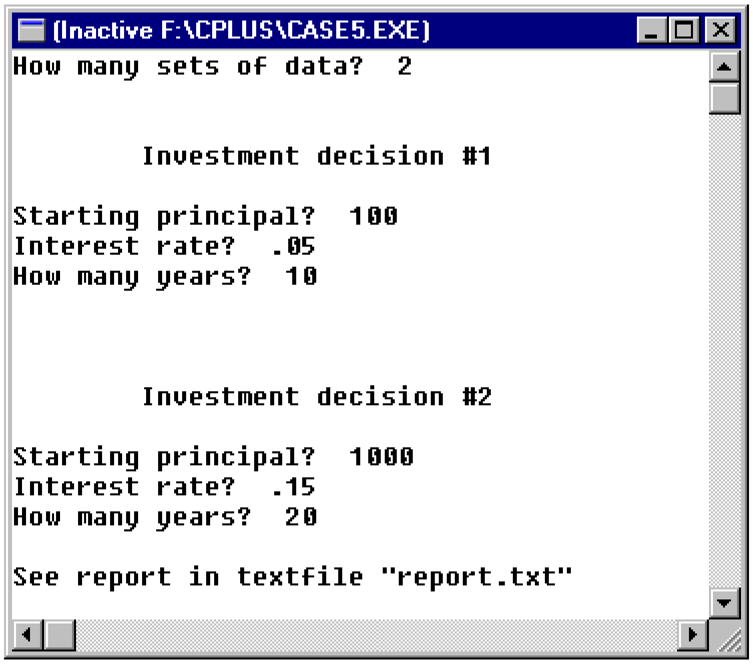
Formatting Output & File I/O

Producing a simple report

```
//case5.cpp
// Compound Interest Program - Case 5
// Using a report textfile
// This program computes interest on some initial investment,
// compounded annually for any number of years.

#include <fstream>
#include <iostream>
#include <iomanip>

int main()
{
   int n, year, years;
   float pzero, p, rate, earned;
   ofstream cprn ("report.txt"); //File contains the report

   cout << "How many sets of data?  ";
   cin >> n;
   for (int count = 1; count <=n; count++) {
      cout << "\n\n\tInvestment decision #" << count << endl;
      cout << "\nStarting principal?  ";
      cin >> pzero;
      cout << "Interest rate?  ";
      cin >> rate;
      cout << "How many years?  ";
      cin >> years;
      cout << endl;
      cprn << setiosflags(ios::fixed |ios::showpoint)
            << setprecision(2);
      p = pzero;
      cprn << "\n\n\tInvestment decision #" << count << endl;
      cprn << "Starting Principal = $" << pzero <<endl
            << "Interest rate = " << rate <<endl<<endl;
      for (year=1; year<=years; year++){
            p = p + p * rate;
            cprn << "After " << setw(3) << year
            << "  years, you will have $"
            << p <<endl;
      } //end for
      earned = p - pzero;
      cprn << "\nTotal interest earned:  $" << earned <<endl;
   }//end for
   cout << "See report in textfile \"report.txt\"\n\n";
   return 0;   //successful termination of program
} //end main
```

ofstream is a class defined in the **fstream.h** header file. We use it to define an object called **cprn**. When we use **cprn** in place of **cout**, the stream output is sent to the **report.txt** textfile rather than to the standard output (screen). This will become clearer when we write our own classes and objects.

Interactive Output Window:



```
How many sets of data?  2


        Investment decision #1

Starting principal?  100
Interest rate?  .05
How many years?  10



        Investment decision #2

Starting principal?  1000
Interest rate?  .15
How many years?  20

See report in textfile "report.txt"
```

Print the contents of report.txt:

```
     Investment decision #1
Starting Principal = $100.00
Interest rate = 0.05

After   1  years, you will have $105.00
After   2  years, you will have $110.25
After   3  years, you will have $115.76
After   4  years, you will have $121.55
After   5  years, you will have $127.63
After   6  years, you will have $134.01
After   7  years, you will have $140.71
After   8  years, you will have $147.75
After   9  years, you will have $155.13
After  10  years, you will have $162.89

Total interest earned:  $62.89


     Investment decision #2
Starting Principal = $100.00
Interest rate = 0.15

After   1  years, you will have $115.00
After   2  years, you will have $132.25
After   3  years, you will have $152.09
After   4  years, you will have $174.90
After   5  years, you will have $201.14
After   6  years, you will have $231.31
After   7  years, you will have $266.00
After   8  years, you will have $305.90
After   9  years, you will have $351.79
After  10  years, you will have $404.56
After  11  years, you will have $465.24
After  12  years, you will have $535.03
After  13  years, you will have $615.28
After  14  years, you will have $707.57
After  15  years, you will have $813.71
After  16  years, you will have $935.76
After  17  years, you will have $1076.13
After  18  years, you will have $1237.55
After  19  years, you will have $1423.18
After  20  years, you will have $1636.65

Total interest earned:  $1536.65
```

