GRADER Manual

2007 June 25

Sunil Issar
Peter B. Andrews
Frank Pfenning
Dan Nesmith

This manual is based upon work supported by NSF grants MCS81-02870, DCR-8402532, CCR-8702699, and CCR-9002546, and a grant from Center for Design of Educational Computing, Carnegie Mellon University. Any opinions, findings, and conclusions or recommendations are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1. Introduction

GRADER is a computer program that can be used to maintain, compute, analyze, and print information about grades. It provides an efficient means of recording grades, and can save many precious end-of-semester hours by quickly and accurately calculating, ordering, displaying, and printing grades, means and other statistics. It can also assign letter grades in accordance with the numerical intervals selected by the teacher, and print these grades with the names of the students.

GRADER is easy to use, and generally prompts the user for required inputs. The easiest way to learn to use it is to just start and proceed step-by-step.

GRADER was designed to process the scores automatically generated by **ETPS**, a logic-teaching system at Carnegie-Mellon University. **GRADER** could also be used to process scores from other teaching programs which create records in a suitable format. The ability to incorporate records from teaching programs is one of the most attractive features of **GRADER**. The growing popularity of computer-assisted instruction will make grading packages with this feature more desirable, and perhaps indispensable. However, **GRADER** is also useful for courses which do not involve computers in any way.

GRADER is written in the Common Lisp computer language and within a stripped-down version of **ETPS**. It will thus accept not only **GRADER** commands, but also **LISP** expressions. However, you need not know anything about **LISP** to use **GRADER**.

If you have comments or suggestions concerning **GRADER** or this manual, please contact Professor Peter Andrews in the Mathematics Department of Carnegie Mellon University, or send email to Andrews@CMU.EDU.

In this manual, program variables will be printed in ***bold italics***. Commands will be printed in **BOLD CAPITALS**. Textual variables will be in *italics*, and explicit values in `typewriter face`. **GRADER** itself disregards the case of characters.


Wherever a **GRADER** command is described, the hand (as appears to the left) will point it out to you.

2. How it works

2.1. The Record

GRADER creates and maintains, for each course, a record analogous to an instructor's grade book, and processes and displays the recorded information. This record is kept in the file whose name is the value of the variable ***grade-file***.

☞**CREATE-GRADEFILE** creates the grade-file, which includes the following:

- The names of exercises
- The type of each exercise. The pre-defined types are `ET` for scores generated by a teaching program and `LE` for letter grades. The user may define additional types such as `HW` for homework, and `TEST` for tests.
- The weights assigned to each exercise when totals are computed. The default weight for each exercise is 1. The user can change it to any desired number by invoking the command **CHANGE-WEIGHT**.
- The maximum possible score, labelled `--MAX--`, for each exercise
- The scores achieved by each student
- Record of assignments submitted late
- Penalty functions and due-dates
- Actual names of exercises with aliases

Several commands, described in Chapter 5, let the user modify this record. One of these is the command **ETPS-GRADE**, which reads files containing records of work done by students using an educational computer program and updates grades accordingly. Each time one of these commands is completed, the file whose name is the value of ***grade-file*** is given the name which is the value of ***old-grade-file***. The new version is then saved in ***grade-file***.

The new version of the grade file not only incorporates changes from the commands in Chapter 5, but also re-computes for each altered exercise the statistical measures listed in ***statistical-options***. See Section 5.5 for details.

Totals for the current record are computed by **CALCULATE-GRADE**. By the same procedure as described above for ***grade-file*** and ***old-grade-file***, the old totals go to ***old-totals-grade-file***, while the new totals are put in ***totals-grade-file***.

3. Coming and Going

3.1. Getting into GRADER

GRADER is part of **TPS** and **ETPS**. Starting **GRADER** is similar to starting **TPS** or **ETPS** (see section 13.3 of the TPS User's Manual), but requires the additional command line argument `-grader`. For example, to start **GRADER** using Allegro Common Lisp 5.0 or greater, you can use the command

```
lisp -I tps3.dxl -- -grader &
```

or

```
lisp -I etps.dxl -- -grader &
```

where tps3.dxl and etps.dxl are the files that were created when **TPS** and **ETPS** were built.

However, Grader can be used by someone who knows nothing about **TPS** or **ETPS**. At Carnegie Mellon University, versions of **GRADER** exist on the Andrew system, and can be started by connecting to a Unix server or running on an Andrew Linux machine and entering the command `/afs/andrew/mcs/math/etps/bin/grader`. If you are at a different site, ask your local **ETPS** maintainer how to use **GRADER**. Before using **GRADER** for the first time, you may wish to create the files mentioned in sections 4.1 and 4.2.

