

AcroT_EX.Net

PDF Flash Card: Arithmetic

D. P. Story

Table of Contents

1	Introduction	3
2	Requirements and Options	4
2.1	Requirements	4
2.2	PDF Creators	4
2.3	Options	4
3	The Components of the Flash Card	5
3.1	Settings	5
	• Options	7
	• Toggle Keypad	8
	• About PDF Flash Card	9
3.2	The <code>\arithProb</code> Command	9
3.3	The <code>\inputRegion</code> Command	11
3.4	The <code>\startAgain</code> and <code>\newCard</code> Command	12
3.5	The <code>\alertbox</code> Command	12
3.6	The <code>\Keypad</code> Command	13
3.7	The <code>\cbTiming</code> Command	13
3.8	The <code>\ansField</code> Command	14
3.9	The <code>\cbOperation</code> Command	15
3.10	The <code>\statsFields</code> Command	16
4	Other Customizations	16
4.1	Setting the Range and Decimal	16
4.2	Customization Strings	17
5	Suggested Layout	19

1. Introduction

The `fc_arith` package is used to create an electronic “flash card” used to drill a student on elementary arithmetic, addition, subtraction, multiplication, and division. There are options for setting the range of the numbers to be used to randomly select numbers that appear in the arithmetic problem. Numbers can be set to 0, 1, or 2 decimal places. There is an optional timing mechanism that can be used to test a student’s quickness in solving problems. There is a collection of fields used to tally the student’s work on the arithmetic problems.

History. This is a re-work of an earlier PDF on drilling arithmetic problems. At that time, I took a blank PDF page and used the user interface of Acrobat to build the flash card.

In this version, I’ve taken the original flash card and developed a \LaTeX package for generating the flash card, added a number of customization options, and many other enhancements.

Settings

PDF Flash Cards Elementary Arithmetic

NORTHWEST FLORIDA STATE COLLEGE

11

Right! + 14

25

Start Again New Card

No timing Answer: 25 Addition

+	-	x	÷	Operation
1				Number Correct
1				Number Attempted
100				Percentage Correct
3				Timed Scores

Figure 1: PDF Flash Card: Arithmetic

2. Requirements and Options

In this section, we review the requirements to build your own custom flash card, and describe the package options available.

2.1. Requirements

The package is designed to be used with either a Acrobat Distiller or a pdf \LaTeX workflow. The only true requirements are

- The eforms package (version 2.5c or later, 2010/03/21 or later), a package distributed with AeB (the Acro \TeX eDucation Bundle¹)
 - The popupmenu package² which creates a menu system.
- Sample files: `fc-acrobat.tex` and `fc-noacrobat.tex`.

I recommend the use of the `web` package to design your page, just as the sample files do. Enhancements include the use of the `aeb_pro` package; see the sample file `fc-acrobat.tex` for an illustration of the use of `aeb_pro`.

2.2. PDF Creators

Any of the PDF creators used by the \LaTeX community (`dvips`/`Distiller`, `pdf \LaTeX` , `lualatex`, and `xelatex`) may be used to build a PDF Flash Card document, as is demonstrated by the sample files `fc-acrobat.tex` and `fc-noacrobat.tex`. Users who prefer `pdf \LaTeX` , `lualatex`, and `xelatex` but own the Acrobat application may want to look at the sample file `fc-acrobat.tex`.

2.3. Options

1. `allownegsub`: The original arithmetic flash card did not allow a negative difference, if this option is used, the differences are allowed to be negative. The default is to not to allow negative differences.
2. `nomenu`: There is a menu, labeled Settings, that allows the user change the range of the numbers generated by the flash card, set the number of decimal places, allows the user to toggle the Keypad on and off, and so on. For further details see ‘Settings’ on page 5.
If this option is used, the menu does not appear. The default is for it to appear.
3. `showkeypadlink`: The `showkeypadlink` option generates a link with a label of Toggle Keypad just above the keypad. Clicking the link toggles the visibility of the keypad.

