The BRITISH UNIVERSITY IN EGYPT

**22CSCI01P**

# Introduction to Computing

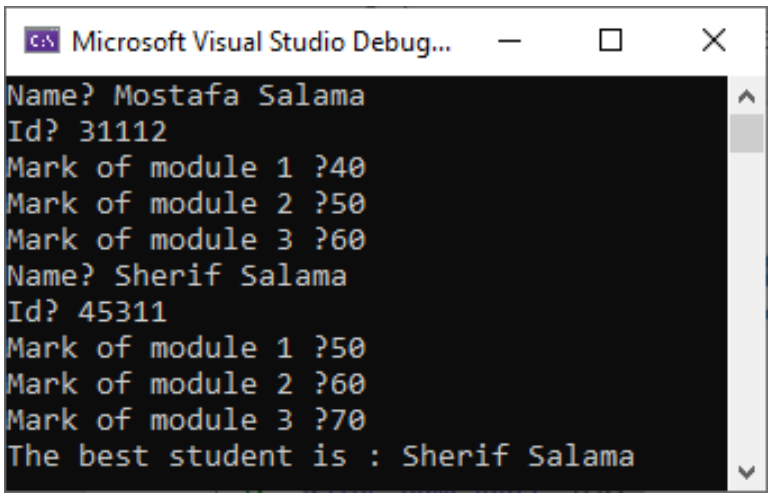
## Lab (8) - Structs

**Exercise 1:** Write, compile, and test the following programs:

**Problem (1)** ***Define two students, get their data from the teacher and compare between them.***

1. Write a C++ program, that defines a structure of name ***student***; this structure contains 5 fields,
   - The student's name as a string; ***name***.
   - The student's id as an integer; ***id*** of default value ***defaultID***.
   - The student's status as a boolean; ***status(success as 1 and fail as 0); default is 0.***
   - The student's final Grade as a character; ***finalGrade (A, B, C, D, F); default is F.***
   - The student's final Mark as a double; ***finalMark; default is 0.0.***
   - The student's ModulesMarks as an array of doubles; ***marks[numOfModules]*** .
2. Such that ***defaultID*** is constant integer of value 00000, and ***numOfModules*** is constant integer of value 3.
3. In the main function; define three objects ***student1*** , ***student2*** and ***bestStudent***.
4. Pass ***student1*** to a function ***setStudent()***, use pass by reference. Then pass ***student2*** to a function ***setStudent()***, use pass by reference. This function has one parameter ***s*** of type ***student***.
5. The function ***setStudent()*** gets student's name, id and the marks of the module, then calculate the rest {which are ***finalMark*** as average of module marks, ***status***, and ***finalGrade***} accordingly.
6. Pass ***student1*** and ***student2*** to function ***compare()*** by value. This function has two parameter of type ***student*** and it returns an object of type ***student***.
7. The function ***compare()*** returns the student whose final mark is greater than the other.

