

```

//      Course: CS2400-60 Computer Science 2
//      Name: Abdalkarim, Marina
//      Assignment: Programming Assignment P4.1
//      Date assigned: 10/13/18
//      Date due: 10/23/17
//      Date handed in: 10/23/17
//      Remark: The program tests all functions.
#include <iostream>
#include <string>
using namespace std;
struct stuRec      // creates a new record "data type named "stuRec"
{
    string name;    // 1st data member of stuRec type
    int credits;    // 2nd data member of stuRec type
    double gpa;     // 3rd data member of stuRec type
};
void fillRec(stuRec &sobj);
// Precondition: a stuRec record (or object) sobj is passed to the function by reference (must!)
// user is prompted to enter name, credits completed, and GPA for the student
// Postcondition: stuRec record referenced by sobj is filled or loaded with three values entered by
// the user
void display(stuRec &sobj);
// Postcondition: contents of sobj is displayed (the format can be found in the sample output
// below)
void sortByName(stuRec s[], int n);
// Precondition: an array of n student records is passed to the function; array is fully loaded with
// data
// Postcondition: n records have been sorted by name alphabetically
void sortByGPA(stuRec s[], int n);
// Postcondition: n records have been sorted by GPA in descending order
void swap(stuRec &s1, stuRec &s2);
// Precondition: two student records are passed to the function by reference (must!)
int main()
{
    const int SIZE = 3;
    stuRec s[SIZE];
    for (int i = 0; i < SIZE; i++)
        fillRec(s[i]);
    cout << endl << "The three student records entered by the user are: " << endl;

```

```

        for (int i = 0; i < SIZE; i++)
            display(s[i]);
        cout << "...Sorting by Name..." << endl;
        sortByName(s, SIZE);
        for (int i = 0; i < SIZE; i++)
            display(s[i]);
        cout << endl;
        cout << "...Sorting by GPA..." << endl;
        sortByGPA(s, SIZE);
        for (int i = 0; i < SIZE; i++)
            display(s[i]);
        cout << endl;
    return 0;
}

void fillRec(stuRec &sobj)
{
    cout << "Enter name, credit, gpa: ";
    cin >> sobj.name >> sobj.credits >> sobj.gpa;
}

void display(stuRec &sobj)
{
    cout << "Name: " << sobj.name << endl;
    cout << "Credit: " << sobj.credits << endl;
    cout << "GPA: " << sobj.gpa << endl << endl;
}

void sortByName(stuRec s[], int n)
{
    for (int pass = 1; pass < n; pass++)
    {
        for (int c = 0; c < n - pass; c++)
        {
            if (s[c].name > s[c + 1].name)
            {
                swap(s[c].gpa, s[c + 1].gpa);
                swap(s[c].name, s[c + 1].name);
                swap(s[c].credits, s[c + 1].credits);
            }
        }
    }
}

```

```

}
void sortByGPA(stuRec s[], int n)
{
    for (int pass = 1; pass < n; pass++)
    {
        for (int c = 0; c < n - pass; c++)
        {
            if (s[c].gpa > s[c + 1].gpa)
            {
                swap(s[c].gpa, s[c + 1].gpa);
                swap(s[c].name, s[c + 1].name);
                swap(s[c].credits, s[c + 1].credits);
            }
        }
    }
}

void swap(stuRec &s1, stuRec &s2)
{
    stuRec temp = s1;
    s1 = s2;
    s2 = temp;
}

```

```
cs.wpunj.edu - PuTTY
-bash-3.2$ date
Sun Oct 21 14:21:02 EDT 2018
-bash-3.2$ pwd
/students/abdalkam
-bash-3.2$ ls
F2018          assign2.cpp    f2018          local.login    try.cpp
a.out          assign3.cpp    first.cpp      local.profile
assign.cpp     assign4.cpp    local.cshrc    pico.save
-bash-3.2$ g++ assign4.cpp
-bash-3.2$ ls
F2018          assign2.cpp    f2018          local.login    try.cpp
a.out          assign3.cpp    first.cpp      local.profile
assign.cpp     assign4.cpp    local.cshrc    pico.save
-bash-3.2$ a.out
Enter name, credit, gpa: Juri 98 1.2
Enter name, credit, gpa: Vegita 13 6.5
Enter name, credit, gpa: Bayonetta 76 9.8

The three student records entered by the user are:
Name: Juri
Credit: 98
GPA: 1.2

Name: Vegita
Credit: 13
GPA: 6.5

Name: Bayonetta
Credit: 76
GPA: 9.8

...Sorting by Name...
Name: Bayonetta
Credit: 76
GPA: 9.8

Name: Juri
Credit: 98
GPA: 1.2

Name: Vegita
Credit: 13
GPA: 6.5
```

```
...Sorting by GPA...  
Name: Juri  
Credit: 98  
GPA: 1.2
```

```
Name: Vegita  
Credit: 13  
GPA: 6.5
```

```
Name: Bayonetta  
Credit: 76  
GPA: 9.8
```

```
-bash-3.2$
```

```
//      Course: CS2400-60 Computer Science 2
```

```
//      Name: Abdalkarim, Marina
```

```
//      Assignment: Programming Assignment P4.2
```

```
//      Date assigned: 10/13/18
```

```
//      Date due: 10/23/17
```

```
//      Date handed in: 10/23/17
```

```
//      Remark: The program tests all functions.
```