This option is useful if the `nomenu` option is taken, but you want the keypad available to users.

¹<http://www.math.uakron.edu/~dpstory/webeq.html>

²<http://mirror.ctan.org/macros/latex/contrib/popupmenu/>

4. `operations`: Gives the ability to declare what operations flash card should show. Supported operations are `add`, `sub`, `mul`, `div`. For example,

```
operations={add,sub}
```

creates a flash card in which only addition and subtraction problems are available. The default is to make available all operations.

5. `notimescores`: As can be seen in [Figure 1](#), page 3, the flash card has many fields that key of the user's performance. One of these is the Timed Scores row; this row gives a score based on time required to answer the problem correctly. Some teachers may not want such pressure placed on their students, so if this option is taken, the row is not generated.

3. The Components of the Flash Card

We describe the elements of the flash card as they appear, from top to bottom, in [Figure 1](#), page 3.

3.1. Settings

The Settings menu, which can be removed with the `nomenu` option, consists of five menu items: Options, Toggle Keypad, Mouse Friendly Keypad, Touch Friendly Keypad, and About PDF Flash Cards, see the figure below.

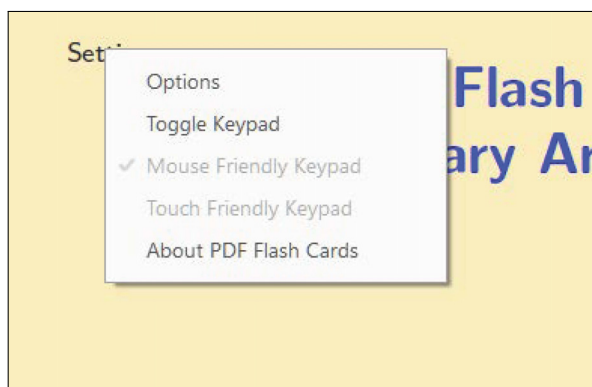


Figure 2: The Settings menu

The labeling for the Settings menu is modified using the command `\fcSettings`, the default definition is `\newcommand{\fcSettings}{Settings}`. The color of the link is set by the command `\fcSettingsColor`, the default definition of which is `\newcommand{\fcSettingsColor}{black}`.

The MenuFC environment. The Settings menu is created by the environment seen below and found in the demonstration files of this distribution. The MenuFC is a special environment defined in `fc_arith` to enclose the menu items.

```
\begin{MenuFC}
  \fcOptionsMenuItem
  \fcToggleKeypadMenuItem
  \fcMouseKPMenuItem
  \fcTouchKPMenuItem
  \fcAboutFC
\end{MenuFC}
```

You can exclude some of the menu items listed above. From Figure 2, the Mouse Friendly Keypad and Touch Friendly Keypad items are initially grayed out. They become active when Toggle Keypad is selected. The Touch Friendly Keypad is larger than the Mouse Friendly Keypad and fits fat fingers better. See ‘[Toggle Keypad](#)’ on page 8 for additional information on the keypad.

The names of the menu items may be localized to your own language.

```
\renewcommand\fcOptionsMenuItemTitle{Options}
\renewcommand\fcToggleKeypadMenuItemTitle{Toggle Keypad}
\renewcommand\fcMouseKPMenuItemTitle{Touch Friendly Keypad}
\renewcommand\fcTouchKPMenuItemTitle{Mouse Friendly Keypad}
\renewcommand\fcAboutFCTitle{About PDF Flash Cards}
```

The default English title defines are shown. All can be redefined in the preamble or in the customization file `fc_custom.def`, discussed in [Section 4.2](#).

