

How the Grader Scores Your Submission

A sample e-mail message from the Grader Server is shown on page 10.

- **Submission Information:** The message tells you which submission this scoring was for and how many submissions are allowed for this project.
- **Point Information:** Provides your score, with a breakdown indicating any bonus or penalty points assessed. Your Instructor will determine how bonus and/or penalty points are to be assigned.
- **Input File:** Shows the input file on which this submission was tested.
- **Correct Output File:** The output file produced by the Instructor's solution using the preceding input file.
- **Student Output File:** The output file your submission produced from the same input file.

The Grader compares your output file to the correct output file, line by line. Each line is broken into "tokens" consisting of characters separated by whitespace (blanks or tabs). The tokens from each line of your output file are compared to those from the corresponding line of the correct output file; your score for each line is determined by the number of correct tokens you produce and the number of points assigned to that line. The point value assigned for each line is indicated by the number in square brackets at the beginning of each line of the correct output file.

Note that the amount of horizontal space you put between tokens does not matter to the Grader, unless you manage to run two tokens together so the Grader treats them as a single token (for example, printing "GrossPay" instead of "Gross Pay").

It is important that your output file not contain any extra lines, or omit any lines. If you do have extra lines or missing lines, then the Grader may compare the wrong lines and you will receive a very low score. For example, if Joe Bob Hokie had omitted the line of equal signs following the line of column labels in the output file, every line after the column labels line would have received a score of zero!

Explanation of scoring of a "good try": So why did Joe Bob get 61 for this submission? Compare the correct output file with Joe Bob's output. It's pretty obvious that Joe Bob doesn't have the correct averages for the insurance, income tax or net pay columns. Comparing the numbers in those columns with the correct answers, you find that Joe Bob has 4 of the insurance amounts wrong, and that all 6 of the income tax amounts are wrong. So, of course, all 6 of the net pay amounts are also wrong. Now each line in the body of the table is worth 10 points, so each amount (token) is worth $10/6$ points. Joe Bob also has three of the averages wrong, and each of those is worth $24/6$ points. If you add it all up, the Grader deducted $16*(10/6) + 3*(24/6)$ points. That works out to a deduction of about 38.67 points for a score of 61.3 — the Grader rounds the score to an integer, so Joe Bob got a 61 for this submission.

It is also vital that your labels match those used in the correct output file exactly. Otherwise those tokens will not match and the Grader will deduct points. Joe Bob modified the program to fix the calculations that were wrong before and resubmitted. Page 11 shows the Grader's response for the revised submission.

The reason Joe Bob received an 87 instead of 100 on this submission is that some of the labels are wrong now. The title on the second line is wrong, costing 2 points (Payroll is compared to March, so it's wrong, and he has one extra token). He also got some of the column headers wrong. Each token in that line is worth $12/8 = 1.5$ points. Joe Bob got careless and omitted the word Pay, which threw the token comparison out of order. Each of Joe Bob's remaining tokens failed the comparison, making 5 wrong tokens; Joe Bob also had two fewer tokens than the correct line and that was counted as two more wrong tokens. So the column headers cost Joe Bob a total of 10.5 points. So his score was 87.5, rounded to 88 by the Grader.

Extra or missing lines: Notice that some of the lines are given a value of 0 points. For instance, the first line is supposed to contain the name of the programmer. "Joe Bob Hokie" doesn't match the line in the correct output, but that's OK since the point value of the line is 0. That means that you don't have to match anything on that line exactly; but remember, if you don't have a line there it will throw the comparison off again.

Also, notice the horizontal lines that separate sections of the e-mail message. The line at the very end, after the student output section may help you determine your mistakes, in some cases. If your output file has extra nonblank lines, when compared to the correct output file then you will be penalized for each of those; the going rate for extra nonblank lines is 10 points. Compare the line counts given in the email message to see if you have too many or too few lines of output.

On the other hand, if your output has fewer lines than the correct output, then you're penalized the number of points that are assigned to those lines. For example, if Joe Bob had failed to print the line of averages after the body of the table, the Grader would have deducted 24 points.