*The output of the program is as follows:*



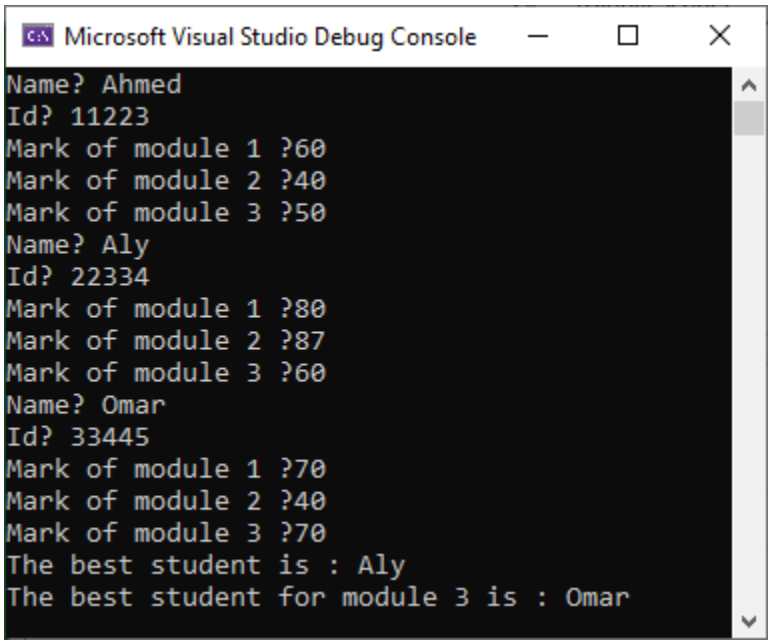
**Solution (1)** *Define two students, get their data from the teacher and compare between them:*

```
#include <iostream>
#include <string>
using namespace std;
const int defaultID = 00000;
const int numOfModules = 3;
struct student {
	string name;
	int id = defaultID;
	bool status = false;
	char finalGrade = 'F';
	double finalMark = 0;
	double marks[numOfModules] = { 0, 0, 0 };
};
void setStudent(student& s) {
	cout << "Name? "; cin.ignore();  getline(cin, s.name);
	cout << "Id? "; cin >> s.id;
	for (int i = 0; i < numOfModules; i++) {
		cout << "Mark of module " << (i + 1) << " ?";
		cin >> s.marks[i];
		s.finalMark += s.marks[i];
	}
	s.finalMark /= numOfModules;
	if (s.finalMark >= 40) {
		s.status = true;
		if (s.finalMark < 50)  s.finalGrade = 'D';
		else if (s.finalMark < 60)  s.finalGrade = 'C';
		else if (s.finalMark < 70)  s.finalGrade = 'B';
		else if (s.finalMark < 100)  s.finalGrade = 'A';
	}
}
student compare(student stud1, student stud2) {
	if (stud1.finalMark > stud2.finalMark) return stud1;
	else return stud2;
}
int main() {
	student student1, student2, bestStudent;
	setStudent(student1);
	setStudent(student2);
	bestStudent = compare(student1, student2);
	cout << "The best student is : " << bestStudent.name;
	return 0;
}
```

**Problem (2)** ***Update the previous program to get the data of 3 students, and do further analysis:***

1. Update the previous program as follows:
    a. In the main function; define three objects ***student1***, ***student2***, ***student3*** and ***bestStudent***.
    b. In the main; pass ***student3*** to a function ***setStudent()*** in addition to ***student1***, ***student2***.
    c. Update the function ***setStudent()***, such the main function passes the ***student*** object by value, not by reference, and returns an object of type student.
    d. Edit the ***compare()***; this function has three objects of type ***student*** instead of two students, it returns the best student among of three students.
    e. In the main; pass ***student1*** to a new function ***compareModule()***; this function has two parameters, the first parameter is an object of type ***student***, the second parameter is an integer i. this function returns and object of type student.
    f. The new function ***compareModule(student stud1, student stud2, student stud3, int i)***; it compare the s.marks[i] of the three students. Then it returns the best student among of three students for a specific module of index ***i***, where ***i*** is either 0, 1 or 2.

*The output of the program is as follows:*



**Solution (2)** *Update the previous program to get the data of 3 students, and do further analysis:*

```
#include <iostream>
#include <string>
using namespace std;
const int defaultID = 00000;
const int numOfModules = 3;
struct student {
	string name; int id = defaultID; bool status = false;char finalGrade = 'F';
	double finalMark = 0; double marks[numOfModules] = { 0, 0, 0 };
};
student setStudent(student s) {
	cout << "Name? "; cin.ignore();  getline(cin, s.name);
	cout << "Id? "; cin >> s.id;
	for (int i = 0; i < numOfModules; i++) {
		cout << "Mark of module " << (i + 1) << " ?";
		cin >> s.marks[i];
		s.finalMark += s.marks[i];
	}
	s.finalMark /= numOfModules;
	if (s.finalMark >= 40) {
		s.status = true;
		if (s.finalMark < 50)  s.finalGrade = 'D';
		else if (s.finalMark < 60)  s.finalGrade = 'C';
		else if (s.finalMark < 70)  s.finalGrade = 'B';
		else if (s.finalMark < 100)  s.finalGrade = 'A';
	}
	return s;
}









int main() {









}
```

