```cpp
#include <iostream>
#include <fstream>
#include <cstring>
using namespace std;

class Student {
private:
    int rno;
    char sname[20];
    char subject[20];
    int sub_code;
    float internal;
    float university;

public:
    Student() {
        rno = 0;
        strcpy(sname, "\0");
        strcpy(subject, "\0");
        sub_code = 0;
        internal = 0.0;
        university = 0.0;
    }
    int acceptRollno() {
        return rno;
    }
    void acceptDetails() {
        cout << "\nEnter Details :";
        cout << "\nRoll no: ";
        cin >> rno;
        cout << "\nName: ";
        cin >> sname;
        cout << "\nSubject Code:";
        cin >> sub_code;
        cout << "\nSubject name :";
        cin >> subject;
        cout << "\nInternal Assessment Marks :";
        cin >> internal;
        cout << "\nUniversity Exam Marks :";
        cin >> university;
    }
    void putData() {
        cout << "\nRoll No.: " << rno;
        cout << "\tName: " << sname;
        cout << "\nSubject Code: " << sub_code;
        cout << "\nSubject Name: " << subject;
        cout << "\nInternal Assessment Marks: " << internal;
        cout << "\nUniversity Exam Marks: " << university << "\n\n";
    }
};

class Operations {
private:
    ifstream fin;
    ofstream fout;
    fstream fs;

public:
    void addRecord();
    void show();
    void search(int);
    int delRecord(int);
    int edit(int);
};

void Operations::addRecord() {
    Student r;
    r.acceptDetails();
    fout.open("Data.txt", ios::app);
    fout.write((char *)&r, sizeof(r));
    fout.close();
}
```

```
void Operations::show() {
    Student r;
    fin.open("Data.txt");
    while (fin.read((char *)&r, sizeof(r))) {
        r.putData();
    }
    fin.close();
}

void Operations::search(int rno) {
    Student r;
    int flag = 0;
    fin.open("Data.txt");
    while (fin.read((char *)&r, sizeof(r))) {
        if (r.acceptRollno() == rno) {
            flag = 1;
            break;
        }
    }
    fin.close();
    if (flag == 1) {
        cout << "\nStudent Found:";
        r.putData();
    } else {
        cout << "\nStudent not Found ";
    }
}

int Operations::delRecord(int rno) {
    Student r;
    int flag = 0;
    fin.open("Data.txt");
    fout.open("Temp.txt", ios::app);
    while (fin.read((char *)&r, sizeof(r))) {
        if (r.acceptRollno() == rno) {
            flag = 1;
        } else {
            fout.write((char *)&r, sizeof(r));
        }
    }
    fin.close();
    fout.close();
    remove("Data.txt");
    rename("Temp.txt", "Data.txt");
    return flag;
}

int Operations::edit(int rno) {
    Student r;
    int flag = 0;
    fs.open("Data.txt", ios::in | ios::out);
    while (fs.read((char *)&r, sizeof(r))) {
        if (r.acceptRollno() == rno) {
            flag = 1;
            cout << "\nEnter New Details ";
            r.acceptDetails();
            fs.seekp(-sizeof(r), ios::cur);
            fs.write((char *)&r, sizeof(r));
        }
    }
    fs.close();
    return flag;
}

int main() {
    Operations f;
    int ch, n, i, flag = 0;
    do {
        cout << "\n\n\t------MENU-------";
        cout << "\n1. Build a Master Table\n2. List a Table\n3. Insert a new Entry\n4. Delete Old Entry\n";
        cout<<"5. Edit Entry\n6. Search Entry\n7. Exit\n";
        cout << "Enter your choice: ";
        cin >> ch;
```

```
        switch (ch) {
            case 1:
                if (flag == 0) {
                    cout << "\nEnter No. of Students to insert: ";
                    cin >> n;
                    for (i = 0; i < n; i++) {
                        f.addRecord();
                    }
                    flag = 1;
                } else {
                    cout << "\nSorry table is already built";
                }
                break;
            case 2:
                f.show();
                break;
            case 3:
                f.addRecord();
                break;
            case 4:
                cout << "\nEnter Roll No of Student whose Student is to be Deleted: ";
                cin >> n;
                i = f.delRecord(n);
                if (i == 1) {
                    cout << "\nStudent Deleted Successfully ";
                } else {
                    cout << "\nStudent not Found ";
                }
                break;
            case 5:
                cout << "\nEnter Roll No of Student whose Student is to be Edited: ";
                cin >> n;
                i = f.edit(n);
                if (i == 1) {
                    cout << "\nStudent Modified Successfully ";
                } else {
                    cout << "\nStudent not Found ";
                }
                break;
            case 6:
                cout << "\nEnter Roll No of Student to be Searched: ";
                cin >> n;
                f.search(n);
                break;
            case 7:
                cout << "\nExiting...";
                break;
            default:
                cout << "\nInvalid Choice";
        }
    } while (ch != 7);
    return 0;
}
```