Adding custom menu items. You can also add custom menu items, but some effort. Suppose we want to add an item titled AcroTeX.Net. The special environment generates a `popupmenu` environment, as defined by the `popupmenu` package. We use the `\item` command, as defined within a `popupmenu` environment and assign values for `title` and `return`. The value of `return` is any string, but `fc_arith` puts a restriction on its value: it must not be 0, 1, 2, 3, or 4, for these values are used by the preexistent menu items.

```
\begin{MenuFC}
  \fcOptionsMenuItem
  \fcToggleKeypadMenuItem
  \fcMouseKPMenuItem
  \fcTouchKPMenuItem
  \item{title=AcroTeX.Net Home,return=acrotexhome}
  \item{title=AcroTeX.Net Blog,return=acrotexblog}
  \fcAboutFC
\end{MenuFC}
```

The MenuFC environment occurs in the preamble of your document. Then to associate an action with the use selecting this new item, add the following code to the preamble as well:

```

\begin{insDLJS}{cmfc}{Custom Menu Events}
function processCustomFcMenu(n) {
  // n=the return value of the selected item
  switch(n) {
    case "acrotexhome":
      app.launchURL("http://www.acrotex.net",false);
      break;
    case "acrotexblog":
      app.launchURL("http://blog.acrotex.net",false);
      break;
  }
}
\end{insDLJS}

```

Here I use the JavaScript `switch` operator to take the incoming argument `n` (the return value) and associating an action. If the user-defined `processCustomerFcMenu()` is created, `fc_arith` passes the code stream from the function `processFcMenu()`, which handles all the predefined menu items, to `processCustomerFcMenu()`.

With these definitions, when the user selects 'AcroTeX.Net Home', the default browser opens and loads the page <http://www.acrotex.net>. This technique is illustrated in the file `fc_noacrobat.tex`.

• Options

The Options menu item calls forth the Options dialog box, [Figure 3](#), page 8. Through this dialog box, you can set the intervals from which random numbers are drawn, and their precision.

Addition, Subtraction, Multiplication. We handle these three operations in the same way, so we explain them together. For these three we have a top number and a bottom number.

top	14	17	12
bottom	+ 2	- 4	× 7

In the dialog box, you can set the range (the interval) of numbers from which randomly generated numbers are taken. For example, the [Figure 3](#) on page 8 specifies that for Addition problems, the top value should come from a range of 2 to 20, the bottom range is 5 to 30, and there should be no decimal places (hence, for this example, we deal with integer arithmetic). PDF Flash Card supports at most two decimal places.

Division. Division is handled separately because of the way the problem is generated. Rather than randomly generating the top and bottom values as is done for addition, subtraction, and multiplication, the `fc_arith` package randomly generates the *quotient* and *divisor*, then calculates the dividend

$$\text{dividend} = \text{quotient} \times \text{divisor}$$

This leads to the assurance of a clean division problem with no nasty rounding, or infinite decimal expansions ($1/30.33333\dots$). It is a design decision made for the original PDF Flash Card, where the targeted audience was elementary school age students.

The screenshot shows a dialog box titled "JavaScript Window" with a close button (X) in the top right corner. The dialog is organized into four sections, each for a different mathematical operation:

- Addition:**
 - Range Top: 2 to 20
 - Range Bottom: 5 to 30
 - Decimals: none, 1, 2
- Subtraction:**
 - Range Top: 1 to 5
 - Range Bottom: 5 to 10
 - Decimals: none, 1, 2
 - Allow Negative Subtraction:
- Multiplication:**
 - Range Top: 1 to 12
 - Range Bottom: 1 to 10
 - Decimals: none, 1, 2
- Division:**
 - Range Quotient: 1 to 4
 - Range Divisor: 1 to 10
 - Decimals: none, 1, 2

At the bottom of the dialog, there are two buttons: "OK" and "Cancel". Below the buttons, a warning message is displayed: "Warning: JavaScript Window".

Figure 3: The Options dialog box

The top is now the dividend and the bottom is the divisor, but in the Options dialog box, Range Top and Range Bottom is replaced by Range Quotient and Range Divisor to reflect the different way of generating the division problem.

► The dialog box retains the settings only during the current session of Adobe Reader. The Reader does not have the ability to save data so once the Flash Card is closed, the data is lost.