3.2. GRADER Top-Level: HELP

The commands you issue to **GRADER** will be received by a part of the program called the top-level. The prompt for the top-level is `<Gr`*n*`>`, where *n* is an integer between 0 and 9.

You may abort any **GRADER** command by typing `^C`, i. e., typing `C` while pressing the control, `CTRL`, key. You will return to the **GRADER** top-level. **INSERT-GRADES** has its own method of resuming work. See Section 5.1 for how **INSERT-GRADES** operates.

A partially specified command can be completed by pressing the `<ESC>` key, then the `<RETURN>` key.

Possible completions of a partially completed command can be gotten by typing `?` before entering. Thus, a complete list of commands can be generated by typing `?`.

One other aid is provided:

☞ **HELP** *command* tells you what *command* does and what kind of arguments it takes. **HELP** *var* tells you about the variable *var* and what its current value is. If you are not sure of something, try asking for help on it.

3.3. Getting out: GR-EXIT and EXIT

There are two ways to leave **GRADER**:

☞ **GR-EXIT** takes you out of **GRADER**.

This command may be abbreviated as **EXIT**.

4. Setting Up

4.1. Variables: CHG-VARS and GR-REVIEW

Just after entering **GRADER**, you should make sure that the values of **GRADER** 's variables are appropriate. The default values set by **GRADER** are listed below. Note that all file names are strings, i.e., enclosed by double quotes.

course-name	`"course"`
grade-file	"***course-name***`.grades"`
totals-grade-file	"***course-name***`.totals"`
old-grade-File	"***course-name***`.xgrades"`
old-totals-grade-file	"***course-name***`.xtotals"`
letter-grade-file	"***course-name***`.letter"`
etps-file	`""`
statistical-options	`(-sdev- -median- -mean-)`
print-n-digits	`0`
cal-percentage	`NIL`
letter-grade-flag	`T`
new-item	`NIL`
drop-min	`NIL`
due-date-flag	`T`
default-penalty-fn	`NO-PENALTY`

These default values may be redefined by having a file in your directory called `grader.ini` which contains lines of the form (`setq` *variable 'new-default*). This is how variables are set in **LISP**, the native tongue of **GRADER**. **GRADER** will read this file when you start up.

EXAMPLE `(setq letter-grade-flag 'nil)` would make `NIL` the default for ***letter-grade-flag***.

A typical `grader.ini` file for a course called LOGIC would look like this:

```
(setq course-name "logic")
(setq default-penalty-fn 'penalty%10)
(setq cal-percentage T)
(setq statistical-options nil)
```

One thing to note is that setting the value of ***course-name*** in your `grader.ini` file will have the effect of changing the values of all the variables above whose values depend on that of ***course-name***.

Once inside **GRADER**, you may change or examine the values of these variables by using either of the following commands:

☞**CHG-VARS** displays a menu similar to the above list and asks you to select the variable you wish to modify.

☞**GR-REVIEW** puts you into a new top-level called **REVIEW**. The basic commands in **REVIEW** are:

- **SUBJECTS** lists the subjects into which the variables are classified.
- **LIST** gives the current settings of all the variables in a subject.
- **SETFLAG** lets you reset the value of a variable.
- **UPDATE** lets you reset the value of any or all variables in a subject.
- **LEAVE** leaves **REVIEW** and returns to **GRADER** with the new variable settings.

EXAMPLE A sample **REVIEW** session follows:

```
<Gr1>gr-review

<R1>subjects
GR-FILENAMES GR-MISC
<R2>update
SUBJECTLIST (SUBJECTLIST):  [No Default]>gr-misc
Subject: GR-MISC
CAL-PERCENTAGE [NIL]>
COURSE-NAME [COURSE]>"victoria"
DEFAULT-PENALTY-FN [NO-PENALTY]>
DROP-MIN [NIL]>
DUE-DATE-FLAG [T]>
LETTER-GRADE-FLAG [T]>
NEW-ITEM [NIL]>
PRINT-N-DIGITS [0]>
STATISTICAL-OPTIONS [(-MEAN- -MEDIAN- -SDEV-)]>
<R3>leave

[Left REVIEW.]
<Gr2>
```

Again, changing the value of ***course-name*** inside **CHG-VARS** or **REVIEW** will cause changes in the names of the files which depend upon it.

Filenames may be specified by "*name*`.`*ext*", or simply "*name*". The quotation marks must be typed.

4.2. The Grade-File: CREATE-GRADEFILE

Once the variables are set to your liking, you will next need to create a ***grade-file*** for each course. This ***grade-file*** should be created before the course begins. Subsequent ***grade-files*** for the course are generated from this initial ***grade-file***.

