```
Script started on 2021-10-12 20:51:05-05:00 [TERM="xterm" TTY="/dev/pts/12" COLUMNS
k_gandhi6@ares:~$ cat XYZ.info
Name: Kush Gandhi

Tittle: Copy XYZ (Lab)

Class: CSC122-001

Levels: 1 + 3.5 = 4.5

- allows place blank lines

- allows whole line comments

- allows end of line comments


Description: Takes in user data file to copy

the data stored in the file and transfer to

another file and allows comments.

k_gandhi6@ares:~$ cat XYZ.h
#ifndef DATA_CLASS_H_INC

#define DATS_CLASS_H_INC


#include <iostream>

using namespace std;


class Data

{

public:

    Data();

    void read(ifstream& input);

    void write(ofstream& output);


private:

    static const int length = 30;

    int idNum;

    double gpa;

    char name[length], grade;

};

#endif

k_gandhi6@ares:~$ cat XYZ.cpp
#include <fstream>

#include <iostream>

#include <cstring>

#include <climits>

#include <cctype>

#include <string>

#include "XYZ.h"

using namespace std;


int main() {

    ifstream data_file;

    ofstream outfiles;

    Data data;

    string fileName1, fileName2;


    cout << "\t\tWelcome to the People Copying Data Program!!";

    cout << "\n\nPlease enter the name of your data file: ";

    getline(cin, fileName1);

    data_file.open(fileName1);


    while (!data_file) {//loop check input file
```

```
    data_file.close();
    data_file.clear();
    cout << "\nI'm sorry, I could not open '" << fileName1
        << "'. Please enter another name: ";
    getline(cin, fileName1);
}
cout << "\nFile '" << fileName1 << "' opened successfully!\n";

cout << "\nPlease enter the name of your data file: ";
getline(cin, fileName2);
outfiles.open(fileName2);

while (!outfiles) {//loop check for output file
    outfiles.close();
    outfiles.clear();
    cout << "\nI'm sorry, I could not open '" << fileName2
        << "'. Please enter another name: ";
    getline(cin, fileName2);
}
cout << "\nFile '" << fileName2 << "' opened successfully!\n";

cout << "\nCopying data from '" << fileName1 << "' to '"
    << fileName2 << "'...\n";
data_file.peek();
while (!data_file.eof()) {
    data.read(data_file);
    data.write(outfiles);
        data_file.peek();
    }
    cout << "\nDone copying data!\n" << "\nThank you for using the PCP!\n"
        << "\nEndeavour to have a tremendous day" << endl;
    return 0;
}

Data::Data() : idNum(000000), gpa(0.0), grade('A') { name[0] = '\0'; }

void comment_Line(char pound, istream& input)
{//function for any # or \n is removed
    input >> ws;
    while (input.peek() == pound || input.peek() == '\n') {
        input.ignore(INT_MAX, '\n');
        input >> ws;
    }
    return;
}

void Data::read(ifstream& input) {
    comment_Line('#', input);
    input.getline(name, length);
    char *check = strchr(name, '#');
    if (check != NULL) {//if comment then set the null chr.
        *check = '\0';
    }
    comment_Line('#', input);
```

```
    input >> idNum;

    comment_Line('#', input);

    input >> gpa;

    comment_Line('#', input);

    input >> grade;

    comment_Line('#', input);

}


void Data::write(ofstream& output) {

    output << name << endl;

    output << idNum << endl;

    output << gpa << endl;

    output << grade << endl << endl;

}
```

```
k_gandhi6@ares:~$ cat Students.txt
# help me

Jason James # name

123456 # id number

9.2 # gpa

B # comment

Tammy James # comment

123457 11.2 A

Familiar Kensei James

123458 # what do i do here

5.6 D

Quincy 2005 is Awesome

110121 8.4 # hi random comment

B # grade letter

# that is me

Kush Gandhi # something

777777

4.0 A

k_gandhi6@ares:~$ CPP XYZ
XYZ.cpp***


k_gandhi6@ares:~$ ./XYZ.out
               Welcome to the People Copying Data Program!!

Please enter the name of your data file: Students.txt

File 'Students.txt' opened successfully!

Please enter the name of your data file: Test.txt

File 'Test.txt' opened successfully!

Copying data from 'Students.txt' to 'Test.txt'...

Done copying data!

Thank you for using the PCP!

Endeavour to have a tremendous day
k_gandhi6@ares:~$ cat Test.txt
Jason James
123456
9.2
B

Tammy James
123457
11.2
A

Familiar Kensei James

123458
5.6
D

Quincy 2005 is Awesome

110121
8.4
B
```

Kush Gandhi
777777
4
A

k_gandhi6@ares:~$ cat XYZ.tpq
1. What weird behavior does open exhibit for output files by default?
   How do we fix this problem?

   - It has the tendency to truncate the file. Since we don't want
   that to happen, we need to code it properly and protect it
   in order for it to work.

2. How much does spacing matter in the input file? The output file?
   (Hint: Would it matter if it weren't present at all? If there
   were many, many spaces?)

   - The answer weighs both side becasue spacing can be impotant.
   If we take a normal input such as cin will ignore the spaces but
   using getline will not ignore it. The data does need to be spaced
   so our program can read the data.

3.Problems with the (C)string piece of data:

3a. What problem might you have with the (C)string data
    (being as it is 'mixed' with so many other data types
    in this file: numbers and characters and such)?
    (Hint: Is the (C)string data one or multiple words?)
    Is this difficult to fix? What assumption did you make to
    solve this problem?

    - The C-string can be many words but we should consider
    about the user using our program on what they eneter. If so,
    we need to use a getline function for the C-string data.

3b. If the (C)string had to beplaced afterthe other data at the
    end of the data group/block, what problem might arise? How do
    we typically avoid this situation (again, assuming the data
    has to be in that order)? [Assume you have re-written your
    code to deal with the new data order but do not do so.]

    - if the C-string was plced at the end it will still read the data
    but not the new line since we only extract the data.

3c. Think about, but do not fix, the potential problem of the user's
    (C)string being longer than you had anticipated. (Answer this
    question even if you used the string class to code your program!)

    - We could use the getline function to read the name. We would place
    it in the 2nd argument which detects the length of the C-string.

4a. What function is used to tell when you've reached the end of a stream?

    - We use the eof() function called end of file.

4b. Can this function be used on the keyboard stream?

- Yes, you are able to do it by crtl-z.

5. How do you pass a stream to a function?

- We need to pass it by reference.

6. Why is it a good idea to make input functions ignorant of whether
or not a stream is cin or a file? Output functions/cout/file?

- You can do it that way but it's not a good approach. The
other way is to use the ifstream and ofstream.

7. Why do we close files?

- We close files becasue once we are done using them,
we ensure to clean the crap out since it will take up
unecessory data. So, we can free up space.

```
k_gandhi6@ares:~$ exit
exit

Script done on 2021-10-12 20:54:03-05:00 [COMMAND_EXIT_CODE="0"]
```