Classes and Objects-Hard(2)

Aim:

Write a program to create a Student class with the following attributes: int rollno,

int mark1, int mark2, int mark3.

Create an array of objects for the above class. In the main class, calculate and print the following.

- Total marks obtained by each student.
- The highest mark in each subject with the roll number of the student who scored it.
- The student who obtained the highest total mark

Source Code:

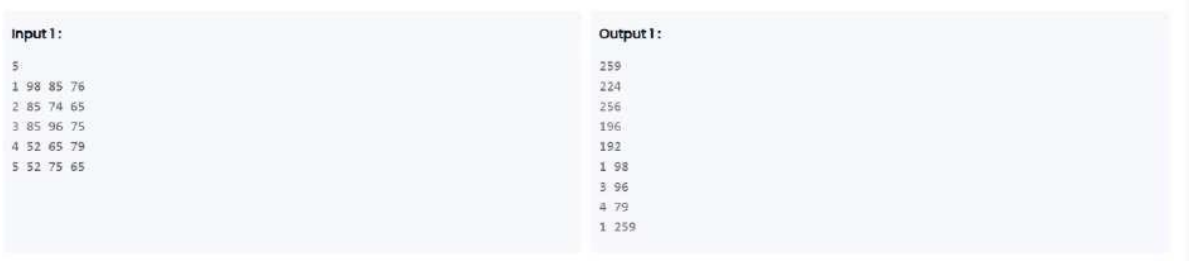
```
#include<iostream> using namespace std; class Student
{
int rollno, mark1, mark2, mark3; public:
void setRno(int x)
{
rollno=x;
}
void setMark1(int x)
{
mark1=x;
}
void setMark2(int x)
{
mark2=x;
}
void setMark3(int x)
{
mark3=x;
}
int totalMarks()
{
return mark1+mark2+mark3;
}
```

```
static void findMaxMarks(Student s[], int n){
int m1, m2, m3, r1, r2, r3, i; m1=m2=m3=0;
for(i=0;i<n;i++)
{
if(m1<s[i].mark1)
{
m1=s[i].mark1; r1=s[i].rollno;
}
if(m2<s[i].mark2)
{
m2=s[i].mark2; r2=s[i].rollno;
}
if(m3<s[i].mark3)
{
m3=s[i].mark3; r3=s[i].rollno;
}
}
cout<<r1<<" "<<m1<<endl; cout<<r2<<" "<<m2<<endl; cout<<r3<<" "<<m3<<endl;
}
static void findMaxTotalMarks(Student s[], int n)
{
int i, max, sum, r; max=0; for(i=0;i<n;i++)
{
sum=s[i].mark1+s[i].mark2+s[i].mark3; if(sum>max)
{
r=s[i].rollno; max=sum;
}
}
cout<<r<<" "<<max;
}
};
int main() { int n;
cin >> n;
```

```
Student s[n];
for (int i = 0; i < n; i++) { int t;
cin >> t; s[i].setRno(t); cin >> t; s[i].setMark1(t); cin >> t; s[i].setMark2(t); cin >> t; s[i].setMark3(t);
}
for (int i = 0; i < n; i++) {
cout << s[i].totalMarks() << endl;
}
Student::findMaxMarks(s, n); Student::findMaxTotalMarks(s, n);

return 0;
}
```

Input / Output:



Input 1 :

```
5
1 98 85 76
2 85 74 65
3 85 96 75
4 52 65 79
5 52 75 65
```

Output 1 :

```
259
224
256
196
192
1 98
3 96
4 79
1 259
```

Constructors and Destructors-Easy(1)

Aim:

Manu is a software developer working on a project that involves handling and displaying data. He is designing a class called **"Demo"** to represent data objects. The class has two private integer variables: num1 and num2, a constructor, a display function, and a destructor.

Source Code:

```
#include<iostream> using namespace std; class Demo
{
int num1, num2; public:
Demo(int n1, int n2):num1(n1), num2(n2)
{
cout<<"Inside Constructor"<<endl; display();
}
~Demo()
{
cout<<"Inside Destructor"<<endl;
}
void display()
{
cout<<num1<<endl; cout<<num2<<endl;
}
};
int main()
{
int x, y; cin>>x>>y; Demo d(x, y);
}
```

Input / Output:

Input 1:

```
4 5
```

Output 1:

```
Inside Constructor
4
5
Inside Destructor
```

Constructors and Destructors-Easy(2)

Aim:

You are tasked with creating a **GradeBook** class to automate grade storage for students. The GradeBook class should have the following features:

1. Private members: courseName and instructorName with corresponding get and set methods.
2. A constructor that allows creating an object without initial values and a setData method to set courseName and instructorName.
3. A displayMessage method to display the course and instructor details.
4. Utilize string data types for data handling.

The program should take input for the course name and instructor name and display them as is.

Source Code:

```
#include<iostream> #include<string> using namespace std; class GradeBook

{

string courseName, instructorName; public:

void setData(string c, string i)

{

courseName=c; instructorName=i; display();

}

void display()

{

cout<<"Welcome to the grade book for "<<courseName<<"!"<<endl; cout<<"This course is presented by: "<<instructorName<<endl;

}

};

int main()

{

string c, i; GradeBook g; getline(cin, c); getline(cin, i); g.setData(c, i);

}
```

Input / Output:

Input 1:

```
CS50
David Malan
```

Output 1:

```
Welcome to the grade book for CS50!
This course is presented by: David Malan
```

Input 2:

```
Computer Science
Addyson David Vicente Saunders
```

Output 2:

```
Welcome to the grade book for Computer Science!
This course is presented by: Addyson David Vicente Saunders
```