• Toggle Keypad

Selecting the Toggle Keypad submenu item toggles the keypad, which is initially hidden, see Figure 4. The keyboard input region is changed to read only, the student can only enter through the keypad.

The touch friendly keypad is larger by an amount of 5 points. This can be changed if you feel the touch friendly is not large enough by the command `\amtChngMouToTou`.


```
\newcommand\amtChngMouToTou{5}
```

The default definition is given above, use `\renewcommand` to change this, keeping in mind a larger keypad may go beyond the page. Some positioning adjustments may be needed.

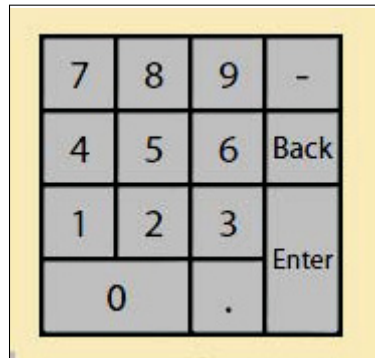


Figure 4: The Keypad

- **About PDF Flash Card**

Some information about the PDF Flash Card.

3.2. The `\arithProb` Command

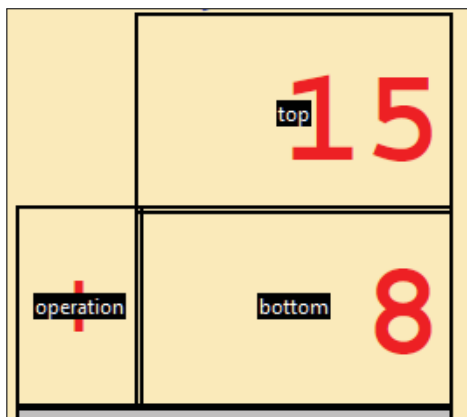
The `\arithProb` command is a trio of form fields laid out to display an arithmetic problem, see [Figure 5](#),³ and [Figure 6](#), page 11.

The three fields and their descriptions follow:

- **top**: The name of this text field is `top`, and it holds the upper most number in the problem (this number is called by different names depending on the operation, we'll just call it the top number).
- **bottom**: The name of this text field is `bottom`, and it holds the lower number in the problem.
- **operation**: The name of this text field is `operation`.

The `top` and `bottom` fields use a monospace font (`courier-bold`) so that when there are decimal numbers involved, the numbers will align properly.

³Figure 5 shows an outline of the fields so you can see their relative positions.

Figure 5: The `\arithProb` Command

Appearance Parameters for `\arithProb`. The three fields are bundled together as a unit, which makes it difficult to set the appearance of each individual field, unless you want to redefine `\arithProb`. There are, however, several commands available to make some adjustments to meet your needs.

- `\tBGNoBorder`: This is a command that holds various eform key-value pairs that are passed to these three fields. The default definition of `\tBGNoBorder` is

```
\newcommand{\tBGNoBorder}{\BC{}\BG{}\autoCenter{n}
\textSize{0}\textColor{1 0 0}\Ff\FfReadOnly}
```

This set of parameters gives a field with transparent border and background, auto-adjusting font size, red text, and with a readonly attribute. This command can be redefined.

- `\monoSpaceFont`: These three fields use a monospace font, you can change this font using `\monoSpaceFont`. The default definition is

```
\newcommand{\monoSpaceFont}{CoBo}
```

which is courier-bold.

- `\setDimOf`: The `\setDimOf` can be used to set the dimensions of these fields. The command takes three parameters, field name, width and height. The default definitions for these three fields are

```
\setDimOf{top}{1in}{0.62in}
\setDimOf{operation}{.38in}{.62in}
\setDimOf{bottom}{1in}{0.62in}
```

Note that the dimensions of `top` and `bottom` are the same, as they are supposed to align vertically.

