```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
        int days[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
        if (isLeap(d.year))
            days[1] = 29;
        if (d.month < 1 || d.month > 12)
            return 0;
        else {
            if (d.day < 0 || d.day > days[d.month - 1])
                return 0;
            else {
                //Checking whether the date is after the current date then invalid
                time_t now = time(0);
                tm *ltm = localtime(&now);
                int c_year = 1900 + ltm->tm_year;
                int c_month = 1 + ltm->tm_mon;
                int c_day = ltm->tm_mday;
                //If input year is after current date
                if (d.year > c_year)
                    return 0;
                //If input year is ok
                else if (d.year == c_year) {
                    //If input year is equal to this year then check month
                    if (d.month > c_month)
                        return 0;
                    else if (d.month == c_month) {
                        if (d.day > c_day)
                            return 0;
                    }

                }
                return 1;
            }
        }

    }
    //Function to check if a year is a leap year or not

    int isLeap(int y) {
        if ((y % 100 == 0 & y % 400 == 0) || (y % 100 != 0 && y % 4 == 0))
            return 1;
        return 0;
    }
};

//Class to store student details

class Student {
public:
    static int count; //Store total number of students

private:
    string name;
    string course;
    Date adDate;
    int marks[5];
    int rollNo;

public:
    //Function to take admission

    void admission() {
        cin.ignore();
        do {
```

```

        cout << "Enter name of student: " << endl;
        getline(cin, name);
        if (!isName(name))
            cout << "Invalid name" << endl;
    } while (!isName(name));
    cout << endl << "Enter course name: ";
    cin >> course;
    cout << endl << "Enter admission date: ";
    while (adDate.getDate() == 0)
        ;
    count++;
    rollNo = 100000 + count;
    cout << "Admission successfull" << endl;
    cout << "Roll number of new student is: " << rollNo << endl;
}

//Function to check whether name is valid

int isName(string n) {
    int flag = 1, i;
    for (i = 0; i < n.length(); i++)
        if (n[i] != ' ')
            if (n[i] < 65 || (n[i] > 90 && n[i] < 97) || n[i] > 122) {
                flag = 0;
                break;
            }
    return flag;
}

//Function to receive marks

void receiveMarks() {
    cout << "Enter marks in 5 subjects" << endl;
    int i;
    for (i = 0; i < 5;) {
        cin >> marks[i];
        if (marks[i] < 0 || marks[i] > 100) {
            cout << "invalid marks" << endl;

        } else
            i++;
    }

}

//Function to display marksheet

void displayMarksheet() {
    int i;
    cout
            << "================================================================="
            << endl;
    cout << "|\tName :" << name << "\t\t";
    cout << "| Roll number: " << rollNo << "\t\t\t|" << endl;
    cout
            << "-----------------------------------------------------------------"
            << endl;
    cout << "|\tSUBJECT\t\t\t|\t\tMARKS\t\t|" << endl;
    cout
            << "-----------------------------------------------------------------"
            << endl;
```

```cpp
        for (i = 0; i < 5; i++) {
            cout << "|\tSubject " << i << "\t\t|\t\t";
            cout << marks[i] << "\t\t|" << endl;
        }
        cout
                << "-----------------------------------------------------------------"
                << endl;
        double per = (marks[0] + marks[1] + marks[2] + marks[3] + marks[4])
                / 5.0;
        cout << "|\tPercentage: " << per << "%\t\t\t\t\t\t|" << endl;
        cout
                << "================================================================="
                << endl;
    }

    //Function to get roll number of student

    int getRoll() {
        return rollNo;
    }

};
int Student::count = 0;

int main() {
    Student st[100];
    int ch, roll, flag = 1, i;
    do {
        cout << "1. New Admission" << endl << "2. Marks-entry" << endl;
        cout << "3. Marksheet display" << endl
                << "4. Display total no of students" << endl;
        cout << "5. Exit" << endl << "Enter choice" << endl;
        cin >> ch;
        switch (ch) {
        case 1:
            st[Student::count].admission();
            break;

        case 2:
            flag = 0;
            do {
                cout << "Enter roll number of student ";
                cin >> roll;
                //Check if a roll exists in the array or not
                for (i = 0; i < Student::count; i++)
                    if (st[i].getRoll() == roll) {
                        flag = 1;

                        break;
                    }
                if (flag == 0)
                    cout << "Invalid roll number... Re-enter" << endl;
            } while (flag == 0);

            st[i].receiveMarks();
            break;

        case 3:
            flag = 0;
            do {
                cout << "Enter roll number of student ";
                cin >> roll;

                //Check if a roll exists in the array or not
                for (i = 0; i < Student::count; i++)
                    if (st[i].getRoll() == roll) {
                        flag = 1;
                        break;
                    }
                if (flag == 0)
                    cout << "Invalid roll number... Re-enter" << endl;
            } while (flag == 0);
            st[i].displayMarksheet();
            break;

        case 4:
            cout << "Total number of students: " << Student::count << endl;
            break;

        case 5:
            cout << "Quitting" << endl;
            exit(0);

        default:
            cout << "Invalid choice" << endl;
        }
    } while (ch != 5);
    return 0;
}

```