```
Script started on 2021-04-22 18:19:21-0500
m_sadaf1@ares:~$ pwd
/home/students/m_sadaf1
m_sadaf1@ares:~$ cat copy_xyz_main.cpp
#include <fstream>
#include <iostream>
#include <limits>
#include <string>
#include <sstream>  // Strings to/from streams
#include "copy_xyz_student.h"

using namespace std;

// Checks if the file exists

inline bool file_exist(const string & name)
{
    bool ret;
    ifstream file;
    file.open(name); // C-string
    if (!file)
    {
        ret = false;
    }
    else
    {
        ret = true;
    }

    file.close();
    file.clear();
    return ret;
}

int main(void)
{
    ifstream original;
    ofstream copy;
    string filename;

    cout << "\nEnter the name of the file to copy: ";
    getline(cin, filename);
    original.open(filename);

    // Loop to ask file name if it can't find it

    while(!original)
    {
        original.close();
        original.clear();
        cout << "\nFile does not exist.\nEnter file name: ";
        getline(cin, filename);
        original.open(filename);
    }

    cout << filename << "selected to copy" << endl;

    cout << "Enter a new output file name: ";
    getline(cin, filename);
    while (file_exist(filename))
    {
        cout << "Error: file" << filename <<
                "already exists."
                "\nEnter a new output file name: ";
        getline(cin, filename);
    }

    copy.open(filename);
    cout << filename << "selected as output file." << endl;

    Student user;

    // Peeks to check if the file is blank, then will
    // return eof

    original.peek();
    while (!original.eof())
    {
        user.read(original);
        user.write(copy);
        original.peek();
    }

    // Close streams when done

    copy.close();
    original.close();
```

```
    cout << "Hit ENTER to exit." << endl;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    return 0;
}

m_sadaf1@ares:~$ cat copy_xyz_student.h
#ifndef COPY_XYZ_STUDENT_H
#define COPY_XYZ_STUDENT_H

#include <fstream>
#include <string>
#include <iostream>

class Student
{
    long ID;
    double GPA;
    char grade;
    std::string name;

public:
        // Constructor          //Default Constructor
        Student() :ID(), GPA(), grade(), name() {}

        // Accessors
        long get_ID() const { return ID; }
        double get_GPA() const { return GPA; }
        char get_grade() const { return grade; }
        std::string get_name() const { return name; }

        // Files

        void read(std::istream& in) // Reads input from file
        {
            using namespace std;
            in >> ws;
            while (in.peek() == '#')
            {
                // Skips to next line if found
                in.ignore(numeric_limits<streamsize>::max(), '\n');
            }
            getline(in, name); // Reads name till new line

            // Finds pos of first #

            size_t pos = name.find('#');

            if (pos != std::string::npos)
            {
                name.erase(pos); // Erases starting from #
            }

            while (in.peek() == '#')
            {
                in.ignore(numeric_limits<streamsize>::max(), '\n');
                in >> ws;
            }
            in >> ID >> ws;

            while (in.peek() == '#')
            {
                in.ignore(numeric_limits<streamsize>::max(), '\n');
                in >> ws;
            }
            in >> GPA >> ws;

            while (in.peek() == '#')
            {
                in.ignore(numeric_limits<streamsize>::max(), '\n');
                in >> ws;
            }
            in >> grade >> ws;
        }

        void write(std::ostream& out) // Writes data to file
        {
            using namespace std;
            out << name << endl << ID << endl << GPA
                    << endl << grade << endl << endl;
        }

        // Mutators
        void set_ID(const long s_ID) { ID = s_ID; }
        void set_GPA(const double s_GPA) { GPA = s_GPA; }
        void set_grade(const char s_grade) { grade = s_grade; }
        void set_name(const std::string s_name) { name = s_name; }
```

```
    };

#endif /* COPY_XYZ_STUDENT_H */

m_sadaf1@ares:~$ CPP copy_xyz_main
copy_xyz_main.cpp***


m_sadaf1@ares:~$ CP.out
bash: ./: Is a directory
m_sadaf1@ares:~$ ./copy_xyz_main.out

Enter the name of the file to copy: exit

File does not exist.
Enter file name:

File does not exist.
Enter file name:

File does not exist.
Enter file name:

File does not exist.
Enter file name:

File does not exist.
Enter file name: ^C
m_sadaf1@ares:~$ exit
exit

Script done on 2021-04-22 18:21:23-0500
```