More on Formatting Program Ouptut

Standard I/O - <iostream.h>
cin
cout

I/O manipulators - use <iomanip.h> header file

setfill(ch)	set the fill character to ch
setw(n)	set the field width to n
setprecision(n)	set the floating-point precision to n places
setiosflags(flags)	set the format flags for the ios [**i**nput/**o**utput **s**tream]

Note: The setiosflags manipulator is also called a parametized manipulator, since it has parameters (arguments).

format flags for setiosflags()
[these flags may be combined by | operators]

ios::showpoint	always show the decimal point (default is 6 decimal digits)
ios::showpos	display a leading + sign when the number is positive
ios::fixed	display up to 3 integer digits and 2 digits after the decimal point. (For larger values, use exponential notation.)
ios::scientific	use exponential notation to display output
ios::showbase	show base indicator on output
ios::left	left justify output
ios::right	right justify output
ios::skipws	skip whitespace on input

ios::fixed forces all subsequent numbers placed on the cout stream to be placed as if they were floating point values. ios::showpoint is then used to always display a decimal point. Since we often want both of these flags set, we will frequently joint them with the OR operator: setiosflags(ios::fixed || ios::showpoint). Then, setprecision(2) is used to ensure that (say) 2 decimal digits will always print even for numbers like 4 (4.00) or 4.1 (4.10).

Flag / in the current usage, a flag sets a certain condition as active. Activating one flag (e.g. right) may automatically deactivate other flag(s) (e.g. left).

(Bronson p.132) Examples:
[with `ios::fixed` flag set]

manipulator(s)	number	display	comments
`setw(2)`	`3`	`| 3|`	`number fits in field`
`setw(2)`	`43`	`|43|`	`number fits in field`
`setw(2)`	`143`	`|143|`	`field width ignored`
`setw(2)`	`2.3`	`|2.3|`	`field width ignored`
`setw(5)`			`field width 5`

```
setprecision(2) 2.366   |  2.37|   w/ 2 decimal digits

setw(5)
setprecision(2) 42.3    | 42.3|    number fits in field

setw(5)                            field width ignored but
setprecision(2) 142.364 |142.36|   precision spec used
```

File Input/Output

For file input/output need the preprocessor directive:

```
#include <fstream>
```

To open file for output:

```
ofstream outfile ("filename.txt”);
```

or `(“c:\\programs\\reports\\filename.txt”)`

where

ofstream is the name of a class defined in the <fstream> library;
outfile is a programmer-defined object of the ofstream class;
filename.txt is the DOS file name of the text file the program prints to.

To test the file to see whether it is open and ready for output:

```
//condition evaluates to false if file failed to open
if (!outfile) cerr << "Sorry! We are unable to open file!\n";
```

where cerr is the same as the standard output object, cout.

To open file for input:

```
ifstream infile ("filename.dat);
```

to test it:

```
if (!infile) cerr << " Sorry! We are unable to open file!\n";
```

when using ifstream class, the file mode is input by default. But, to protect input-only data from unintentional modification, use:

```
ifstream infile ("filename.dat", ios::in);
```

where in specifies that the file must be opened in input mode.