How extra or missing blank lines are handled depends on where they occur. Extra blank lines at the end of your output file should be ignored by the Grader. If you insert an extra blank line in the middle of your output file, the Grader will assess a penalty for that, determined by your instructor, and then attempt to resynchronize its comparison by reading the next line of your output. If you omit a specified blank line, that is handled in a similar manner. Note that the email header specifies how many mismatches there were involving blank lines and the total penalty assessed for them.

Other messages, other problems: The two example Grader e-mail messages discussed above are the most common sort. However, there are several other scenarios that you should be aware of. If your submission does not compile (without errors) you will receive the following message:

```
Your source code failed to compile.  
Please correct your code so that it compiles and then submit it again.  
Your score for this submission is ZERO.
```

Of course, you should never submit a program that does not compile since all that does is waste a submission. Note two possible causes of this are using a different compiler (the Grader uses Visual C++ 5.0) or submitting the wrong file. There have even been instances where students have submitted term papers to the Grader – you shouldn't be surprised that a term paper usually doesn't compile properly.

The Grader gives your program a fixed amount of time to finish (the default is 10 seconds). If your program has not finished within the time limit, the Grader kills your program and you'll receive the message:

```
Your source code failed to exit and took too long to execute.  
Possible problems include:  
- an infinite loop  
- a runtime error such as an array index out of bounds  
  or divide by zero  
- your program expects input from the keyboard  
Please correct your code so that it terminates properly.  
Your score for this submission is ZERO.
```

When this happens, you need to do further testing and fix the problem before resubmitting. There is absolutely no point in resubmitting the same source code. This sort of problem may occur for a variety of reasons, including but not limited to the ones listed in the Grader message. The message will also include the input file and correct output, so you may use that input file to try to determine why your program misbehaved.

It is also possible that your program may be terminated abnormally (killed) by the operating system on the Grader machine before the time limit has expired. In that case, the Grader doesn't actually "know" that happened. It will look for an output file to score as usual and you will receive an e-mail message similar to the first one discussed above. However, if this happens you may notice that your output is incomplete since your program did not run to a normal termination. If that happens, you need to debug your program to eliminate the error before using another submission.

You may also find that when you run your program on your computer with the input file the Grader used, your program produces a complete output file and appears to operate correctly. If the Grader indicates your program is not producing correct results, but it does something different on your computer, the most likely explanation is that you are testing your program under Windows 95. The Grader operates on a machine using Windows NT, which is much less forgiving of misbehaving programs than Windows 95 is. You may not be able to discover the source of the problem and fix it under Windows 95. In that case, you should test your program in the Computer Science Undergraduate Lab on one of the computers equipped with Windows NT. Understand: the fact that your program appears to run correctly under Windows 95 but not under Windows NT indicates only that Windows 95 tolerates bad behavior, not that your program is correct.

You may also get an e-mail message similar to:

```
Your source code failed to produce the output file: outpay.dat
Possible problems include:
- you specified the wrong output file name
- you specified the wrong input file name
- your program had a runtime error and was terminated
  by the system
Please correct your code so that it produces this output file.
Your score for this submission is ZERO.
```

In this case, the Grader couldn't find your output file. The correct name for your output file will be given in the specification for your assignment. If you use another name, the Grader will not find your output. If your program is terminated by the system before it produces any output, you'll see the same message. Again, test and determine what the problem is, and fix it, before using another submission.

The moral of all this is simple: follow the project specifications for output precisely!

Date: Tue, 18 Aug 1998 14:27:47 -0400 (EDT)
From: grader@vt.edu
Subject: C++ Grader Notification

PLEASE DO NOT REPLY TO THIS EMAIL MESSAGE

This information is for submit number 2 of 3 allowed submits.

