# MKEVALS

## A program to draft Narrative Evaluations

Peter Scott

The **mkevals** program will produce a draft of narrative evaluations, in a format suitable for submission to the UCSC registrar (via script@ucsc.edu). Evaluations are based on exam scores (or quiz scores) and homework scores. The program takes as input a lightly edited class roster file.

Although **mkevals** was originally designed to run on UNIX platforms (including Linux), there is now a version that will run on Windows. It has been tested on Windows XP, but may also work on other Windows platforms. For installation instructions, on either Windows or UNIX, click on this link

The source code for **mkevals**, for either UNIX or Windows, is available here.

# 1 Overview

**mkevals** produces a draft of written narrative evaluations by analyzing numerical scores on a number of *items* such as examinations, quizzes or homework. The program takes as input the file '`infile`', which is the class roster supplied by the UCSC registrar's office (currently available from the "myUCSC" website), suitably edited as described below. If the **nroff** (or **groff**) command exists on your system, the program produces the file '`infile.roff`', which is a plain text file ready for submission to the registrar's office (via script@ucsc.edu) following optional minor editing. If the **nroff** command is not found by the program, the raw file '`infile.evs`' is produced, ready to be processed by **nroff** or **groff**. The program also suggests grades (P or NP or letter grades), based on item scores. These appear in the file '`infile.grd`'.

In addition, a histogram of the grade distribution appears on the terminal screen.

For a quick test after compiling the program, try giving the command

**mkevals** '`sample.new`'

where the file '`sample.new`' is the file supplied by the distribution.

Designed to be appropriate for physics courses, the program may be useful for courses in other disciplines, particularly those courses for which evaluations are based primarily on scores achieved on exams or homework.

# 2 Options

[[NOTE: long options will be available only if you compile the program with a GNU C compiler.]]

‘**-h** `or` **--help**’
: Prints a short summary on the screen and exits.

‘**-v** `or` **--version**’
: Prints the version number on the screen and exits.

‘**-o** `or` **--old**’
: Accommodates old style (before Summer, 2005) class rosters in the input file ‘`infile`’. The new style rosters now contain an additional comma that separates the first and second names of each student listed.

‘**-p** `or` **--plus**’
: This option allows the use of *plus* and *minus* grades in addition to the usual *P*, *NP* and straight letter grades, so that the full grade spectrum (A+, A, A-, B+, B, B-, C+, C, D, F) may be used if it is desired.

‘**-r** `or` **--rank**’
: Invoking this option will produce the file ‘`infile.rank`’, which shows the ranking of the class according to ten categories (9 to 0) corresponding to the following grades: A+=9, A=8, A-=7, B+=6, B=5, B-=4, C+=3, C=2, D=1, F=NP=W=0. The grade P is translated to a letter grade for this purpose. (Some students included in the class list supplied by the registrar may have withdrawn from the course (the notation `"ADM"` will appear in the student’s header line); these are shown with a grade ‘`W`’. Such students do not need to be deleted from the class list.)

‘**-u** `or` **--upgrade**’
: If the ‘`upgrade`’ option is supplied, performance on a midterm or quiz will be discounted if a student’s final exam score shows significant improvement over a low (or non-existent) midterm score.

  The way it works is that if it is found that a midterm score falls in a category less than that of the final exam, it is given zero weight and the remaining weights are renormalized, the total score `"breakpoint"` is recalculated, and the total score and category are redetermined. If this procedure results in an increase of the total score category, the evaluation of the overall performance is rewritten, and a phrase noting improved performance on the final exam is added. Evaluations of individual exams (including discounted ones) are left unchanged. I invariably use this option, and explain to students that significantly improved performance on the final will be acknowledged.

‘**-d** `or` **--debug**’
: Use of the debug option will cause a rather detailed display on the terminal screen that shows category breakpoints and item categories (see below) for each student, including any changes to these that might occur if the upgrade option is also given. This can be useful for analyzing borderline cases, or for just gaining a clearer picture for how the program works.

# 3 How to use the Program

The program requires as input the file currently provided by the UCSC registrar's office, which must be edited as described in the following paragraphs. As of this writing, this file may be obtained by logging in at http://my.ucsc.edu/, clicking on *Manage My Classes*, then on *My Class Rosters*, then selecting the class. Then click on the *Savable Eval Data* button, (NOT the *Savable Roster* button). Then when the file appears in your browser window, use your browser's *Save As* button to write the relevant ASCII text file to any appropriate directory on your computer. Below is a (bogus) sample of seven lines from such a file. This sample file is in the format scheduled for use starting Fall Quarter, 2005, containing a new comma that separates the first and second names of each student:

```
-----------------------------------------
@@@@,PHYS 7B 01       ,51986,2032,0242172
&&&&,Adams,Aurora,Renee            ,0161978,F,CROWN,GRD,ara@ucsc.edu
&&&&,Bosworth,Michael,David        ,0217784,M,KRESGE,PNP,mdb@ucsc.edu
&&&&,Dellinger,Kristy,Marie        ,0232265,F,NINE,ADM,kmd@ucsc.edu
&&&&,Kilroy,Caleb,Manning Star     ,0196734,M,OAKES,GRD,cmk@ucsc.edu
&&&&,Mc Donald,Charlotte,Marie     ,0199823,F,OAKES,PNP,cmm@ucsc.edu
&&&&,O'Brien,Siobhan,              ,0223046,F,KRESGE,GRD,sob@ucsc.edu
-----------------------------------------
```

