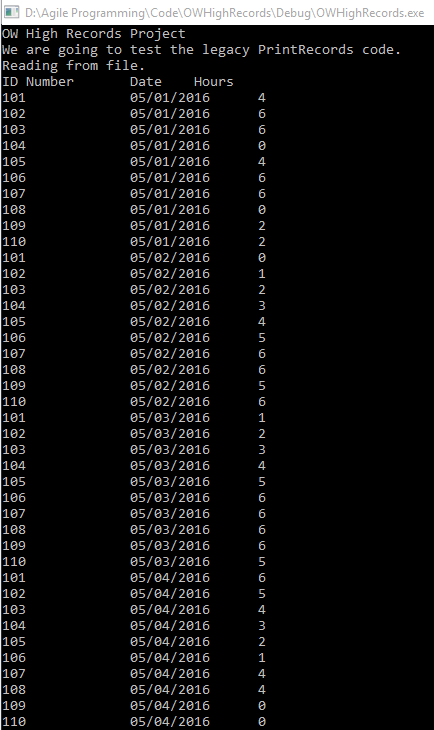
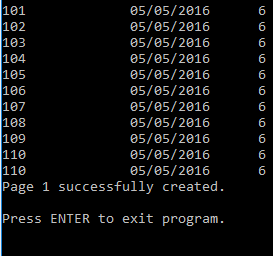
Sarah Wood

Agile Programming

P2-Deliverable 3

Screen Caps





The code executed, but the output file was blank. On further examination,

bool PrintRecords::Prints(char \*FileName, char \*Page)

{

ofstream outMyStream(FileName, ios::app);

if (outMyStream.is\_open()){

outMyStream << Page;

outMyStream.close();

return true;

}

the legacy code presented a problem. The line

ofstream outMyStream(FileName, ios::app);

creates an output file variable, names it, locates it in the filesystem, and gives it directions. It does NOT open it. The line

if (outMyStream.is\_open()){

only works with the file if it’s open, which it’s not.

It was a great refresher on finer details of pointers, file I/O, and char arrays. I tried a number of things to make it work before I caught the “Duh, the output file is never opened.”

I already have code for file I/O that works, so in my final release, I’ll use my own code. I don’t really need print-related objects.

Source Code (relevant parts)

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

PrintRecords.cpp

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

// #include "stdafx.h" //SJW -- VS could not find file

#include "PrintRecords.h"

#include <fstream>

#include <string>

using namespace std;

PrintRecords::PrintRecords() //SJW changed 05/13/2016. Added "s" to instance to fix bug.

{

}

PrintRecords::~PrintRecords(void)

{

}

bool PrintRecords::Prints(char \*FileName, char \*Page)

{

ofstream outMyStream(FileName, ios::app);

if (outMyStream.is\_open()){

outMyStream << Page;

outMyStream.close();

return true;

}

return false;

}

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

Source.cpp

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

void unitTestingFunction(Student \*stArr, StudentDailyAttendance \*att, Teacher \*tArr, Admin \*aArr)

{

//build an output format that is a char array divided into "pages".

//(50 lines fit on one page) X (80 char/line) = 4000 char per page

char Page1[4000] = {};

char Page2[4000] = {};

//open file, read

fstream attRecordIN;

string tempDate;

int tempID;

double tempHrs;

attRecordIN.open("PrintRecordsTest.txt");

if (attRecordIN)

{

cout << "Reading from file." << '\n';

cout << "ID Number" << '\t' << "Date" << '\t' << "Hours" << '\n';

attRecordIN.ignore(256, '\n'); //ignore the first line: it's just headers

for (int i = 1; !attRecordIN.eof(); i++) // "i" will correspond to the unique attendance entry ID.

{

attRecordIN >> tempID >> tempDate >> tempHrs;

att[i].IDNumber = tempID;

att[i].Date = tempDate;

att[i].TimeInClass = tempHrs;

cout << setprecision(2);

cout << tempID << '\t' << '\t' << tempDate << '\t' << tempHrs << '\n';

}

}

else

{

cout << "Record File Input Error." << endl;

}

attRecordIN.close();

//write records to "page" char arrays

attRecordIN.open("PrintRecordsTest.txt");

int counter = 0;

while (attRecordIN)

{

char ch;

attRecordIN.get(ch);

if (counter<4000)

{

Page1[counter] = ch;

}

else if (counter<8000)

{

Page2[counter] = ch;

}

counter++;

}

PrintRecords TestPrint;

if (TestPrint.Prints("testPrint.txt", Page1))

{

cout << "Page 1 successfully created." << endl;

}

else

{

cout << "Page 1 file error." << endl;

}

}