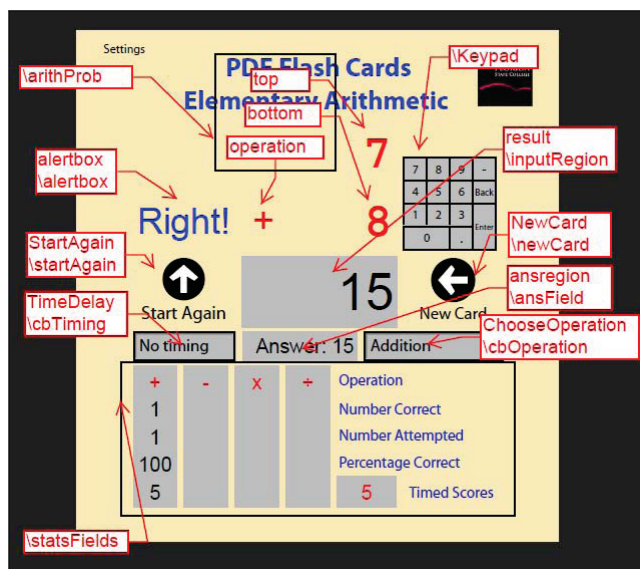


Figure 6: The PDF Flash Card, Annotated

3.3. The `\inputRegion` Command

When the keypad field is hidden (see the ‘[The `\Keypad` Command](#)’ on page 13), the user inputs the response to the arithmetic problem into the field titled `result`. This field is placed on the flash card using the `\inputRegion` command; this field is usually placed immediately beneath the `\arithProb` command so that the `result` field is aligned vertically with the `top` and `bottom` fields.

The syntax for `\inputRegion` is

```
\inputRegion[<eform_parameters>]
```

The default appearance properties are pre-defined as

```
\newcommand{\cBGNoBorder}{\Q{1}\BC{} \BG{}}\autoCenter{n}
\textSize{0}\textColor{0 0 0}\BG{.75 .75 .75}
\Ff\FfReadOnly}
```

You can redefine `\cBGNoBorder`, or you can introduce a few changes through the optional parameter.

The dimensions of the `result` fields can be set using `\setDimOf`. The default definition is

```
\setDimOf{result}{1.38in}{0.62in}
```

The `0.62in` is the same width used for the `top` and `bottom` fields.

3.4. The `\startAgain` and `\newCard` Command

The `\newCard` command generates a push button named `NewProblem`, when pushed, a new arithmetic is randomly generated. The `\startAgain` command creates a push button named `StartAgain`, when pushed, all fields are cleared, and any JS variables are re-initialized. The syntax for these two fields is

```
\startAgain[⟨eform_parameters⟩]
\newCard[⟨eform_parameters⟩]
```

The default appearance properties are pre-defined as

```
\newcommand{\tBGNoBorderI}{\BC{}\BG{}\autoCenter{n}
\textSize{0}\textColor{1 0 0}}
```

You can redefine `\tBGNoBorderI`, or you can introduce a few changes through the optional parameter.

The dimensions of `NewProblem` and `StartAgain` can be set using `\setDimOf`. The default definition is

```
\setDimOf{StartAgain}{0.88in}{0.62in}
\setDimOf{NewProblem}{0.88in}{0.62in}
```

The text font used by these two fields is determined by the command `\fieldFont`. The default definition of `\fieldFont` is

```
\newcommand{\fieldFont}{Helv}
```

The name for the font should be one of the 13 basic fonts, or a PostScript font. In the latter case, most likely, the Acrobat Distiller is needed for PDF creation.

3.5. The `\alertbox` Command

The `\alertbox` command creates a transparent text field named `alertbox`, which displays `Right!` and `Wrong!` messages. The syntax for this field is

```
\alertbox[⟨eform_parameters⟩]
```

The default appearance properties are pre-defined as

```
\newcommand{\tBGNoBorder}{\BC{}\BG{}\autoCenter{n}
\textSize{0}\textColor{1 0 0}\Ff\FfReadOnly}
```

You can redefine `\tBGNoBorder`, or you can introduce a few changes through the optional parameter.