Example, Programming Assignment #8 - A Grading Problem

Program Listing:

```
//grading.cpp, run under Borland C++ 5.02
//Programming Assignment #8 - A Very Grading Problem
//using strings, functions, static variables,
// input data file, output report file
//Target Type: win32 application; Target Model: console
//author:  Linda Weiser Friedman
//date compiled: October 1999.

#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
using namespace std;
                                               //function prototypes
void PrintSummaryInfo(int, float, float, float);
void PrintReportHeadings(string, string, float, float, float, float);
void PrintColumnHeadings();
void PrintStudentLine(string, string, string, int, int, int, int, float);
void PrintaLine(int, char);
void SkipLines (int);
void indent (int);
float FindMax(float);
float FindMin(float);

ifstream infile ("a:students.dat");            //input data file
ofstream outReport ("a:report.txt");           //output text file - Report

int main (){
                                               //variable declarations
   string id, course, semester, LastName, FirstName;
   float weight1, weight2, weight3, weight4;
   int n = 0, grade1, grade2, grade3, grade4;
   float FinalGrade, SumGrades = 0, HiGrade, LoGrade;

   if (!infile)                                //testing files
   cerr << "Error: could not open input file\n";
   else if (!outReport)
   cerr << "Error: could not open output file\n";
   else {                                      //files OK - //do rest of program

   infile >> course >> semester >> weight1 >> weight2 >> weight3 >> weight4;
   PrintReportHeadings(course, semester, weight1, weight2, weight3, weight4);

   while (infile >> id >> LastName >> FirstName >> grade1 >> grade2 >> grade3
       >> grade4){
       FinalGrade =  weight1*grade1 + weight2*grade2 + weight3*grade3 +
          weight4*grade4;
       PrintStudentLine(id, LastName, FirstName, grade1, grade2, grade3, grade4,
          FinalGrade);
       SumGrades += FinalGrade;
       n++;
       HiGrade = FindMax(FinalGrade);
       LoGrade = FindMin(FinalGrade);
   } //end while

   PrintSummaryInfo(n, SumGrades, HiGrade, LoGrade);
   cout << "Close console window?  "; char c; cin>>c;
   return 0;
   }//end if/else from testing files
} //end main

float FindMax(float FinalGrade){
   static float MaxSoFar = -1;
   if (FinalGrade > MaxSoFar)
      MaxSoFar = FinalGrade;
   return MaxSoFar;
}

float FindMin(float FinalGrade){
   static float MinSoFar = 150;
   if (FinalGrade < MinSoFar)
      MinSoFar = FinalGrade;
   return MinSoFar;
}
                                              //functions for
                                              // printing the report
void PrintStudentLine (string id, string lName, string fName, int g1, int g2,
   int g3, int g4, float FinalGrade)
{
   outReport << setiosflags (ios::fixed) << setprecision(2)
      << setw (11) << setiosflags (ios::left) << id
      << setw(10) << fName << setw(10) << lName
      << setiosflags(ios::right)
      << setw (5) << g1
      << setw (5)<< g2 << setw (5)<< g3
      << setw (5)<< g4 << setw (10) << FinalGrade << endl;
}

void PrintReportHeadings(string course, string semester, float w1, float w2,
   float w3, float w4)
{
   PrintaLine (65, ':');
   indent(19);
   outReport << "Little School of Soft Knocks\n";
   indent (23);
   outReport << "Student Grade Report" << endl;
   PrintaLine (65, ':');
   outReport << "\nSemester:  " << semester << "\nCourse:  " << course <<endl;
   PrintaLine (65, '.');
   outReport << "\nIn computing the Final Grade for each student, the
      following\n"
      <<"weights were used:\n"
      << setw(10) << "Grade 1" << setw(6) << w1*100 << "%\t"
      << setw(10) << "Grade 2" << setw(6) << w2*100 << "%\n"
      << setw(10) << "Grade 3" << setw(6) << w3*100 << "%\t"
      << setw(10) << "Grade 4" << setw(6) << w4*100 << "%\n";
   SkipLines(1);
   PrintaLine (65, '.');
   PrintColumnHeadings();
   PrintaLine (65, '.');
}

void PrintColumnHeadings(){
   outReport << setiosflags(ios::right) << setw (10) << "Student ID"
      << setw(10) <<"Name" << setw(10) << ' '
      << setw(20) << "Grades 1 - 4" << setw(14) <<"Final Grade" << endl;
}

void PrintSummaryInfo(int n, float sum, float max, float min)
```

```
{
   SkipLines (2);
   PrintaLine (65, ':');
   SkipLines (1);
   outReport << setiosflags(ios::showpoint | ios::fixed | ios::left)
      << setw(25) << "number of students = " << n << endl
      << setw(25) << "Class Average = " << sum/n << endl
      << setw(25) << "Maximum Grade is " << max << endl
      << setw(25) << "Minimum Grade is " << min << endl;
   PrintaLine (65, ':');
}

void PrintaLine(int n, char c){
for (int i=1; i<=n; i++)
   outReport << c;
outReport << endl;
}

void SkipLines (int n){
for (int i=1; i<=n; i++)
   outReport << endl;
}

void indent (int n){
for (int i=1; i<=n; i++)
   outReport << ' ';
}
```