To create this ***grade-file***, you must first create a file (with extension `.lst`) containing a list (enclosed in parentheses) of student-ids for each student in the course. The form of these student-ids depends on whether you intend to use the **ETPS-GRADE** command. If you do, **GRADER** will need to know the names the computer gives your students. Thus the student-ids, in this case, will be lists of the form:

(*student-name student-user-id*)

EXAMPLE The student file class.lst for our hypothetical course on the Victorian Era might look like:

```
((ruskin-john jr03)
 (arnold-matthew ma24)
 (dickens-charles cd51)
 (gaskell-elizabeth eg21)
 (carlyle-thomas tc32)
 (disraeli-benjamin bd79))
```

If you do not intend to use the **ETPS-GRADE** command, the student-ids will just be the *student-names*.

EXAMPLE In this case the student file might look like:

```
(
ruskin-john
arnold-matthew
dickens-charles
gaskell-elizabeth
carlyle-thomas
disraeli-benjamin
```

)

Since **LISP** uses blanks to separate elements in a list, and commas and periods for more esoteric parsing, none of these characters may appear in any *student-name* or *student-user-id*. We recommend using hyphens between first and last names.

☞**CREATE-GRADEFILE** can now be used to create your ***grade-file***. Once this command is invoked, you will be prompted for the name of the class-list file you just formed. **CREATE-GRADEFILE** will order the students alphabetically by name and form a ***grade-file***, using that class list.

EXAMPLE Running **CREATE-GRADEFILE** on the student list in the previous example produced a ***grade-file*** with the following contents.

```
(
(NAME                ID)
(TYPE                **)
(--MAX--             **)
(-MEAN-              **)
(-MEDIAN-            **)
(-SDEV-              **)
(-WEIGHT-            **)
(ARNOLD-MATTHEW    MA24)
(CARLYLE-THOMAS    TC32)
(DICKENS-CHARLES   CD51)
(DISRAELI-BENJAMIN BD79)
(GASKELL-ELIZABETH EG21)
(RUSKIN-JOHN       JR03)
)
(;;;Record of late assignments:
()
;;;Record of penalty functions and due-dates:
()
;;;List of actual names
NIL)
```

4.3. Precautions and Protections

Given the risk of a computer malfunction, it is advisable to keep a recent, printed copy of these records. For this, you must use your system's print command on the ***totals-grade-file*** if you want the totals, or on the ***grade-file*** if you want the grades for each assignment.

GRADER writes over the old versions of its files when it creates new versions. Make copies of these files with new names if you wish to keep the old versions.

Don't forget to keep your grade files in a protected directory, where unauthorized persons cannot access them.

There is another danger to which **GRADER** is susceptible. Among its inputs are the names of students, exercises and exercise-types. These names become variables in **GRADER**. In addition, **GRADER** has certain program variables which subtly affect the way it works. Together with the **LISP** constants, `T` and `NIL`, these constitute the global objects of **GRADER**. That is, if any two of these have the same name, then a value given to one will be given to the other. Clearly, this is not an advisable circumstance and should be avoided.

5. Keeping Records

5.1. Manual Grades: INSERT-GRADES, RESUME-INSERT-GRADES, MODIFY-GRADE, LATE-EXERCISES

☞**INSERT-GRADES** must be used every time you manually enter grades for a new assignment. If you later wish to change the grades on that assignment, you should use **MODIFY-GRADE**.

INSERT-GRADES first prompts you for the names of the new exercises. Recall the warning in Section 4.3 about the names of exercises et al. Also, you should use short names to improve the legibility of your output. You can then use the **ALIASES** command to assign the real name to the short name. See Section 5.2 for details.

Next, the type for each exercise must be given. There are two pre-defined types, `LE` and `ET`. `ET` is the type for the computer-generated grades retrieved by **ETPS-GRADE**. `LE` is for letter-grades, the only type of grades which **CALCULATE-GRADE** (see Chapter 6) treats as symbols rather than numbers.

Finally, you are asked for the grades. For each student in the class, you will be prompted for that student's grades on the new exercises. These grades may be entered as any **LISP** expression. You will usually give expressions such as: `98, 4.5, B+, D-` or `*`. (Standing alone, * indicates that the exercise was not done; when * is the first element of a list, it is the multiplication operator, e.g. `(* 2 3)` means 6).

