Our Starting Point -- structs And Objects, Reading

In your prerequisite course you should have learned how to design and apply objects with code like this (applying coding conventions learned in the previous module):

```
// Programmer: First Last
// Programmer's ID: 1234567

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

#include <cstdlib> // for atoi and atof

struct Student
{
  string name;
  int studentID;
  float gpa;
};

Student input( );
void output(const Student&);

int main( )
{
  cout << "Programmer: First Last\n";
  cout << "Programmer's ID: 1234567\n";
  cout << "File: " << __FILE__ << endl;

  // create an array of student objects
  const int CAPACITY = 2;
  Student x[CAPACITY]; // an array of objects

  // enter data for each student
  for (int i = 0; i < CAPACITY; i++)
    x[i] = input( );

  // output each student's data
  for (int i = 0; i < CAPACITY; i++)
    output(x[i]);
}

Student input( ) // a value-returning function that returns an object
{
  Student result;

  cout << "Enter a student's name ";
  getline(cin, result.name);
```

```
  char buf[100];
  cout << "Enter " << result.name <<"'s student ID: ";
  cin >> buf; result.studentID = atoi(buf);
  cin.ignore(1000, 10);

  cout << "Enter " << result.name << "'s GPA: ";
  cin >> buf; result.gpa = atof(buf);
  cin.ignore(1000, 10);

  return result;
}

void output(const Student& s) // a void function
{
  cout.width(32);
  cout << s.name;
  cout.width(10);
  cout << s.studentID;
  cout.setf(ios::fixed);
  cout.precision(2);
  cout.width(10);
  cout << s.gpa << endl;
}
```

Study this code listing and make sure that you can copy, paste, and run it. Make sure you understand each of the following aspects of the code before moving on in this reading, because this is what you are expected to know at this point.

1. the struct declaration with its "attributes" (or "fields" or "data members" or "properties"),
2. the use of the "dot operator" (or "member of") to access attributes of the objects,
3. the static array declaration with its two objects,
4. the for-loops to traverse the arrays, reusing "i" as their index,
5. the value-returning function that returns a Student object,
6. the void function with a read-only reference parameter,
7. the user "prompts" with cout,
8. the console output formatting with width and precision specified, and the use of ios::fixed,
9. the console input for name, using getline (allowing for first *and* last name),
10. the console input for numbers, using the "string buffer" method with buf, atoi, and atof,
11. the use of `cin.ignore(1000, 10);` because the program has both `cin>>` *and* `getline(cin` in it,
12. the function prototype placement, and unnamed parameters,
13. the code blocks for C++ and C library includes, and the placement of `using namespace std;`,
14. the code alignment and 2-space indenting, and
15. the programmer identification with comments and cout statements.

And here are some formatting specifics you should know:

1. `cout.setf(ios::fixed)` says to interpret `cout.precision` as #of digits after the decimal, and not as the #of "significant digits",
2. `cout.precision` says how many significant digits to show for floating point numbers *or* how many decimal digits depending on whether `ios::fixed` is set.
3. To unset a setting, it's `cout.unsetf(ios::fixed)`.
4. `cout.width` says how many spaces to allocate to the *very next* output value *only*, right justified.

In case you are completely clear on any of these, contact the professor to discuss them, because otherwise he will expect you to know and apply these in the remainder of this course.