

```

Script started on Sun 14 Oct 2012 09:25:42 PM CDT
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at input.dat
nAMe = student student
id = 29358
GpA = 4.5
gender=male

name = another student
Id =32597
Gpa = 2.7
gender= f

name= third stude
ID= 23539
GPA =3.8
gender = m
ID=32523
name=Fourth Student
gender=f
GPA=4.2

town=Williamsburg
gender=male
GPA = 3.1
name= Fifth Student
name=YET another
GPA=1.1

ID=29572
name= last one
GPA=2.2
gender=f
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at student.h
#ifndef STUDENT_H_INC
#define STUDENT_H_INC

#include<string>
#include<fstream>

using namespace std;

class student
{
    string name;
    long id;
    double gpa;
    char gender;

public:

    student(void) : name(""), id(0), gpa(0.0), gender('\0') { }

    student(const student & s) : name(s.name), id(s.id),
                                gpa(s.gpa), gender(s.gender) { }

    student(const string newname, const long newid, const double newgpa,
            const char newgender) : name(""), id(0), gpa(0.0), gender('\0')
    {
        set_name(newname);
        set_id(newid);
        set_gpa(newgpa);
        set_gender(newgender);
    }

    bool print(ofstream & output) const;
    bool read(ifstream & input);

```

```

    string get_name(void) const {return name;}
    long get_id(void) const {return id;}
    double get_gpa(void) const {return gpa;}
    char get_gender(void) const {return gender;}

    void set_name(string newname)
    {
        name = newname;
        return;
    }
    void set_id(long newid)
    {
        id = newid;
        return;
    }
    void set_gpa(double newgpa)
    {
        gpa = newgpa;
        return;
    }
    void set_gender(char newgender)
    {
        gender = newgender;
        return;
    }
};

long strtolong(string & str);

double strtodub(string & str);

short char_to_ones(const char & c);

# endif
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at student\033[Knt.cpp
#include "student.h"
#include <string>
#include <cctype>
#include <iostream>
#include <fstream>

using namespace std;

bool student::print(ofstream & output) const
{
    if (output.is_open() && output.good())
    {
        output << "\nname = " << name
                << "\nID = " << id
                << "\nGPA = " << gpa
                << "\ngender = " << gender << "\n";
    }

    return output.good();
}

bool student::read(ifstream & input)
{
    string line, label, value;
    string::iterator vit;
    string::size_type pos;
    bool labels[4] = {false, false, false, false};
    bool endblock;
    long filepos;
    short i;

```

```

while (!input.eof() && !endblock)
{
    filepos = input.tellg();

    getline(input, line);

    pos = line.find('=');
    label = string(line, 0, pos);
    value = string(line, pos+1, line.length()-1);
    vit = value.begin();

    for (i=0; i<label.length(); i++)
    {
        label[i] = toupper(label[i]);
    }
    while(isspace(*vit))
    {
        value.erase(vit);
        vit = value.begin();
    }

    if (label.find("NAME") != string::npos)
    {
        if (!labels[0])
        {
            set_name(value);
            labels[0] = true;
        }
        else
        {
            endblock = true;
            input.seekg(filepos);
        }
    }
    else if (label.find("ID") != string::npos)
    {
        if (!labels[1])
        {
            set_id(strtolong(value));
            labels[1] = true;
        }
        else
        {
            endblock = true;
            input.seekg(filepos);
        }
    }
    else if (label.find("GPA") != string::npos)
    {
        if (!labels[2])
        {
            set_gpa(strtodub(value));
            labels[2] = true;
        }
        else
        {
            endblock = true;
            input.seekg(filepos);
        }
    }
    else if (label.find("GENDER") != string::npos)
    {
        if (!labels[3])
        {
            set_gender(value[0]);
            labels[3] = true;
        }
        else
        {

