

~\OneDrive\Desktop\focp-II\Assignment-II.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  class Student {
5      private:
6          int rno;
7          float cgpa;
8          string name;
9          string courses[10];
10         int courseCount;
11
12     public:
13
14     Student() {
15         name = "Unknown";
16         rno = 0;
17         cgpa = 0.0;
18         courseCount = 0;
19         cout << "Default constructor called.\n";
20     }
21
22
23     Student(string n, int r, float c) {
24         name = n;
25         rno = r;
26         setCGPA(c);
27         courseCount = 0;
28     }
29
30
31     Student(const Student& other) {
32         name = other.name;
33         rno = other.rno;
34         cgpa = other.cgpa;
35         courseCount = other.courseCount;
36         for (int i = 0; i < courseCount; i++) {
37             courses[i] = other.courses[i];
38         }
39     }
40
41     // Destructor
42     ~Student() {
43         cout << "Destructor called for: " << name << endl;
44     }
45
46
47     void getInfo() {
48         cout << "Enter name: ";
```

```
49     cin >> name;
50     cout << "Enter roll number: ";
51     cin >> rno;
52     cout << "Enter CGPA: ";
53     float tempCgpa;
54     cin >> tempCgpa;
55     setCGPA(tempCgpa);
56 }
57
58
59 void setCGPA(float c) {
60     if (c >= 0.0 && c <= 10.0)
61         cgpa = c;
62     else {
63         cout << "Invalid CGPA. Must be between 0.0 and 10.0. Setting CGPA to 0.\n";
64         cgpa = 0.0;
65     }
66 }
67
68
69 void inputCourses() {
70     int n;
71     cout << "How many courses (max 10)? ";
72     cin >> n;
73     if (n > 10) n = 10;
74     for (int i = 0; i < n; i++) {
75         cout << "Enter course " << i + 1 << ": ";
76         cin >> courses[i];
77     }
78     courseCount = n;
79 }
80
81
82 void addCourses() {
83     int choice;
84     cout << "Do you want to add more courses? (1 = Yes, 0 = No): ";
85     cin >> choice;
86
87     if (choice == 1) {
88         int extra;
89         cout << "How many courses to add? ";
90         cin >> extra;
91         if (courseCount + extra > 10) {
92             cout << "You can only have up to 10 courses total.\n";
93             extra = 10 - courseCount;
94         }
95
96         for (int i = 0; i < extra; i++) {
97             cout << "Enter additional course " << i + 1 << ": ";
98             cin >> courses[courseCount];
```

```
99         courseCount++;
100     }
101 }
102 }
103
104
105 void display() const {
106     cout << "\nStudent Name: " << name;
107     cout << "\nRoll Number: " << rno;
108     cout << "\nCGPA: " << cgpa;
109     cout << "\nCourses: ";
110     for (int i = 0; i < courseCount; i++) {
111         cout << courses[i] << " ";
112     }
113     cout << "\n";
114 }
115
116 int getRollNumber() const {
117     return rno;
118 }
119 };
120 class StudentManagementSystem {
121     private:
122         Student students[100];
123         int studentCount;
124
125     public:
126         StudentManagementSystem() {
127             studentCount = 0;
128         }
129
130         void addStudent() {
131             if (studentCount >= 100) {
132                 cout << "Maximum number of students reached.\n";
133                 return;
134             }
135
136             Student s;
137             s.getInfo();
138             s.inputCourses();
139             s.addCourses();
140             students[studentCount] = s;
141             studentCount++;
142
143             cout << "Student added successfully!\n";
144         }
145
146         void searchStudent(int rollNo) const {
147             for (int i = 0; i < studentCount; i++) {
148                 if (students[i].getRollNumber() == rollNo) {
```

```
149         cout << "Student found:\n";
150         students[i].display();
151         return;
152     }
153 }
154 cout << "Student with roll number " << rollNo << " not found.\n";
155 }
156
157 void displayAllStudents() const {
158     if (studentCount == 0) {
159         cout << "No students in the system.\n";
160         return;
161     }
162
163     for (int i = 0; i < studentCount; i++) {
164         students[i].display();
165     }
166 }
167 };
168 int main() {
169     StudentManagementSystem sms;
170
171     int choice;
172     do {
173         cout << "\n----- Student Management Menu ----- \n";
174         cout << "1. Add Student\n";
175         cout << "2. Search Student by Roll Number\n";
176         cout << "3. Display All Students\n";
177         cout << "0. Exit\n";
178         cout << "Enter your choice: ";
179         cin >> choice;
180
181         switch (choice) {
182             case 1:
183                 sms.addStudent();
184                 break;
185             case 2: {
186                 int rollNo;
187                 cout << "Enter roll number to search: ";
188                 cin >> rollNo;
189                 sms.searchStudent(rollNo);
190                 break;
191             }
192             case 3:
193                 sms.displayAllStudents();
194                 break;
195             case 0:
196                 cout << "Exiting...\n";
197                 break;
198             default:
```

```
199         cout << "Invalid choice.\n";
200     }
201 } while (choice != 0);
202
203 return 0;
204 }
205
206
207
```