CS 1428
Lab 12 Sections L06 and L19

Jared Wallace

Structs

Structs are what we use to ”group” different datatype variables into one, easy to access container. For example, if we had a program that dealt with employees that was used by the HR department, we might have a struct called Employee, that contained fields for payrate, social security number, and name.

```
struct Employee
{
   string name;
   float payrate;
   int social;
};
```

We would then use our struct like this:

```
// inside of main
Employee Bender,
         Fry,
         Leela;
Bender.name = "Bucket Of Bolts";
Bender.payrate = 110.00;
Bender.social = 000101010;
Fry.name = "Slacker";
Fry.payrate = 0.00;
Fry.social = 008675309;
...
```

You'll notice that in order to access the different parts of the struct (called member variables), we use the dot operator.

It is absolutely crucial that you understand the difference between a struct declaration, which is just like a blueprint, and the instantiation (creation) of a struct object, which is like building a house. With our one struct declaration, Employee, we were able to create (instantiate) 3 objects, called Bender, Fry and Leela.

When we call a function and need to pass in a struct object, we can do so by simply passing the name of the object. For example, if we wanted to pass Bender into a function called BiteMyShinyMetalA**, we would do so like this:

```
void BiteMyShinyMetalA**(Employee);
int main()
{
   Employee Bender;
   Bender.name = "Bucket Of Bolts";
   Bender.payrate = 110.00;
   Bender.social = 000101010;

   BiteMyShinyMetalA**(Bender);
   return 0;
}
void BiteMyShinyMetalA**(Employee Bender)
{
   cout << "I'm back baby!";
   cout << "exclaimed " << Bender.name;
}
```

This can obviously be very convenient.

1. Today I want you to write a program that reads in student's names, along with their final test grade from an input file (input.txt, provided). Then I want the program to output each student's name, followed by their average, and finally the corresponding letter grade of their final into an output file called ”output.txt”.

2. Student data should be stored in a struct variable of the type studentRecord, which has three components:

   - First (string)
   - Last (string)
   - average (double)
   - grade (char)

3. The class has 15 students. Use an array with 15 elements of the struct data type studentRecord. Your program must contain at least the following functions:

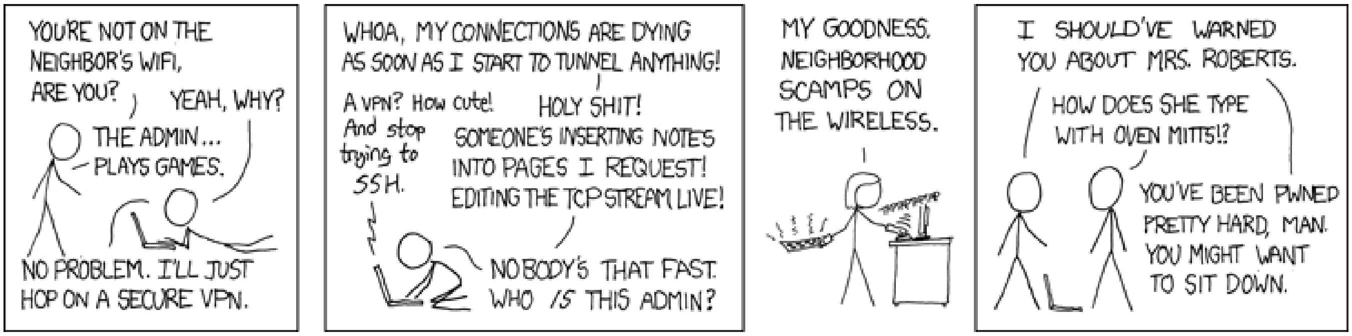
   - A function to read the students' data into the array
   - A function to assign the corresponding letter grade to each student **Note:** Round up! An 89.5 = A)
   - A function to print out data to the output file (output.txt)

4. Example:

   Student Name: Doctor Zoidburg Average: 89 Grade: B

   Student Name: Richard Nixon Average: 56 Grade: F

Upload your source code through the homework upload utility. Place a printout of you source code behind this page and staple before turning it in at my desk.


YOU'RE NOT ON THE NEIGHBOR'S WIFI, ARE YOU?
YEAH, WHY?
THE ADMIN... PLAYS GAMES.
NO PROBLEM. I'LL JUST HOP ON A SECURE VPN.
WHOA, MY CONNECTIONS ARE DYING AS SOON AS I START TO TUNNEL ANYTHING!
A VPN? How cute! And stop trying to SSH.
HOLY SHIT! SOMEONE'S INSERTING NOTES INTO PAGES I REQUEST! EDITING THE TCP STREAM LIVE!
NOBODY'S THAT FAST. WHO IS THIS ADMIN?
MY GOODNESS. NEIGHBORHOOD SCAMPS ON THE WIRELESS.
I SHOULD'VE WARNED YOU ABOUT MRS. ROBERTS.
HOW DOES SHE TYPE WITH OVEN MITTS!?
YOU'VE BEEN PWNED PRETTY HARD, MAN. YOU MIGHT WANT TO SIT DOWN.

If you're not cool enough to do it manually, you can look up tools like Upside-Down-Ternet for playing games with people on your wifi.