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class stuRec          // creates a new data type named "stuRec" using class
```

```
{
```

```
public:
```

```
    string getName();
```

```
    // An "accessor" function; gaining access to the value of the "name" data member of the
```

```
    // calling stuRec object
```

```
    // Postcondition: returns the name of the calling object
```

```
    double getGPA();
```

```
    // Another accessor function; gaining access to the value of the "gpa" data member of the
```

```
    // calling stuRec object
```

```
    // Postcondition: returns the name of the calling object
```

```
    void fillRec();
```

```
    // calling object is a stuRec object that calls the public member function
```

```
    // Postcondition: the calling object (a stuRec record) is filled or loaded with three values
```

```
    // entered by the user
```

```
    void displayRec();
```

```
    // Postcondition: contents of the calling object is displayed (the format can be found in
```

```
    // the sample output below)
```

```

private:
    string name;    // 1st data member of stuRec type
    int credits;    // 2nd data member of stuRec type
    double gpa;     // 3rd data member of stuRec type
};

void sortByName(stuRec s[], int n);
// Precondition: an array of n student records is passed to the function; array is fully loaded with
// data
// Postcondition: n records have been sorted by name alphabetically
void sortByGPA(stuRec s[], int n);
// Postcondition: n records have been sorted by GPA in descending order
void swap(stuRec &s1, stuRec &s2);
// Precondition: two student records are passed to the function by reference (must!)
int main()
{
    const int SIZE = 3;
    stuRec s[SIZE];
    for (int i = 0; i < SIZE; i++)
        s[i].fillRec();
    for (int i = 0; i < SIZE; i++)
        s[i].displayRec();
    cout << endl;
    cout << "...Sorting by Name..." << endl;
    sortByName(s, SIZE);
    for (int i = 0; i < SIZE; i++)
        s[i].displayRec();
    cout << endl;
    cout << "...Sorting by GPA..." << endl;
    sortByGPA(s, SIZE);
    for (int i = 0; i < SIZE; i++)
        s[i].displayRec();
    cout << endl;
    return 0;
}

void stuRec::fillRec()
{
    cout << "Enter name, credit, gpa: ";
    cin >> name >> credits >> gpa;
}

```

```

void stuRec::displayRec()
{
    cout << "Name: " << name << endl;
    cout << "Credit: " << credits << endl;
    cout << "GPA: " << gpa << endl << endl;
}
void sortByName(stuRec s[], int n)
{
    for (int pass = 1; pass < n; pass++)
        for (int c = 0; c < n - pass; c++)
            if (s[c].getName() > s[c + 1].getName())
                swap(s[c], s[c + 1]);
}
void sortByGPA(stuRec s[], int n)
{
    for (int pass = 1; pass < n; pass++)
        for (int c = 0; c < n - pass; c++)
            if (s[c].getGPA() > s[c + 1].getGPA())
                swap(s[c], s[c + 1]);
}
string stuRec::getName()
{
    return name;
}
double stuRec::getGPA()
{
    return gpa;
}
void swap(stuRec &s1, stuRec &s2)
{
    stuRec temp = s1;
    s1 = s2;
    s2 = temp;
}

```

```
cs.wpunj.edu - PuTTY
-bash-3.2$ date
Sun Oct 21 16:33:06 EDT 2018
-bash-3.2$ pwd
/students/abdalkam
-bash-3.2$ ls
F2018          assign2.cpp    f2018          local.login    try.cpp
a.out          assign3.cpp    first.cpp      local.profile
assign.cpp     assign4.cpp    local.cshrc    pico.save
-bash-3.2$ g++ assign4.cpp
-bash-3.2$ ls
F2018          assign2.cpp    f2018          local.login    try.cpp
a.out          assign3.cpp    first.cpp      local.profile
assign.cpp     assign4.cpp    local.cshrc    pico.save
-bash-3.2$ a.out
Enter name, credit, gpa: Juri 98 1.2
Enter name, credit, gpa: Bayonetta 12 4.9
Enter name, credit, gpa: Vegita 34 6.7
Name: Juri
Credit: 98
GPA: 1.2

Name: Bayonetta
Credit: 12
GPA: 4.9

Name: Vegita
Credit: 34
GPA: 6.7

...Sorting by Name...
Name: Bayonetta
Credit: 12
GPA: 4.9

Name: Juri
Credit: 98
GPA: 1.2

Name: Vegita
Credit: 34
GPA: 6.7
```


...Sorting by GPA...

Name: Juri

Credit: 98

GPA: 1.2

Name: Bayonetta

Credit: 12

GPA: 4.9

Name: Vegita

Credit: 34

GPA: 6.7

-bash-3.2\$