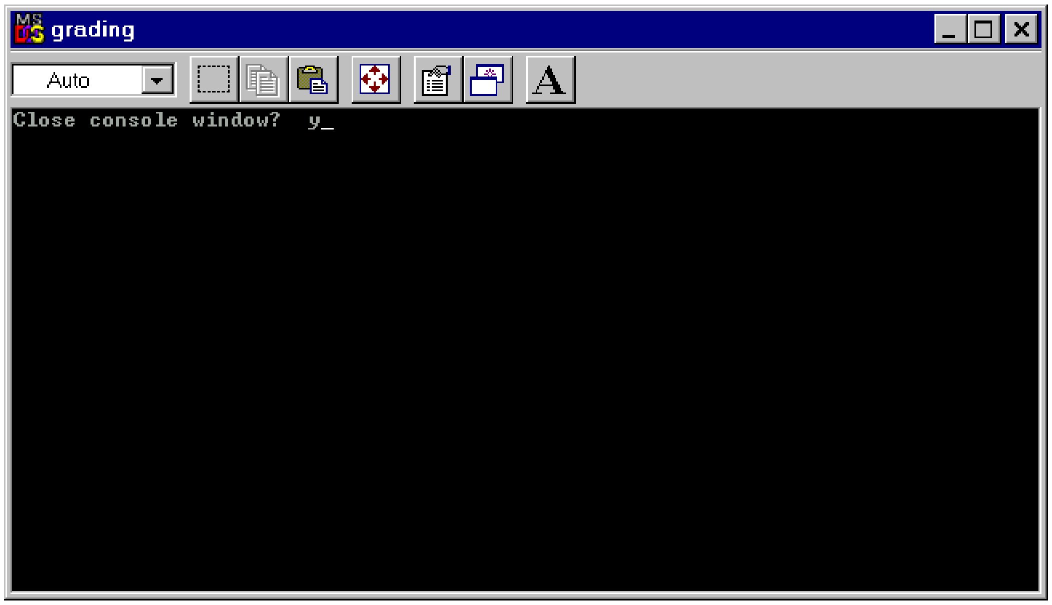
students.dat - Data file used for input data:

```
Object-Oriented_Programming
Fall1999
0.15 0.25 0.25 0.35
123456789 Archer Lew 99 62 101 89
111111111 Bond James 100 98 99 89
101010101 Burke Amos 65 77 88 98
222222222 Chambers Pat 44 84 88 101
999999999 Chambers Diane 70 32 90 95
333333333 Clousseau Inspector 42 65 85 54
444444444 Ed Mister 88 99 77 99
666666666 Hammer Mike 88 87 98 78
111222333 Hope Matthew 89 90 80 87
777777777 Kent Clark 99 99 98 99
555555555 Kramer Cosmo 98 87 76 65
000111222 Marlowe Philip 78 76 65 67
888888888 Rockford James 89 78 87 89
444555666 Sunnydale Buffy 87 32 78 92
777888999 Wolfe Nero 100 100 99 100
```

Printout of report.txt:

```
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                   Little School of Soft Knocks
                       Student Grade Report
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

Semester:  Fall1999
Course:  Object-Oriented_Programming
.................................................................

In computing the Final Grade for each student, the following
weights were used:
   Grade 1    15%        Grade 2    25%
   Grade 3    25%        Grade 4    35%

.................................................................
Student ID      Name                  Grades 1 - 4   Final Grade
.................................................................
123456789  Lew       Archer       99   62  101   89      86.75
111111111  James     Bond        100   98   99   89      95.40
101010101  Amos      Burke        65   77   88   98      85.30
222222222  Pat       Chambers     44   84   88  101      84.95
999999999  Diane     Chambers     70   32   90   95      74.25
333333333  Inspector Clousseau    42   65   85   54      62.70
444444444  Mister    Ed           88   99   77   99      91.85
666666666  Mike      Hammer       88   87   98   78      86.75
111222333  Matthew   Hope         89   90   80   87      86.30
777777777  Clark     Kent         99   99   98   99      98.75
555555555  Cosmo     Kramer       98   87   76   65      78.20
000111222  Philip    Marlowe      78   76   65   67      70.40
888888888  James     Rockford     89   78   87   89      85.75
444555666  Buffy     Sunnydale    87   32   78   92      72.75
777888999  Nero      Wolfe       100  100   99  100      99.75


:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

number of students =     15
Class Average =          83.99
Maximum Grade is         99.75
Minimum Grade is         62.70
:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

Console window (no error messages):



```
Close console window?  y_
```

students.dat - Data file used for input data:

```
Object-Oriented_Programming
Fall1999
0.15 0.25 0.25 0.35
123456789 Archer Lew 99 62 101 89
111111111 Bond James 100 98 99 89
101010101 Burke Amos 65 77 88 98
222222222 Chambers Pat 44 84 88 101
999999999 Chambers Diane 70 32 90 95
333333333 Clousseau Inspector 42 65 85 54
444444444 Ed Mister 88 99 77 99
666666666 Hammer Mike 88 87 98 78
111222333 Hope Matthew 89 90 80 87
777777777 Kent Clark 99 99 98 99
555555555 Kramer Cosmo 98 87 76 65
000111222 Marlowe Philip 78 76 65 67
888888888 Rockford James 89 78 87 89
444555666 Sunnydale Buffy 87 32 78 92
777888999 Wolfe Nero 100 100 99 100
```