

March 3, 2014

```
struct Student
```

```
{
```

```
    string last;
```

```
    string first;
```

```
    int mid Got; // what they scored
```

```
    int mid Possible; // what they could have scored
```

```
    // two more pairs
```

```
    // for attendance & find
```

```
    int hws[20];
```

```
    int proj[10];
```

```
    int hw Count;
```

```
    int proj Count;
```

```
    double ave;
```

```
};
```

gradebook.h

```
void gradebook (istream & in, ostream & out)
```

```
{
```

```
    // declare int weight/percentage variables
```

```
    Student students[100];
```

```
    int howMany;
```

```
    string command, second;
```

```
    in >> command;
```

```
    while (!in.fail())
```

```
    {
```

```
        in >> second;
```

```
if ( command == "load" || command == "Load" )
{
    load File ( second, students, // file weight, how Many);
    out << "Loaded: " << second << "...",
}
else ( command == "Display" )
{
    }
}
ih >> command;
}
}
```

```
void loadfile (string file, Student students[],
int &aw, int &pw, int &mw, int &hw, int &fw, int &ns)
{
```

```
    string junk;
    ifstream in (file.c_str());
```

```
    ns = 0;
```

```
    in >> junk >> aw;
```

```
    "      "      nw;
```

```
    "      "      fw;
```

```
    "      "      pw;
```

```
    "      "      hw;
```

```
    in.ignore (200, '\n');
```

```
    getline (in, students[ns].last, ',');
```

```
    while (!in.fail())
```

```
    {
```

```
        students[ns].hwCount = 0;
```

```
        students[ns].projCount = 0;
```

```
        getline (in, student[ns].first);
```

```
        string cat;
```

```
        int got, possible;
```

```
        in >> cat;
```

```
        while (cat != "final:")
```

```
        {
```

```
            in >> got >> possible;
```

```
            if (cat == "Attendance:")
```

```
            {
```

```
                student[ns].hwCount = possible;
```

• \equiv - method

else if (cat == "Homework")

{

int cnt = students[ns].hwCount;

students[ns].hw[cnt] = got;
students[ns].hwCount++;

}

else

{

projects

{

ih >> cnt;

// end of cat != final

ih >> got >> possible;

students[ns].finalGot = got;

students[ns].finalPossible = possible;

// compute average

ns++;

in.ignore(200, '\n');

getline(in, students[ns].last, ',');

} // end of !in.eof()

// end of load

Computing the Average

$$\text{ave} = \overbrace{\text{got} \times \text{weight}}^{\text{attend}} + \underbrace{x}_{m} + \underbrace{y}_{f} + \underbrace{hw}_{w} + \underbrace{prg}_{w}$$

$$\begin{aligned} \text{got mid term} &= 80 \\ \text{possible} &= 100 \end{aligned}$$

$$\text{ave} = \left(\frac{80}{100} \right) \times 20 = 16$$

$$100 \overline{)80}$$

$$\text{ave} = \frac{1.0 \times 80}{100} \times 20 = 16$$