

```
1  #include <iostream.h>
2  #include <conio.h>
3  #include <string.h>
4  #include <stdlib.h>
5
6  #define yAddStudent 1
7  #define yPrintStudentByID 2
8  #define yPrintAll 3
9  #define yExit 4
10
11 #define ArrowUp 72
12 #define ArrowDown 80
13 #define Enter 13
14 #define Escape 27
15 #define Home 71
16 #define End 79
17
18 //===== Functions Prototype =====//
19 void PrintMenu(int x);
20 void findColor(int y);
21 void redraw(int x, int y);
22 //===== class Student =====//
23 class Student
24 {
25     //=====By Default: Private=====//
26     int id;
27     char *name;
28
29 public:
30     //=====Setters=====//
31     void SetID(int reqID);
32     void SetName(char M[]);
33     //=====Getters=====//
34     int GetID();
35     char *GetName();
36     Student() {};
37     //=====Destructor=====//
38     ~Student() { delete (name); };
39     //=====Copy Constructor=====//
40     Student(Student &rst);
41 };
```

```
39      //==Copy Constructor==//
40      Student(Student &rst);
41  };
42  //===== Non-Member Functions =====//
43  Student FillStudent(void);
44  void PrintStudent(Student S);
45  Student FindStudentByID(int reqID);
46  void PrintAll(void);
47  Student ar[10];
48  int Size, s;
49  Student *ptr;
```

```
51 int main()
52 {
53     // I. Variables Declaration
54
55     int x = 1, y, App = 1, d, id;
56     char Ascii;
57
58
59     // II. Expressions & Function Calling
60     clrscr();
61     cout << "Enter The Required Number of Students  ";
62     cin >> Size;
63     ptr = (Student *)malloc(Size * sizeof(Student));
64     clrscr();
65     PrintMenu(x);
66     y = 1;
67     gotoxy(x, y);
68     findColor(y);
69     while (App != 0)
70     {
71
72         Ascii = getch(); // Normal Key
73         if (Ascii == 0)
74             Ascii = getch(); // Extended Key
75         switch (Ascii)
76         {
77             case ArrowUp:
78                 if (y == yAddStudent)
79                 {
80                     y = yExit;
81                     redraw(x, y);
82                 }
83             else
84             {
85                 y--;
86                 redraw(x, y);
87             }
88             break;
89             case ArrowDown:
90                 if (y == yExit)
91                 {
92                     y = yAddStudent;
93                     redraw(x, y);
94                 }
95             }
96         }
```

```

92         y = yAddStudent;
93         redraw(x, y);
94     }
95     else
96     {
97         y++;
98         redraw(x, y);
99     }
100    break;
101    case Enter:
102        switch (y)
103        {
104            case yAddStudent:
105                clrscr();
106                ptr[s] = FillStudent();
107                s++;
108                clrscr();
109                redraw(x, y);
110                break;
111            case yPrintStudentByID:
112                clrscr();
113                cout << "Enter The Required Student ID: ";
114                cin >> id;
115                PrintStudent(FindStudentByID(id));
116                getch();
117                clrscr();
118                redraw(x, y);
119                break;
120            case yPrintAll:
121                clrscr();
122                PrintAll();
123                getch();
124                clrscr();
125                redraw(x, y);
126                break;
127            case yExit:
128                App = 0;
129                break;
130        }
131        break;
132    case Escape:
133        App = 0;
134        break;
135    case Home:
136        y = yAddStudent;
137        redraw(x, y);
138        break;
139    case End:
140        y = yExit;
141        redraw(x, y);
142        break;
143 }

```

```
141         redraw(x, y);
142         break;
143     }
144 }
145 if (Ascii != Escape && !(Ascii == Enter && y == yExit))
146     getch();
147 clrscr();
148 return 0;
149 }
```

```
150 // III.Functions
151
152 void PrintMenu(int x)
153 {
154     textcolor(WHITE);
155     textbackground(BLACK);
156     gotoxy(x, yAddStudent);
157     cprintf("1.Add a New Student");
158     gotoxy(x, yPrintStudentByID);
159     cprintf("2.Student Info By ID");
160     gotoxy(x, yPrintAll);
161     cprintf("3.Print All Students");
162     gotoxy(x, yExit);
163     cprintf("4.Exit");
164 }
165
166 void findColor(int y)
167 {
168     textcolor(RED);
169     textbackground(WHITE);
170     switch (y)
171     {
172     case yAddStudent:
173         cprintf("1.Add a New Student");
174         break;
175     case yPrintStudentByID:
176         cprintf("2.Student Info By ID");
177         break;
178     case yPrintAll:
179         cprintf("3.Print All Students");
180         break;
181     case yExit:
182         cprintf("4.Exit");
183         break;
```



```

182         cprintf("4.Exit");
183         break;
184     }
185 }
186 Student FillStudent(void)
187 {
188     Student S;
189     int id;
190     char n[2];
191     cout << "Enter the Student ID";
192     cin >> id;
193     S.SetID(id);
194     cout << "Enter the Student Name";
195     cin >> n;
196     S.SetName(n);
197     return S;
198 }
199 Student FindStudentByID(int reqID)
200 {
201     for (int i = 0; i < s; i++)
202     {
203         if (ptr[i].GetID() == reqID)
204             return ptr[i];
205     }
206 }
207 void PrintStudent(Student S)
208 {
209     cout << "Student ID: " << S.GetID() << endl;
210     cout << "Student Name: " << S.GetName() << endl;
211 }
212 void PrintAll(void)
213 {
214     for (int i = 0; i < s; i++)
215     {
216         cout << "***** Student Number "
217             << ptr[i].GetID()
218             << "*****"
219             << endl;
220         cout << "Student ID :      "
221             << ptr[i].GetID()
222             << endl;

```

```

221         << ptr[i].GetID()
222         << endl;
223         cout << "Student Name :      "
224         << ptr[i].GetName()
225         << endl;
226     }
227 }
228 void redraw(int x, int y)
229 {
230     PrintMenu(x);
231     gotoxy(x, y);
232     findColor(y);
233 }
234 void Student::SetID(int reqID)
235 {
236     id = reqID;
237 }
238 void Student::SetName(char N[])
239 {
240     // delete (this->name);
241     this->name = new char[strlen(N) + 1];
242     strcpy(name, N);
243 }
244 int Student::GetID()
245 {
246     return id;
247 }
248 char *Student::GetName()
249 {
250     return name;
251 }
252 Student::Student(Student &rst)
253 {
254     id = rst.id;
255     // delete (this->name);
256     this->name = new char[strlen(rst.name) + 1];
257     strcpy(name, rst.name);
258 }

```