EXAMPLE

```
<Gr3>insert-grades

Please enter the names of the exercises, e.g.,HW1 EXAM1 LETTER1:
essay1
ESSAY1
Please enter 1 items specifying the type of each exercise:
hw
--MAX-- : 60
ARNOLD-MATTHEW : 57
CARLYLE-THOMAS : (* 9 6)
DICKENS-CHARLES : *
DISRAELI-BENJAMIN : (+ 4 (* 6 7))
GASKELL-ELIZABETH : (+ 25 23 1)
RUSKIN-JOHN : 52
```

Producing the following record in the ***grade-file***:

```
(
(NAME                ID ESSAY1)
(TYPE                **     HW)
(--MAX--             **     60)
(-MEAN-              **     52)
(-MEDIAN-            **     52)
(-SDEV-              **      4)
(-WEIGHT-            **      1)
(ARNOLD-MATTHEW    MA24     57)
(CARLYLE-THOMAS    TC32     54)
(DICKENS-CHARLES   CD51      *)
(DISRAELI-BENJAMIN BD79     46)
(GASKELL-ELIZABETH EG21     49)
(RUSKIN-JOHN       JR03     52)
)
(;;;Record of late assignments:
```

```
()
;;;Record of penalty functions and due-dates:
()
;;;List of actual names
NIL)
```

☞**RESUME-INSERT-GRADES** allows you to resume entering grades.

☞**MODIFY-GRADE** lets you enter or revise the grades on an old exercise. This command will first prompt you for the names of the students whose grades you want to modify. For each such student, you will be asked for the exercises involved. The old grades will be printed and you will be prompted for the revised grades.

EXAMPLE

```
<Gr4>modify-grade

Please type names of the students whose grades need to be
  modified:
dickens-charles gaskell-elizabeth
DICKENS-CHARLES
Please type the names of the grades that you want to change:
essay1
DICKENS-CHARLES's scores are: (ESSAY1 . *)
Please enter the new scores: score1 score2 ...:
53
GASKELL-ELIZABETH
Please type the names of the grades that you want to change:
essay1
GASKELL-ELIZABETH's scores are: (ESSAY1 . 49)
Please enter the new scores: score1 score2 ...:
51
```

☞**ALTER-GRADE** is a supplement to MODIFY-GRADE. It takes a list of grade names and prompts the user with the name of each student, allowing the grades in the list for each of those students to be changed.

☞**LATE-EXERCISES** lets the instructor record the number of days that a manually entered exercise is late. You will be prompted for the students, exercises and days of lateness. This information may be used later by **CALCULATE-GRADE**.

EXAMPLE

```
<Gr6>late-exercises

Please type the names of the students: dickens-charles

DICKENS-CHARLES
    Please enter the names of the exercises: essay1

Please specify days late:
ESSAY1: 3
```

5.2. Making the Output Convenient: ALIASES, COMMENT

When you are printing the files from **GRADER**, you may find that complete exercise names take up too much space. Therefore, it is convenient to refer to exercises by one- or two- letter names, which we refer to as short names. These, however, are not very mnemonic, so you may want to keep a record of the actual names which the short names represent.

☞**ALIASES** records the actual names corresponding to the short names you have already given the

exercises.

EXAMPLE

```
<Gr10>aliases
(ID ESSAY1)
Please type the (short) names of the exercises: essay1
Please specify the alias (the actual name):
ESSAY1: bulgarian-atrocities
```

☞**COMMENT** lets you include some explanations or comments in ***grade-file*** without stopping everything to use a file-editor. This command will ask for your comments, which must be enclosed in double-quotes, and append them to the end of ***grade-file***.

EXAMPLE

```
<Gr1>comment

Please enter the comment (enclosed in double-quotes):
"Charles missed TEST1 on account of illness."
```

Puts that comment in the ***grade-file***.

5.3. Automatic Grades: ETPS-GRADE, DUE-DATES

☞**DUE-DATES** lets you assign due-dates before using the **ETPS-GRADE** command described below, so that information about exercises completed late will automatically be recorded in the ***grade-file***. If due-dates have already been assigned, you can change them with this command. This command provides a convenient way of keeping track of due-dates even if the **ETPS-GRADE** command will not be used.

The date should be in the form *yy mm dd*, where `85 02 13` (for example) means 1985 Feb 13.

☞**ETPS-GRADE** collects grades from the record file, ***etps-file***, generated by a teaching program. The data in this record file is expected to be a column of parenthesized lists. The format is as follows: the student's user-id is the first element of the list; the exercise name is the second element; and the date completed should be the fifth element of the list. The date should be in the form of a list such as (*yy mm dd*), e.g., `(85 02 13)` means 1985 Feb 13. See the chapter *Notes on setting things up* in the TPS User's Manual for information about the creation of the ***etps-file*** to record grades for **ETPS** exercises.

As mentioned in Chapter 4, user-id's are required for all students in the class. If a student does not have one or it is not known, the dummy id `****` may be specified.

ETPS-GRADE will prompt you for exercises and the grades given for completing them on time. For each list in the ***etps-file*** showing a student completing one of these exercises, **ETPS-GRADE** gives that student the grade corresponding to that exercise. When **ETPS-GRADE** is called at a later date, it will leave previously recorded scores alone and insert scores for newly completed exercises. **GRADER** remembers the exercises which **ETPS-GRADE** has earlier processed, so only new exercises need to be entered when you are prompted for exercises. (If there are no new exercises, just reply with a return.)