```

```

            endblock = true;
            input.seekg(filepos);
        }
    }

    if (!labels[0])
    {
        value = "";
        set_name(value);
    }
    if (!labels[1])
    {
        set_id(0);
    }
    if (!labels[2])
    {
        set_gpa(0.0);
    }
    if (!labels[3])
    {
        value = "";
        set_gender(value[0]);
    }

    return endblock;
}

long strtolong(string & str)
{
    long answer=0, t;
    bool neg = false;
    string::iterator it = str.begin();

    if (*it == '-' || *it == '+')
    {
        if (*it == '-')
        {
            neg = true;
        }
        str.erase(it);
    }

    for (it = str.end()-1, t=1; it >= str.begin() && t < 2147483647; it--,t*=10)
    {
        answer += (char_to_ones(*it) * t);
    }

    if(neg)
    {
        answer = -answer;
    }

    return answer;
}

double strtodub(string & str)
{
    double pre = 0.0, post = 0.0, t = 0.0;
    string::size_type pos;
    string::iterator it = str.begin();
    bool neg = false;

    if (*it == '-' || *it == '+')
    {
        if (*it == '-')
        {

```

```

        neg = true;
    }
    str.erase(it);
}

pos = str.find('.');

for (it = str.end()-1, t=1.0; it > str.begin() + pos; it--, t*=10.0)
{
    post += (static_cast<double>(char_to_ones(*it)) * t);
}

post = post/t;

for (it = str.begin()+pos-1, t=1.0; it >= str.begin(); it--, t*=10.0)
{
    pre += (static_cast<double>(char_to_ones(*it)) * t);
}

pre += post;

if(neg)
{
    pre = -pre;
}

return pre;
}

short char_to_ones(const char & c)
{
    short answer;
    switch (c)
    {
        case '0':
            answer = 0;
            break;
        case '1':
            answer = 1;
            break;
        case '2':
            answer = 2;
            break;
        case '3':
            answer = 3;
            break;
        case '4':
            answer = 4;
            break;
        case '5':
            answer = 5;
            break;
        case '6':
            answer = 6;
            break;
        case '7':
            answer = 7;
            break;
        case '8':
            answer = 8;
            break;
        case '9':
            answer = 9;
            break;
        default:
            answer = 0;
    }
    return answer;
}

```

```

\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at fi\033[Klecopy.cpp
#include<iostream>
#include<string>
#include<fstream>
#include<vector>
#include<iomanip>
#include "student.h"

using namespace std;

int main(void)
{
    string filename;
    ifstream infile;
    ofstream outfile;
    student singlestu;
    vector<student> stuvec;
    vector<student>::iterator it;

    cout << "\n\n\tWelcome to the file copier!\n"
           "Input file name: ";
    getline(cin, filename);
    infile.open(filename.c_str());

    infile.peek();
    while ( !infile.eof() )
    {
        singlestu.read(infile);
        stuvec.push_back(singlestu);
        infile.peek();
    }
    infile.close();
    infile.clear();

    cout << "\nFile read successfully. Enter the output file name: ";
    getline(cin, filename);
    outfile.open(filename.c_str());

    for (it = stuvec.begin(); it < stuvec.end(); it++)
    {
        it->print(outfile);
    }

    outfile.close();
    outfile.clear();

    return 0;
}
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ .
/filecopy.out

Welcome to the file copier!
Input file name: input.dat

File read successfully. Enter the output file name: output.dat
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at output.dat

name = student student
ID = 29358

```

```
GPA = 4.5
gender = m

name = another student
ID = 32597
GPA = 2.7
gender = f

name = third stude
ID = 23539
GPA = 3.8
gender = m

name = Fourth Student
ID = 32523
GPA = 4.2
gender = f

name = Fifth Student
ID = 0
GPA = 3.1
gender = m

name = YET another
ID = 29572
GPA = 1.1
gender = \000

name = last one
ID = 0
GPA = 2.2
gender = f
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ e
xit
exit

Script done on Sun 14 Oct 2012 09:26:56 PM CDT
```