The first line of such a file contains information about the course and instructor. Subsequent lines, which we refer to as *header* lines, contain information about each student enrolled in the course.

To use **mkevals**, two steps are necessary in modifying the file you have obtained:

(A) — Using a text editor, the student's numerical scores for each item must be inserted after each header line. For example, suppose there are four items to be evaluated, say two midterms, a final exam, and weekly homework. Suppose Ms. Adams scored 68 and 75.5 on the first and second midterms, 96 on the final, and 272 on the weekly homework. These numbers would be inserted after this student's header line:

```
-----------------------------------------
&&&&,Adams,Aurora,Renee            ,0161978,F,CROWN,GRD,ara@ucsc.edu
68    75.5   96    272
-----------------------------------------
```

These item scores may be in any order (except midterm scores must be consecutive and in chronological order), but must be in the *same* order for each student. Item scores that do not exist (for example, a missed exam) should be entered as 0. Item scores should be separated by white space (spaces, tabs, or a newline), not by commas or other characters. The above file would therefore be modified to look something like this:

```
-----------------------------------------
@@@@,PHYS 7B 01       ,51986,2032,0242172
&&&&,Adams,Aurora,Renee            ,0161978,F,CROWN,GRD,ara@ucsc.edu
68    75.5 96    272
&&&&,Bosworth,Michael,David        ,0217784,M,KRESGE,GRD,mdb@ucsc.edu
43    60   81    230
&&&&,Dellinger,Kristy,Marie        ,0232265,F,NINE,GRD,kmd@ucsc.edu
```

```
57    30    0    100
&&&&,Kilroy,Caleb,Manning Star     ,0196734,M,OAKES,ADM,cmk@ucsc.edu
57    10    0     90
&&&&,Mc Donald,Charlotte,Marie     ,0199823,F,OAKES,PNP,cmm@ucsc.edu
55    0    61    170
&&&&,O'Brien,Siobhan,              ,0223046,F,KRESGE,GRD,sob@ucsc.edu
89    86   90.5  280
-----------------------------------------
```

(B) — A table must be placed at the beginning of the input file, either before or just after the course description line. This table must contain a row for each item (exam, quiz or homework), and in that row, there must be, in order, a *keyword*, nine *breakpoints*, and a *weight factor*. (Preparing this table is the chief task for the user of this program.)

Since each row (*itemline*) of this table refers to an item, four items (two midterms, a final and homework) would require four rows in the table. The itemlines must be in the *same* order as the scores assigned to each student.

The *keyword* describes the *item*. Legal keywords are *midterm*, *final*, *homework* and *quizzes*, or actually any word whose first character is *m*, *f*, *h* or *q*. No space characters may exist before these words: Do not indent these lines. The keywords are case-insensitive: *M*, *F*, *H* or *Q* would work equally well.

The nine *breakpoints* (bp[0] through bp[8]) must be chosen by the user so as to divide the item score range into ten *categories* (cat[0] through cat[9]), currently described in words like this:

```
category      phrase        grade             range
-------------------------------------------------------------
   0        "very weak"    F or NP             cat[0] < bp[0]
   1          "weak"       D or NP    bp[0] <= cat[1] < bp[1]
   2        "adequate"     C or P     bp[1] <= cat[2] < bp[2]
   3        "adequate"      (C+)      bp[2] <= cat[3] < bp[3]
   4          "good"        (B-)      bp[3] <= cat[4] < bp[4]
   5          "good"         B        bp[4] <= cat[5] < bp[5]
   6        "very good"     (B+)      bp[5] <= cat[6] < bp[6]
   7        "excellent"     (A-)      bp[6] <= cat[7] < bp[7]
   8        "excellent"      A        bp[7] <= cat[8] < bp[8]
   9       "outstanding"    (A+)      bp[8] <= cat[9]
-------------------------------------------------------------
```

Breakpoints are score values that separate the categories. A score falling precisely on a breakpoint is assigned to the higher category. Normally it might make sense to arrange the breakpoints such that the class median falls somewhere toward the high end of category 5 (B) in the above table, although of course this is not required. Consecutive breakpoints may be equal to each other so as to define the ranking more coarsely. (This eliminates a category for the item to which such breakpoints refer.) Otherwise breakpoints must be in increasing order, if they are not, the program will complain.