If the assignment was late, the exercise and the number of days late are recorded. If no due-date can be found for an exercise, you will be prompted for it.

EXAMPLE If the the ***etps-file*** contains the following lists:

```
(bd79 diplo1 done 1 (85 05 02))
(ma24 diplo1 done 2 (85 05 08))
(jr03 diplo1 done 3 (85 05 08))
```

then running **ETPS-GRADE**

```
<Gr7>etps-grade
Please enter a list of exercises and scores. e.g. (x6200 15)(X6201
  20) :
(diplo1 40)

[merging ...]
Please specify the due-date for DIPLO1: 85 05 08
```

produces the new record:

```
NAME                 ID ESSAY1 DIPLO1
--MAX--             NIL     60     40
ARNOLD-MATTHEW     MA24     57     40
CARLYLE-THOMAS     TC32     54      *
DICKENS-CHARLES    CD51     53      *
DISRAELI-BENJAMIN  BD79     46     40
GASKELL-ELIZABETH  EG21     51      *
RUSKIN-JOHN        JR03     52     40
```

If aliases for exercises are being used, the actual names may appear in the ***etps-file***, but you should refer to the exercises by their short names when calling **ETPS-GRADE**. Thus, you should first call **DUE-DATES** using the short names to get these names into the ***grade-file***, then call **ALIASES** so **GRADER** will know which actual names in the ***etps-file*** correspond to which short names, then call **ETPS-GRADE**. Be sure that `ET` grades have this type.

EXAMPLE Suppose we wish to update the grade file by entering the scores the students have obtained on two exercises, x2109 and x2110, to which we wish to give the aliases E3 and E4, respectively. We shall give each of these exercises the due date 2006 October 27 (which we shall enter as 06 10 27), and we shall assign 9 points to E3 and 12 points to E4.

```
<Gr3>due-dates
ID A D C B G E I F H J T1 T1L E1 E2 K MID M L
Please type the names of the exercises(exe1 exe2 ...): E3 E4
Please specify the due-date:
E3: 06 10 27
E4: 06 10 27

<Gr4>aliases

ID A D C B G E I F H J T1 T1L E1 E2 K MID M L E3 E4
Please type the (short) names of the exercises: E3 E4
Please specify the alias (the actual name):
E3: x2109
E4: x2110

<Gr5>etps-grade
Please enter a list of exercises and scores. eg. (x6200 15)(X6201
  20)
Type <return> if you only want to update the previous scores
:(E3 9)(E4 12)
```

If you are using grader for several courses or sections of one course and you need to use **ETPS-GRADE** for each of these, exit **GRADER** and start fresh for each section to avoid complications.

5.4. Updating the Class List: ADD-STUDENTS, DELETE-STUDENT

There are two commands for updating the class list in ***grade-file***:

☞**ADD-STUDENTS** prompts you for a list of student records. Each student record is itself a parenthesized list where the first element is the student's name, the second his user-id (if these are used) and the subsequent elements are his grades on the exercises to date, in chronological order. Be warned that grades past the latest exercise will be ignored. **ADD-STUDENTS** will not introduce new exercises. If fewer grades are listed than exercises, the student will be given a grade of * on the later exercises.

EXAMPLE

```
<Gr8>add-students

Please type a list of students and their grades, e.g.,
(student1 grade1 grade2)(student2 grade1 grade2) etc.:
(wilde-oscar ow36 59 40)

Inserting WILDE-OSCAR.
```

☞**DELETE-STUDENT** prompts you for the names of students to be taken off the list.

EXAMPLE

```
<Gr9>delete-student

Please type the names of the  students:
carlyle-thomas gaskell-elizabeth

Deleted CARLYLE-THOMAS.
Deleted GASKELL-ELIZABETH.
```

So now the ***grade-file*** contains:

```
(
(NAME                ID ESSAY1 DIPLO1)
(TYPE                **     HW     ET)
(--MAX--             **     60     40)
(-MEAN-              **     53     40)
(-MEDIAN-            **     53     40)
(-SDEV-              **      4      0)
(-WEIGHT-            **      1      1)
(ARNOLD-MATTHEW    MA24     57     40)
(DICKENS-CHARLES   CD51     53      *)
(DISRAELI-BENJAMIN BD79     46     40)
(RUSKIN-JOHN       JR03     52     40)
(WILDE-OSCAR       OW36     59     40)
)
(;;;Record of late assignments:
((DICKENS-CHARLES ESSAY1 3 ))
;;;Record of penalty functions and due-dates:
(DIPLO1 NONE (85 5 8))
;;;List of actual names
(ESSAY1 BULGARIAN-ATROCITIES))
```

DELETE-STUDENT puts a record of students who have been removed from the class list at the end of the ***grade-file***.

