```cpp
#include<iostream>
#include<string>
#include<ctime>
#include<cstdlib>

using namespace std;

//Class to create date type

class Date {
    int day;
    int month;
    int year;

public:
    //Constructor

    Date(int d = 1, int m = 1, int y = 1960) {
        day = d;
        month = m;
        year = y;
    }

    //Function to take date as input returns 1 if input is valid

    int getDate() {
        string d;
        cout << "Enter date in dd-mm-yyyy format" << endl;
        cin >> d;
        int dy, mn, yr;
        //Validating input
        if (d.length() != 10) {
            cout << "Invalid date format" << endl;
            return 0;
        }            //If length is correct then check
        else {
            if (d[2] != '-' || d[5] != '-') {
                cout << "Invalid date format" << endl;
                return 0;
            } else {
                dy = 10 * (d[0] - '0') + (d[1] - '0');
                mn = 10 * (d[3] - '0') + (d[4] - '0');
                yr = 1000 * (d[6] - '0') + 100 * (d[7] - '0')
                        + 10 * (d[8] - '0') + (d[9] - '0');
                //Validating hour
                Date dt(dy, mn, yr);
                if (!isValid(dt)) {
                    cout << "Invalid date entered" << endl;
                    return 0;
                }
            }
        }
        day = dy;
        month = mn;
        year = yr;
        return 1;

    }

    //Function to check if date is valid or not
    int isValid(Date d) {
        //Validating correct date
```

```
63          int days[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
64          if (isLeap(d.year))
65              days[1] = 29;
66          if (d.month < 1 || d.month > 12)
67              return 0;
68          else {
69              if (d.day < 0 || d.day > days[d.month - 1])
70                  return 0;
71              else {
72                  //Checking whether the date is after the current date then invalid
73                  time_t now = time(0);
74                  tm *ltm = localtime(&now);
75                  int c_year = 1900 + ltm->tm_year;
76                  int c_month = 1 + ltm->tm_mon;
77                  int c_day = ltm->tm_mday;
78                  //If input year is after current date
79                  if (d.year > c_year)
80                      return 0;
81                  //If input year is ok
82                  else if (d.year == c_year) {
83                      //If input year is equal to this year then check month
84                      if (d.month > c_month)
85                          return 0;
86                      else if (d.month == c_month) {
87                          if (d.day > c_day)
88                              return 0;
89                      }
90
91                  }
92                  return 1;
93              }
94          }
95
96      }
97      //Function to check if a year is a leap year or not
98
99      int isLeap(int y) {
100         if ((y % 100 == 0 & y % 400 == 0) || (y % 100 != 0 && y % 4 == 0))
101             return 1;
102         return 0;
103     }
104 };
105
106 //Class to store student details
107
108 class Student {
109 public:
110     static int count; //Store total number of students
111
112 private:
113     string name;
114     string course;
115     Date adDate;
116     int marks[5];
117     int rollNo;
118
119 public:
120     //Function to take admission
121
122     void admission() {
123         cin.ignore();
124         do {
```

```
125
126             cout << "Enter name of student: " << endl;
127             getline(cin, name);
128             if (!isName(name))
129                 cout << "Invalid name" << endl;
130         } while (!isName(name));
131         cout << endl << "Enter course name: ";
132         cin >> course;
133         cout << endl << "Enter admission date: ";
134         while (adDate.getDate() == 0)
135             ;
136         count++;
137         rollNo = 100000 + count;
138         cout << "Admission successfull" << endl;
139         cout << "Roll number of new student is: " << rollNo << endl;
140     }
141
142     //Function to check whether name is valid
143
144     int isName(string n) {
145         int flag = 1, i;
146         for (i = 0; i < n.length(); i++)
147             if (n[i] != ' ')
148                 if (n[i] < 65 || (n[i] > 90 && n[i] < 97) || n[i] > 122) {
149                     flag = 0;
150                     break;
151                 }
152         return flag;
153     }
154
155     //Function to receive marks
156
157     void receiveMarks() {
158         cout << "Enter marks in 5 subjects" << endl;
159         int i;
160         for (i = 0; i < 5;) {
161             cin >> marks[i];
162             if (marks[i] < 0 || marks[i] > 100) {
163                 cout << "invalid marks" << endl;
164
165             } else
166                 i++;
167         }
168
169     }
170
171     //Function to display marksheet
172
173     void displayMarksheet() {
174         int i;
175         cout
176                 << "================================================================="
177                 << endl;
178         cout << "|\tName :" << name << "\t\t";
179         cout << "| Roll number: " << rollNo << "\t\t\t|" << endl;
180         cout
181                 << "-----------------------------------------------------------------"
182                 << endl;
183         cout << "|\tSUBJECT\t\t\t|\t\tMARKS\t\t|" << endl;
184         cout
185                 << "-----------------------------------------------------------------"
186                 << endl;
```

```
187         for (i = 0; i < 5; i++) {
188             cout << "|\tSubject " << i << "\t\t|\t\t";
189             cout << marks[i] << "\t\t|" << endl;
190         }
191         cout
192                 << "----------------------------------------------------------------"
193                 << endl;
194         double per = (marks[0] + marks[1] + marks[2] + marks[3] + marks[4])
195                 / 5.0;
196         cout << "|\tPercentage: " << per << "%\t\t\t\t\t\t|" << endl;
197         cout
198                 << "================================================================"
199                 << endl;
200     }
201 
202     //Function to get roll number of student
203 
204     int getRoll() {
205         return rollNo;
206     }
207 
208 };
209 int Student::count = 0;
210 
211 int main() {
212     Student st[100];
213     int ch, roll, flag = 1, i;
214     do {
215         cout << "1. New Admission" << endl << "2. Marks-entry" << endl;
216         cout << "3. Marksheet display" << endl
217                 << "4. Display total no of students" << endl;
218         cout << "5. Exit" << endl << "Enter choice" << endl;
219         cin >> ch;
220         switch (ch) {
221         case 1:
222             st[Student::count].admission();
223             break;
224 
225         case 2:
226             flag = 0;
227             do {
228                 cout << "Enter roll number of student ";
229                 cin >> roll;
230                 //Check if a roll exists in the array or not
231                 for (i = 0; i < Student::count; i++)
232                     if (st[i].getRoll() == roll) {
233                         flag = 1;
234 
235                         break;
236                     }
237                 if (flag == 0)
238                     cout << "Invalid roll number... Re-enter" << endl;
239             } while (flag == 0);
240 
241             st[i].receiveMarks();
242             break;
243 
244         case 3:
245             flag = 0;
246             do {
247                 cout << "Enter roll number of student ";
248                 cin >> roll;
```

```
249
250                 //Check if a roll exists in the array or not
251                 for (i = 0; i < Student::count; i++)
252                     if (st[i].getRoll() == roll) {
253                         flag = 1;
254                         break;
255                     }
256                 if (flag == 0)
257                     cout << "Invalid roll number... Re-enter" << endl;
258             } while (flag == 0);
259             st[i].displayMarksheet();
260             break;
261
262         case 4:
263             cout << "Total number of students: " << Student::count << endl;
264             break;
265
266         case 5:
267             cout << "Quitting" << endl;
268             exit(0);
269
270         default:
271             cout << "Invalid choice" << endl;
272         }
273     } while (ch != 5);
274     return 0;
275 }
276
```