Each item must also be assigned a *weight factor*, also chosen by the user. Suppose we evaluate, as in the example above, performance based on the two midterms, the final exam and the homework, and want the final to have twice the weight of a midterm, and the

homework to have half the weight of a midterm. Thus we might think to assign a weight of 1.0 to each midterm, 2.0 to the final, and 0.5 to the homework. However in our case these weight factors would be *incorrect*, because while the range of scores for each exam (including the final) is 0 to 100, that for the homework is 0 to 300, or 3 times the range of exam scores. Thus, to make the homework count as half the weight of a midterm, we should give it a weight factor of $0.5/3.0 = 0.167$.

We do not need to normalize the weight factors (that is, to arrange that they sum to 1.0), since the program does it for us.

With this in mind, an appropriate table of breakpoints and weights might appear like this, put at the start of our sample input file, so the completed sample input file (to which we give the name *sample*), will look like this:

```
----------------------------------------
 # any line in this file that contains a "#" at any position is
 # judged a comment line and is ignored.
 # scores falling on a breakpoint are assigned to upper category.
 # categ: 0   1    2    3    4    5    6    7    8    9
 #
 # grade: F   D    C    C+   B-   B    B+   A-   A    A+
 #
 midterm   55   60   65   68   75   75   83   83   90   1.0
 midterm   55   60   65   65   75   75   83   83   90   1.0
 final     40   55   60   63   70   74   82   85   93   2.0
 homework  90  130  180  180  210  210  240  240  280   0.167
 @@@@,PHYS 7B 01       ,51986,2032,0242172
 &&&&,Adams,Aurora,Renee            ,0161978,F,CROWN,GRD,ara@ucsc.edu
 68    75.5 96    272
 &&&&,Bosworth,Michael,David        ,0217784,M,KRESGE,GRD,mdb@ucsc.edu
 43    60   81    230
 &&&&,Dellinger,Kristy,Marie        ,0232265,F,NINE,GRD,kmd@ucsc.edu
 57    30    0    100
 &&&&,Kilroy,Caleb,Manning Star     ,0196734,M,OAKES,ADM,cmk@ucsc.edu
 57    10    0     90
 # This is a comment. (Kilroy missed taking the final.)
 &&&&,Mc Donald,Charlotte,Marie     ,0199823,F,OAKES,PNP,cmm@ucsc.edu
 55    0    61    170
 This is also a comment.#  (Mc Donald didn't take the second midterm.)
 &&&&,O'Brien,Siobhan,              ,0223046,F,KRESGE,GRD,sob@ucsc.edu
 89    86   90.5  280
----------------------------------------
```

Note that comment lines (those containing a "#") may be inserted anywhere in the file. Such lines are ignored by the program.

The program will complain if an itemline does not contain eleven character strings (entries), or if the breakpoints are not in ascending order.

The above sample input file is now complete, and ready to be processed by the **mkevals** program.

In the simplest case (no options) simply give the command

**mkevals** ‘`sample`’

The program will calculate the breakpoints for the (weighted) total score, and from the weighted total score for each student, writes an appropriate statement describing his or her overall performance in the course, along with appropriate statements regarding performance on individual items.

The program also suggests a grade for each student, based on the student's total score category. This grade will be a letter grade if the student has chosen that option (the notation "GRD" will appear in the student's header line). Otherwise the grade will be *P* or *NP*. (Students listed as withdrawn will be assigned a grade of *W*, and the simple statement "Mr. ____ withdrew from the course" will be written in the evaluation.)

[Note that for this simplest case, "+" and "-" grades will not be assigned. See below for how to use the "‘-p’" (or "‘--plus’") option to include "+" and "-" grades.]

The above command should produce this on the screen:

```
-----------------------------------------

 Good.  It worked.

 You have drafted 6 evaluations.
 Here is the distribution for the
 suggested assigned grades:

  A:(2) ##
  B:(1) #
  C:(0)
  D:(0)
  F:(1) #
  P:(0)
 NP:(1) #
  W:(1) #

 A list of suggested grades will be found in
 the file "sample.grd".

 The completed evaluations will be found in
 the file "sample.roff", ready for any editing
 before sending to the registrar at script@ucsc.edu.

-----------------------------------------
```