5.5. Generated Values: STATISTICS

GRADER automatically generates statistical data when the record is updated. The statistical measures it updates are indicated by the value of ***statistical-options***. The default value for this variable is (`-sdev- -median- -mean-`), which stand for standard deviation, median and mean, respectively.

☞**STATISTICS** forces the calculation of the statistical measures. Normally this is done automatically, so this command is rarely needed.

5.6. Displaying Information: DISPLAY, NUMBER-OF-STUDENTS, INFO-EXERCISES

☞**DISPLAY** will show the grades for one or all of the students on the desired exercises. If all the grades for a student are requested, the late exercises and their number of days late will also be shown. The mean, standard deviation or median for an exercise may be found by responding with `-mean-, -sdev-` or `-median-` when prompted for a student name.

☞**NUMBER-OF-STUDENTS** gives you the size of the class.

☞**INFO-EXERCISES** lists the due-dates, maximum scores, aliases, weights and penalty-functions (see section 6.1) for the desired exercises.

EXAMPLE

```
<Gr4>infO-EXERCISES

Please type the names of the exercises: DIPLO1
DIPLO1:
     Maximum Score:      40
     Weight:             1
     Mean:               40
     Median:             40
     Standard Dev:       0
     Penalty Function:   NO-PENALTY
     Due-Date:           (85 5 8)
```

6. Figuring Grades

When you are ready to assign course grades, you should first make sure that the ***grade-file*** is up to date. You might need to use **ETPS-GRADES** and **LATE-EXERCISES** once more. Then use **CHANGE-WEIGHT**, **PENALTY-FNS**, and **CHG-VARS** to set the variables appropriately. Then call **CALCULATE-GRADE** to get the score on each item you wish to consider for each student. (An item can be a weighted total of various scores on exercises.) Then use **SORT-FN** to get nice displays of these scores. This should facilitate deciding which item you wish to use as the basis for assigning grades, and where the cut-offs between various grades (such as C+ and B-) should be. Then use **LETTER-GRADE** to record the letter grade for each student. (If you leave **GRADER** to examine the displays, you will probably have to run **CALCULATE-GRADE** again before using **LETTER-GRADE**, since the scores computed by **CALCULATE-GRADE** are considered temporary and are not recorded in ***grade-file***.)

6.1. Totaling: CHANGE-WEIGHT, CALCULATE-GRADE, PENALTY-FNS

☞**CHANGE-WEIGHT** allows you to examine and change the weights associated with exercises. The default weight is 1.

☞**CALCULATE-GRADE** computes the totals for each student in the class and prints the results in ***totals-grade-file***. All numbers calculated by **GRADER** are printed to values rounded ***print-n-digits*** places past the decimal. Thus, when ***print-n-digits*** is `2`, the value `64.98715` is printed `64.99`.

For each student, **CALCULATE-GRADE**

1. Sets aside the exercises with type `LE`. Their scores, being letter grades, are only symbols to **CALCULATE-GRADE**.
2. Reduces the effective score for each exercise submitted late by applying the penalty function assigned to that exercise. (See below.)
3. Drops the lowest effective score of each exercise-type which is an element of the list ***drop-min***.
4. Multiplies the effective score of each remaining exercise by the weight of the exercise, and sums these weighted scores for each exercise type. These sums for exercise types are recorded in ***totals-grade-file*** along with the items mentioned below.
5. Adds these sums for exercise types together, giving the total grade.
6. Computes the percentage score for this total grade, if ***cal-percentage*** is not `NIL`.
7. Computes the value of each item defined in ***new-item***.

The elements of ***new-item*** are lists such as (SCORE1 HW 1.8 ET 0.5), in which the first element is the name of the item and subsequent pairs of elements are each composed of the name of an exercise-type and the weight to be given that type. The value of the item is found by adding together the sums for the exercise-types multiplied by the corresponding weights.

EXAMPLE Items should be added using **CHG-VARS**:

```
<Gr5>chg-vars
The current value of some of the variables is:
1   Course-Name             "victoria"
2   Grade-File              "victoria.grades"
3   Totals-Grade-File       "victoria.totals"
4   Old-Grade-File          "victoria.xgrades"
5   Old-Totals-Grade-File   "victoria.xtotals"
6   Letter-grade-file       "victoria.letter"
7   ETPS-file               "etps.rec"
```

```
8   Statistical-options     (-MEAN- -MEDIAN- -SDEV-)
9   print-n-digits          0
10  cal-percentage          NIL
11  letter-grade-flag       T
12  new-item                NIL
13  drop-min                NIL
14  due-date-flag           T
15  default-penalty-fn      NO-PENALTY

Please type one of the numbers in the first column, or 16 to look
  at the
variables, or 17 to exit[Default:17]:12
Please type the new item(s): (home-study hw 1.5)
Do you want to change another variable,or look at their values?
  [No]>

<Gr6>calculate-grade

The old file victoria.totals has been renamed to
victoria.xtotals
Creating new copy of file victoria.totals
```

