CSC240

Sample Program with Array of Records - C++

```
//  jetengines.cpp  --  display IDs of
//    engines with above-average scores

#include <iostream>
using namespace std;
#include <fstream>

struct JetEngine
{
  int ID;
  double score;
};

void loadEngines
   ( JetEngine engines[], int& nEngines );
double computeAverage
   ( JetEngine engines[], int nEngines );
void displayAboveAverage( JetEngine engines[],
     int nEngines, double averageScore );

void main( void )
{
  JetEngine engines[100];
  int nEngines;
  double averageScore;

  cout << endl
     << "Display Above-average Engines"
     << endl << endl;

  loadEngines( engines, nEngines );
  averageScore =
     computeAverage( engines, nEngines );
  displayAboveAverage
     ( engines, nEngines, averageScore );
  cout << endl << "Have a nice day"
       << endl << endl;
}

void loadEngines
   ( JetEngine engines[], int& nEngines )
{
  char another;
  nEngines = 0;
  do
  {
    cout << endl << "For engine #"
         << nEngines+1 << "..." << endl;

    cout << "Enter the engine ID: ";
    cin >> engines[ nEngines ].ID;

    cout << "Enter the engine score: ";
    cin >> engines[ nEngines ].score;

    ++nEngines;

    cout << endl << "Another engine (y/n)? ";
    cin >> another;
  }
  while ( another != 'n' );
}
```

```
double computeAverage
  ( JetEngine engines[], int count  )
{
  int ix;
  double sum, average;

  sum = 0;
  for ( ix = 0; ix < count; ++ix )
    sum = sum + engines[ix].score;

  if ( count > 0 )
    average = sum / count;
  else
    average = 0;

  return average;
}


void displayAboveAverage( JetEngine engines[],
    int nEngines, double averageScore )
{
  int ix;

  cout << endl << endl
       << "Above-average Engines"
       << endl << endl;

  cout << "\tScore\tID\tProduced By" << endl;

  for ( ix = 0; ix < nEngines; ++ix )
  {
    if ( engines[ix].score > averageScore )
    {
      cout << "\t" << engines[ix].score
           << "\t" << engines[ix].ID
           << endl;
    }
  }

  cout << endl << "Average Score="
       << averageScore << endl << endl;
}
```