The file ‘`sample.grd`’ should look like this:

```
-----------------------------------------
 Adams,Aurora Renee          A
 Bosworth,Michael David      B
 Dellinger,Kristy Marie      F
 Kilroy,Caleb Manning Star   W
```

```
 Mc Donald,Charlotte Marie   NP
 O'Brien,Siobhan             A
-----------------------------------------
```

and the file '`sample.roff`' should look like this:

```
-----------------------------------------
 @@@@,PHYS 7B 01       ,51986,2032,0242172

 &&&&,Adams,Aurora,Renee            ,0161978,F,CROWN,GRD,ara@ucsc.edu
 Ms. Adams's overall performance was excellent.  She demonstrated a
 thorough grasp of the basic concepts, and was able to apply the
 concepts to solve often quite difficult problems.

 On the basis of numerical scores, her performances on individual exams
 were as follows:

 &&
 First midterm:       Good, near the class median.
 Second midterm:      Very good, well above the class median score.
 Comprehensive final: Outstanding, among the very top scores.
 &&

 Her performance on the weekly homework was particularly good.

 &&&&,Bosworth,Michael,David        ,0217784,M,KRESGE,GRD,mdb@ucsc.edu
 Mr. Bosworth's overall performance was good.  He demonstrated a good
 grasp of the basic concepts, and an ability to apply the concepts to
 solve non-trivial problems.

 On the basis of numerical scores, his performances on individual exams
 were as follows:

 &&
 First midterm:       Very weak.
 Second midterm:      Adequate.
 Comprehensive final: Very good, well above the class median score.
 &&

 &&&&,Dellinger,Kristy,Marie        ,0232265,F,NINE,GRD,kmd@ucsc.edu
 Ms. Dellinger did not take the final exam.  Thus her overall
 performance is not sufficient to warrant a passing grade for the
 course.

 &&&&,Kilroy,Caleb,Manning Star     ,0196734,M,OAKES,ADM,cmk@ucsc.edu
 Mr. Kilroy withdrew from the course.

 &&&&,Mc Donald,Charlotte,Marie     ,0199823,F,OAKES,PNP,cmm@ucsc.edu
```

```
Ms. Mc Donald's overall performance, while showing understanding of a
few topics, was not sufficiently strong to warrant a passing grade for
the course.

On the basis of numerical scores, her performances on individual exams
were as follows:

&&
First midterm:       Weak.
Second midterm:      Apparently did not take this exam.
Comprehensive final: Adequate.
&&

&&&&,O'Brien,Siobhan,              ,0223046,F,KRESGE,GRD,sob@ucsc.edu
Ms. O'Brien's overall performance was excellent.  She demonstrated a
thorough grasp of the basic concepts, and was able to apply the
concepts to solve often quite difficult problems.

On the basis of numerical scores, her performances on individual exams
were as follows:

&&
First midterm:       Excellent, among the best in the class.
Second midterm:      Excellent, among the best in the class.
Comprehensive final: Excellent, among the best in the class.
&&

Her performance on the weekly homework was particularly good.
-----------------------------------------
```

The above file may be edited to do any desired fine-tuning before sending to the registrar's office (*i.e.*, to *script*`@`*ucsc.edu*).

The program as currently written assumes that evaluations will be based on (a) exactly one final exam score, (b) any number of midterm exams (including zero) *or* (c) a possible set of quizzes (some instructors give short weekly quizzes instead of midterm exams), and (d) a possible set of homework assignments. The program will not work correctly if both midterms *and* quizzes are present.

Note that as written, a statement commending the quality of the student's homework is added to the evaluation (of passing students) only if the total homework score is in one of the top three categories; otherwise nothing is written on the evaluation. If more detailed analysis of homework is desired, the code at the end of '`write_eval.c`' may be modified and the program recompiled using the **make** command.

Use of various options cause varying types of output; see the descriptions above under *OPTIONS*. For example, try giving the command

**mkevals -purd** '`sample`'

which will produce the most complex output available, including `"+"` and `"-"` grades, an upgrading of students' later work in the course, a ranking of students by category, and a detailed look at the inner workings of the program.

# 4 How to Customize Evaluation Phrases

Phrases written into the evaluation may be altered to suit individual tastes. To change phrases, simply edit the program function files '`write_eval.c`' (for overall evaluations) and '`eval_item.c`' (for evaluations of individual items), and recompile the program using the **make** command. Even if you don't want to alter the phrases it might be good to look at these files so you can see how the phrases are produced, and how they might be altered.

As mentioned above, if a more detailed treatment of homework is desired, the code at the end of '`write_eval.c`' can be modified in a suitable manner and the program recompiled.

Don't forget that it is always possible to edit the output of this program (the file '`infile.roff`') to insert any specially chosen phrases for particular students.

# 5 About the Author

The program was written by me, following up on an early version long ago. Since that time the program has been revised or altered by various other members of the Physics faculty at UCSC. (Suggestions by Peter Young have been particularly helpful.) The current program is an attempt to make it easier to use, less error-prone and more adaptable. I welcome suggestions, comments, or bug reports, which can be sent to me at drip@ucsc.edu. Let me know also if you would like to be informed of future revisions or updates.

— Peter Scott