Point Information

Raw Score	:	61
Early Bonus	:	2
Late Penalty	:	0
Blank Line Penalty	:	0 (0 mismatches)

Total Score	:	63

Input File

1912	64	1957.00	B
2179	35	4282.00	B
6008	54	3324.00	B
9171	59	1026.00	D
5237	43	2318.00	D
5131	36	3621.00	B

Correct Output File: 13 lines

```
[ 0]Bill McQuain
[ 4]Macro$oft Corporation Payroll
[ 0]
[12] IdNum    Gross Pay      Ins.      F.I.T.      SSI      Net Pay
[ 0]=====
[10] 1912      1957.00      150.00      547.96      132.10      1126.94
[10] 2179      4282.00      100.00      1413.06      289.04      2479.91
[10] 6008      3324.00      150.00      1096.92      224.37      1852.71
[10] 9171      1026.00      200.00      287.28       69.26       469.46
[10] 5237      2318.00      200.00      649.04      156.47      1312.49
[10] 5131      3621.00      150.00      1194.93      244.42      2031.65
[ 0]=====
[24] Avg:      2754.67      158.33      864.86      185.94      1545.53
```

Student Output File: 13 lines

Joe Bob Hokie
Macro\$oft Corporation Payroll

IdNum	Gross Pay	Ins.	F.I.T.	SSI	Net Pay
=====					
1912	1957.00	175.00	489.25	132.10	1160.65
2179	4282.00	125.00	1498.70	289.04	2369.27
6008	3324.00	175.00	1163.40	224.37	1761.23
9171	1026.00	200.00	256.50	69.26	500.24
5237	2318.00	200.00	579.50	156.47	1382.04
5131	3621.00	175.00	1267.35	244.42	1934.23
=====					
Avg:	2754.67	175.00	875.78	185.94	1517.94

Date: Tue, 18 Aug 1998 14:35:49 -0400 (EDT)
From: grader@vt.edu
Subject: C++ Grader Notification

PLEASE DO NOT REPLY TO THIS EMAIL MESSAGE

This information is for submit number 3 of 5 allowed submits.

Point Information

Raw Score	:	88
Early Bonus	:	2
Late Penalty	:	0
Blank Line Penalty	:	0 (0 mismatches)

Total Score	:	90

Input File

5300	23	602.00	D
3319	47	2359.00	B
9760	62	986.00	B
3876	68	2146.00	D
6067	23	3481.00	D
1607	19	4953.00	B
1980	44	2194.00	D
8079	52	775.00	D

Correct Output File: 15 lines

```
[ 0]Bill McQuain
[ 4]Macro$oft Corporation Payroll
[ 0]
[12] IdNum    Gross Pay      Ins.      F.I.T.      SSI      Net Pay
[ 0]=====
[ 7] 5300      602.00      140.00      90.30      40.64      331.07
[ 7] 3319      2359.00      150.00      660.52      159.23      1389.25
[ 7] 9760       986.00      150.00      147.90      66.56      621.54
[ 7] 3876      2146.00      300.00      600.88      144.86      1100.27
[ 7] 6067      3481.00      140.00      1148.73      234.97      1957.30
[ 7] 1607      4953.00      100.00      1634.49      334.33      2884.18
[ 7] 1980      2194.00      200.00      614.32      148.10      1231.58
[ 7] 8079       775.00      200.00      116.25      52.31      406.44
[ 0]=====
[28] Avg:      2187.00      172.50      626.67      147.62      1240.20
```

Student Output File: 15 lines

Joe Bob Hokie
Macro\$oft Corporation March Payroll

IdNum	Gross	Ins	FIT	SSI	Net
=====					
5300	602.00	140.00	90.30	40.64	331.07
3319	2359.00	150.00	660.52	159.23	1389.25
9760	986.00	150.00	147.90	66.56	621.54
3876	2146.00	300.00	600.88	144.86	1100.27
6067	3481.00	140.00	1148.73	234.97	1957.30
1607	4953.00	100.00	1634.49	334.33	2884.18
1980	2194.00	200.00	614.32	148.10	1231.58
8079	775.00	200.00	116.25	52.31	406.44
=====					
Avg:	2187.00	172.50	626.67	147.62	1240.20
