "Job deleted: " << job.jobDescription << " (ID: " << job.jobID << ")" << endl;

        }
    }

    void displayJobs() {
        if (q.empty()) {
            cout << "Queue is empty." << endl;
        } else {
            queue<Job> temp = q;
            cout << "Current jobs in the queue:" << endl;
            while (!temp.empty()) {
                Job job = temp.front();
                temp.pop();
                cout << "Job ID: " << job.jobID << ", Description: " << job.jobDescription << endl;
            }
        }
    }
};

int main() {
    JobQueue jq;
    int choice, id;
    string description;

    while (true) {
        cout << "\nJob Queue Menu:" << endl;
        cout << "1. Add Job" << endl;
        cout << "2. Delete Job" << endl;
        cout << "3. Display Jobs" << endl;
        cout << "4. Exit" << endl;
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter job ID: ";
                cin >> id;
                cin.ignore();  // to ignore the newline character left by cin
                cout << "Enter job description: ";
                getline(cin, description);
                jq.addJob(id, description);
                break;
            case 2:
                jq.deleteJob();
                break;
            case 3:
                jq.displayJobs();
                break;
            case 4:
                cout << "Exiting program." << endl;
                return 0;
            default:
                cout << "Invalid choice. Please try again." << endl;
        }
    }
}
```