This creates the following table in the ***totals-grade-file***:

```
NAME              HOME-STUDY TOTAL ET HW
-SDEV-                 7    17 16  4
--MAX--               90   100 40 60
-MEAN-                80    85 32 53
-MEDIAN-              80    92 40 53
ARNOLD-MATTHEW        86    97 40 57
DICKENS-CHARLES       80    53  0 53
DISRAELI-BENJAMIN     69    86 40 46
RUSKIN-JOHN           78    92 40 52
WILDE-OSCAR           88    99 40 59

HOME-STUDY:
   HW 1.5
```

☞**PENALTY-FNS** assigns the penalty-functions used by **CALCULATE-GRADE**. These assignments are then recorded in the grade-file. The default penalty-function is `NO-PENALTY`. This function does not deduct any points for late exercises. The user can specify a different default penalty-function, by setting the ***default-penalty-fn*** to the name of the default function of his choice. The functions `PENALTY`, `PENALTY-0` and `PENALTY%10` are also available. `PENALTY` deducts one point for each day that the exercise was late. `PENALTY-0` assigns 0 points for each late exercise. `PENALTY%10` deducts `10%` for each day late.

With the default penalty-function, **CALCULATE-GRADE** produced the above table in ***totals-grade-file***.

However, if we execute **PENALTY-FNS** as follows:

```
<Gr7>penALTY-FNS
ID ESSAY1 DIPLO1
Please type the names of the exercises: essay1
Please specify the penalty-fn:
ESSAY1: penalty%10
```

and now have the grade-file:

```
(
```

```
(NAME                ID ESSAY1 DIPLO1)
(TYPE                **     HW     ET)
(--MAX--             **     60     40)
(-MEAN-               *     53     40)
(-MEDIAN-             *     53     40)
(-SDEV-               *      4      0)
(-WEIGHT-             *      1      1)
(ARNOLD-MATTHEW    MA24     57     40)
(DICKENS-CHARLES   CD51     53      *)
(DISRAELI-BENJAMIN BD79     46     40)
(RUSKIN-JOHN       JR03     52     40)
(WILDE-OSCAR       OW36     59     40)
)
(;;;Record of late assignments:
((DICKENS-CHARLES ESSAY1 3 ))
;;;Record of penalty functions and due-dates:
(ESSAY1 PENALTY%10 NONE
DIPLO1 NONE (85 5 8))
;;;List of actual names
(ESSAY1 BULGARIAN-ATROCITIES))
```

then **CALCULATE-GRADE** gives:

```
NAME              HOME-STUDY TOTAL ET HW
-SDEV-                    13    24 16  9

--MAX--                   90   100 40 60
-MEAN-                    75    82 32 50
-MEDIAN-                  78    92 40 52
ARNOLD-MATTHEW            86    97 40 57
DICKENS-CHARLES           52    35  0 35
DISRAELI-BENJAMIN         69    86 40 46
RUSKIN-JOHN               78    92 40 52
WILDE-OSCAR               89    99 40 59

HOME-STUDY:
   HW 1.5
```

The user may assign penalties using his own **LISP** functions, defined in his `GRADER.INI` file. These functions should expect to receive three arguments *score*, *days-late*, and *max-score*, and must return the penalized score. The function `penalty%,` a generalization of `PENALTY%10`, is provided for this purpose.

EXAMPLE

```
(defun penalty%5 (score dayslate max-score)
  (penalty% score dayslate max-score 5))
```

will deduct 5% for each day late.

6.2. Sorting: SORT-FN

☞**SORT-FN** will sort grades alphanumerically by the desired field. These fields either come with **GRADER** or are created by **CALCULATE-GRADE**. The sorted grades are appended to the ***totals-grade-file***. The first table of grades in ***totals-grade-file*** are already sorted by student name.

Before appending the sorted grades, **SORT-FN** will ask you which fields to display in addition to the student-name and the sorting fields. It will also prompt you for the style of the display, either Simple or Incremental.

In the Simple Display, the values in the fields are printed by descending order of the sorting field.

For the Incremental Display, you will be asked for an increment. The resulting display is partitioned by multiples of that increment.

