March 3, 2014

```
struct Student
{
    string last;
    string first;
     int attendPossible;
     int attendGot;
    // two more pairs
    // for midterm & final
     int hws[20];
     int projs[10];
     int hwCount;
     int projCount;
      double ave;
  };
```

gradebook.h

```
void gradebook(istream& in, ostream &out)
{
      string command, second;
      Student students[100];
      int   attendWeight;
       "      homeWeight;
       // 3 more for prj, mid, final;
      int  numberOfStudents;

       in >> command;                    load basic.txt
      while( !in.fail())
      {
           in>> second;
           if( command == "load" || command == "Load")
           {
                 loadFile(second, studentss, numberOfStudents,
                       //all the weights);
              out << "Loaded: " << second << " # students:"
                    << numberofStudents << endl;
           }
           else if (command == "Display" ...   )
           {
           }
           else
           {
           }
           in >> command ;
      }
}
```

```
void load file( string file, Student students[],
     int& nS, int& aW, int& pW, int& hw, int &mW, int& fW
{
     string junk;
      int  possible, got;
     ifstream  in (file.c_str());

      nS = 0;

      in >> junk >> aw;
      in >> junk >> mw;
       "      "       fw;
       "      "       hw;
       "      "       pw;

      in.ignore(200, '\n');
     getline( in, students[nS].last, ',');
      while (! in.fail())
     {
          getline(in, students[nS].first );
           string  cat;
             students[ns].hwCount= 0;
             students[nS].prjCount= 0;
             in >> cat;
             while ( cat != "Final:" )
             {
                  in >> got >> possible;
```

```
            if( cat == "Attendance:")
            {
                students(nS].attendGot = got;
                students [nS]. attendPossible = possible;
            }
            else if( cat == "Midterm:")
            {    ≡
            }
            else if (cat == "Homework:")
            {
                int  cnt= students(nS].hwCount;
                students(nS]. hws[cnt] = got;
                students (nS].hwCount ++;
            }
            else
            {   //proj
            }
        in >> cat;
      }// end while cat! = final
        in >> got >> possible;
      students (nS]. finalGot = got;
      students [nS]. finalPossible = possible;

    // compute average

    nS++;
    in.ignore (200, '\n');
    getline in, students [nS]. last, ',');
  }// end of while !in.fail

}// end of function
```

$$ave = \overbrace{\frac{got}{possible} \times weight}^{attend} + \overbrace{\qquad\qquad}^{final} + \overbrace{\qquad\qquad}^{midterm}$$

$$+ \overbrace{\qquad\qquad}^{hw} + \overbrace{\qquad\qquad}^{prj}$$

$$\frac{got}{possible} \times weight \qquad \frac{1.0 \times 80}{100} \times 20 = 16$$

$$\begin{array}{r} 0 \\ 100 \overline{)\,80} \end{array}$$