Practical No 3   **Date:** 31/08/2023

Title: Write a C++ program using a class student to calculate the total, percentage of marks and grade obtained by him/her.

Description:

A class in C++ is the building block, that leads to Object-Oriented programming. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.

Classes are like templates that define common properties or attributes. Objects are instances of a class. Once a class has been defined, we can create any number of objects belonging to that class. Each object is associated with the data or type class with which they are created.

Declaration of class:

        class <class_name>
         {
            data members;
            member functions;
         };

Example:
class item
{private:
   int number;   //variables declaration
   float cost;   // private by default
public: void getdata(int a, int b);   // functions declaration
        void putdata();   // using prototype
}; // ends with semi-colon

Syntax for creating object:
        classname   objectname;
Example: fruit mango;

Program Code:

```cpp
9  #include <iostream>
10 #include <iomanip>
11 using namespace std;
12
13 class Student
14 {
15 private:
16     string name;
17     int sub1, sub2, sub3;
18     double total, percentage;
19     char grade;
20
21 public:
22     Student(const string& n, int s1, int s2, int s3) : name(n), sub1(s1), sub2(s2), sub3(s3)
23     {
24         gettotal();
25         getpercent();
26         getgrade();
27     }
28
29     void gettotal(){
30         total=sub1+sub2+sub3;
31     }
32     void getpercent(){
33         percentage=total/3.0;
34     }
```

```cpp
void getgrade(){
    if(percentage>=90)
    {
        grade='A';
    }
    else if(percentage>=80)
    {
        grade='B';
    }
    else if(percentage>=70)
    {
        grade='C';
    }
    else if(percentage>=60)
    {
        grade='D';
    }
    else if(percentage>=50)
    {
        grade='E';
    }
    else
    {
        grade='F';
    }
}
```

```cpp
void result()
{
    cout << "----------------------------------------" << endl;
    cout << "                Report Card             " << endl;
    cout << "----------------------------------------" << endl;
    cout << left << setw(15) << "Name:" << setw(25) << name << endl;
    cout << left << setw(15) << "Subject 1:" << setw(25) << sub1 << endl;
    cout << left << setw(15) << "Subject 2:" << setw(25) << sub2 << endl;
    cout << left << setw(15) << "Subject 3:" << setw(25) << sub3 << endl;
    cout << left << setw(15) << "Total:" << setw(25) << total << endl;
    cout << left << setw(15) << "Percentage:" << setw(25) << fixed << setprecision(2) << percentage << "%" <<
    cout << left << setw(15) << "Grade:" << setw(25) << grade << endl;
    cout << "----------------------------------------" << endl;
}
};
```

```cpp
int main()
{
    string name;
    int marks1, marks2, marks3;

    cout << "Enter student's name: ";
    getline(cin, name);

    cout << "Enter marks for Subject 1: ";
    cin >> marks1;
    cout << "Enter marks for Subject 2: ";
    cin >> marks2;
    cout << "Enter marks for Subject 3: ";
    cin >> marks3;

    Student student(name, marks1, marks2, marks3);
    student.result();

    return 0;
}
```
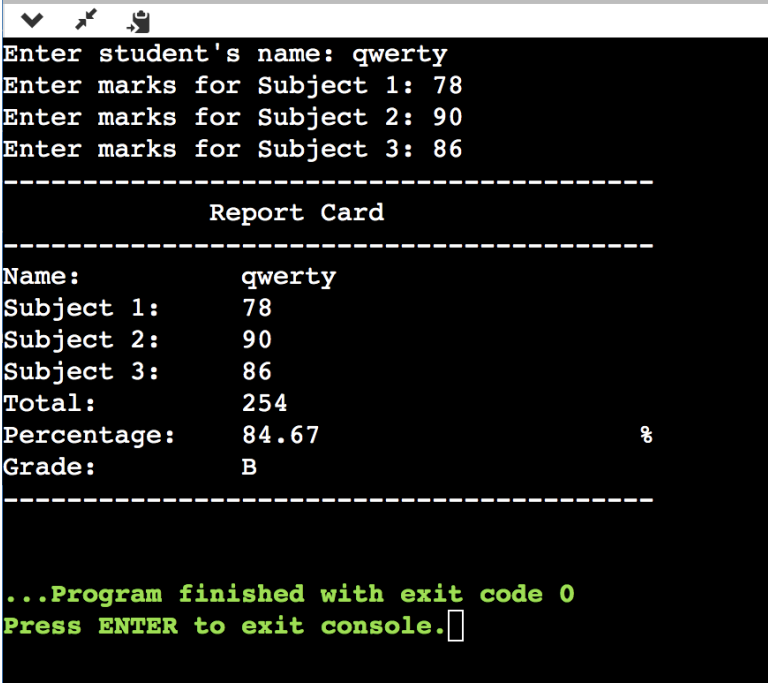

Input and Output

```
 ∨  ⤢  ⬒
Enter student's name: qwerty
Enter marks for Subject 1: 78
Enter marks for Subject 2: 90
Enter marks for Subject 3: 86
-----------------------------------------
            Report Card
-----------------------------------------
Name:          qwerty
Subject 1:     78
Subject 2:     90
Subject 3:     86
Total:         254
Percentage:    84.67                    %
Grade:         B
-----------------------------------------


...Program finished with exit code 0
Press ENTER to exit console.⌷
```

Conclusion: Thus we have implemented the concept of classes in C++

Practice programs: Write a program to print the area of a triangle, rectangle and circle by creating a class named 'Area' having two functions. First function named as 'setDim' takes the length and breadth of the rectangle as parameters and the second function named as 'getArea' returns the area of the rectangle. Length and breadth of the rectangle are entered through keyboard.

PROGRAM CODE:

Input and output