EXAMPLE

```
<Gr8>sort-fn
HW ET TOTAL HOME-STUDY NAME ESSAY1 DIPLO1
Please choose the field: total
Do you want a display with increments? [Yes]>
ID ESSAY1 DIPLO1 HOME-STUDY TOTAL ET HW
Please select the items to be displayed (in addition to TOTAL)
[Default: NONE]: home-study
The maximum and minimum scores on TOTAL are 100 and 35
  respectively.
Please specify an increment for the display[Default:5]: 10
[Updating victoria.totals]

[Done.]
Do you want to sort the grades using another field? [No]>y
HW ET TOTAL HOME-STUDY NAME ESSAY1 DIPLO1
Please choose the field: essay1
ID ESSAY1 DIPLO1 HOME-STUDY TOTAL ET HW
Please select the items to be displayed (in addition to ESSAY1)
[Default: NONE]: diplo1
The maximum and minimum scores on ESSAY1 are 60 and 46
  respectively.
Please specify an increment for the display[Default:5]:
[Updating victoria.totals]

[Done.]
Do you want to sort the grades using another field? [No]>
```

The ***totals-grade-file*** would now contain the following tables:

```
NAME               HOME-STUDY TOTAL ET HW
-SDEV-                     13    24 16  9

--MAX--                    90   100 40 60
-MEAN-                     75    82 32 50
-MEDIAN-                   78    92 40 52
ARNOLD-MATTHEW             86    97 40 57
DICKENS-CHARLES            52    35  0 35
DISRAELI-BENJAMIN          69    86 40 46
RUSKIN-JOHN                78    92 40 52
WILDE-OSCAR                88    99 40 59

HOME-STUDY:
   HW 1.5

Grades sorted on the basis of TOTAL
     NAME              TOTAL HOME-STUDY



100  --MAX--             100         90
     WILDE-OSCAR          99         88
     ARNOLD-MATTHEW       97         86
     -MEDIAN-             92         78
     RUSKIN-JOHN          92         78
90   DISRAELI-BENJAMIN    86         69
```

```
     -MEAN-               82         75
80
70
60
50
40   DICKENS-CHARLES      35         52

Grades sorted on the basis of ESSAY1
     NAME              ESSAY1 DIPLO1



60   --MAX--              60     40
     WILDE-OSCAR          59     40
     ARNOLD-MATTHEW       57     40
55   -MEDIAN-             53     40
     -MEAN-               53     40
     DICKENS-CHARLES      52      *
     RUSKIN-JOHN          52     40
50   DISRAELI-BENJAMIN    46     40
```

6.3. Letter Grades: LETTER-GRADE

☞**LETTER-GRADE** assigns letter grades for a field with a numerical grade. A field can be an exercise, a type, a total or an item in ***new-item***. These grades will be recorded in ***letter-grade-file***, if the value of ***letter-grade-flag*** is not `nil`. They are also included in either ***grade-file*** or ***totals-grade-file***.

LETTER-GRADE will prompt you for the name of the new letter grade, the choice of ***grade-file*** or ***totals-grade-file*** and the field to which the letter grade is to be assigned.

Finally, you are prompted for a list of letter grades and the corresponding minimum scores. These grades need not be in numerical order. If a score below all the minimum scores is encountered, a warning message will be printed and the grade `*` assigned.

EXAMPLE

```
<Gr9>letter-grade
Please type the name for the new letter grade: midsem
Please type 1 if you want to add a letter grade to
  victoria.grades,
       type 2 if you want to add a letter grade to
  victoria.totals,
       type 3 if you want to add a letter grade to both: 2
ESSAY1 DIPLO1 HOME-STUDY TOTAL ET HW
Please choose the field: total
Please type a list of letter grades and the corresponding minimum
  scores,
e.g., (a 90)(b+ 80)(a- 85)(b 75)...(r 0):
(c 70)(a 90)(b 80)(r 40)
Unable to decide grade for DICKENS-CHARLES. Assigned *.
Creating file victoria.letter.
Updating file victoria.totals
DONE
```

Producing in ***letter-grade-file***:

```
ARNOLD-MATTHEW       A
DICKENS-CHARLES      *
DISRAELI-BENJAMIN    B
RUSKIN-JOHN          A
```

```
WILDE-OSCAR            A
```

It also appends the following table at the end of ***totals-grade-file***:

```
NAME              HOME-STUDY TOTAL ET HW MIDSEM
-SDEV-                    13    24 16  9      *
--MAX--                   90   100 40 60      A
-MEAN-                    75    82 32 50      B
-MEDIAN-                  78    92 40 52      A
ARNOLD-MATTHEW            86    97 40 57      A
DICKENS-CHARLES           52    35  0 35      *
DISRAELI-BENJAMIN         69    86 40 46      B
RUSKIN-JOHN               78    92 40 52      A
WILDE-OSCAR               88    99 40 59      A

TOTAL:
((A 90) (B 80) (C 70) (R 40))
```

Index

Table of Contents