**Problem (3)** ***Define an array 6 students, set their data, then sort these students according to their grades:***

1. Write a C++ program, that defines a structure of name ***student*** this structure contains student's name as a string; ***studentName***, the student's id as an ***studentId***, , the student's grade as an ***studentGrade***.
2. Define a constant integer ***numOfStudents*** of value 6.
3. In the main function; define an array ***students*** of size ***numOfStudents***. Initialize the values of the array by arbitrary values.
4. Pass by reference the array ***students*** to a function ***sortStudents()***. This function has one parameter which is an array of the structure ***student***. of name ***unSortedstudents***.
5. The function sort the ***student***'s objects in the array ***students*** according to their ***studentGrade*** values.
6. Print out the values in the array after sorting.

**Solution (3)** ***Define an array 6 students, set their data, then sort these students according to their grades:***

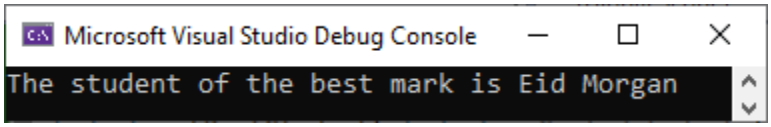
```
#include <iostream>
#include <fstream>
using namespace std;
const int numOfStudents = 6;
struct student {
       string studentName;
       int studentId;
       float studentGrade;
};
void sortStudents(student unSortedstudents[]) {
    for (int i = 0; i < numOfStudents - 1; i++) {
        for (int j = i + 1; j < numOfStudents; j++) {
          if (unSortedstudents[i].studentGrade < unSortedstudents[j].studentGrade) {
                student temp = unSortedstudents[i];
                unSortedstudents[i] = unSortedstudents[j];
                unSortedstudents[j] = temp;
          }
        }
    }
}
void main() {
    student students[] = { {"Ahmed Mahfoz", 211231, 45},
                           {"Samir AbdelLatif", 223412, 67},
                           {"Marwa Solimn", 267644, 55},
                           {"Eid Morgan", 255543, 78},
                           {"Sara Amin", 265633, 62},
                           {"Mostafa Negm", 265633, 62} };
    sortStudents(students);
    cout << "Students ordered from best to lowest grade:\n";
    for (int i = 0; i < numOfStudents; i++)
        cout << students[i].studentName << endl;
}
```

```
Microsoft Visual Studio Debug Console
Students ordered from best to lowest grade:
Eid Morgan
Samir AbdelLatif
Sara Amin
Mostafa Negm
Marwa Solimn
```

**Problem (4)** ***Replace sorting function by another function that finds the best student:***

1. Update the previous program as follows:
    a. In the main function, pass the array ***students*** to a function ***bestStudents()***. This function has one parameter which is an array of the structure ***student***. and it returns an object of type ***student***.
    b. The function ***getBestStudent()*** replaces the function ***sortStudents()***, this function ***getBestStudent ()*** searches in an unsorted array for the student whose mark is the greatest mark, then returns the object of this student back to the main function.
    c. In the main function, print out the name of the name of the returned student object from the function ***getBestStudent()***.

*The output of the program is as follows:*




**Solution (4)** ***Update the previous program to get the student of the best mark and the student of the best mark:***

```
#include <iostream>
#include <fstream>
using namespace std;
const int numOfStudents = 6;
struct student {
    string studentName;
    int studentId;
    float studentGrade;
};




void main() {
    student students[] = { {"Ahmed Mahfoz", 211231, 45},
                           {"Samir AbdelLatif", 223412, 67},
                           {"Marwa Solimn", 267644, 55},
                           {"Eid Morgan", 255543, 78},
                           {"Sara Amin", 265633, 62},
                           {"Mostafa Negm", 265633, 62} };









}
```