The dimensions of `alertbox` can be set using `\setDimOf`. The default definition is

```
\setDimOf{alertbox}{.88in}{.62in}
```

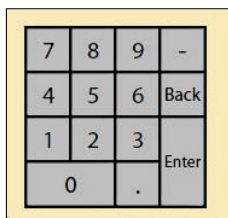


Figure 7: The Keypad

3.6. The `\Keypad` Command

The `\Keypad` creates a number of fields in the form of a keypad, see [Figure 3.6](#). When the keypad is visible, the result field (created by `\inputRegion`) is read-only.

The default appearance properties are pre-defined as

```
\newcommand{\tBGNoBorder}{\BC{} \BG{} \autoCenter{n}
\textSize{0} \textColor{1 0 0} \Ff\FfReadOnly}
```

You can redefine `\tBGNoBorder`, or you can introduce a few changes through the optional parameter.

The dimension of each key is based on the value of `\szNum`, the default definition is

```
\newcommand{\szNum}{14bp}
```

The default appearance properties are pre-defined as

```
\newcommand{\myNumPadI}{\F\FHidden \autoCenter{n}
\textSize{8} \textFont{\fieldFont} \S{S}}
```

You can redefine `\myNumPadI`.

The labeling for the Enter and Back keys can be redefined as well, their definitions are

```
\newcommand{\kpBack}{Back}
\newcommand{\kpEnter}{Enter}
```

3.7. The `\cbTiming` Command

The `\cbTiming` command creates a combo box, named `TimeDelay`, that lists a menu of time values for the user to practice against. The default is No Timing.

Rules for Timed Responses. When No Timing is in effect, if the student correctly answers the problem in the time allotted, credit is awarded; otherwise, no credit is given. When another time setting is in effect, the same rule applies as above, except no credit is awarded if the time is up, even if the response is correct. Click on the push button labeled Timed Scores to see the rules for awarding the points, depicted in [Figure 8](#).

The strings of this dialog box may be localize to another language, refer to [Section 4.2](#) for details.

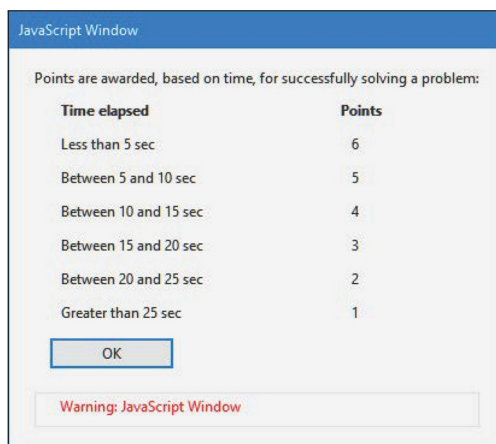


Figure 8: The timed scores dialog box

Note: The original idea behind the timed scores is to have students compete against each other. For example, for addition, let each student attempt 10 problems, the highest score wins! Try this with No Timing first, then start decreasing the time value progressively down to 5 sec. Who is the fastest with the mostest!

The syntax for `\cbTiming` is

```
\cbTiming[<efrom_parameters>]
```

The default appearance properties are pre-defined as

```
\newcommand{\cBGBorder}{\BC{0 0 0}\BG{.75 .75 .75}
\autoCenter{n}\textSize{0}\textColor{0 0 0}}
```

You can redefine `\cBGBorder`, or you can introduce a few changes through the optional parameter.

The dimensions of `TimeDelay` field can be set using `\setDimOf`:

```
\setDimOf{TimeDelay}{0.9in}{0.24in}
```

The font used in the list box is determined by the value of `\fieldFont`, the default definition is `\newcommand{\fieldFont}{Helv}`.

The string `No Timing` can be modified using `\fcNoTiming`:

```
\newcommand{\fcNoTiming}{No Timing}
```

3.8. The `\ansField` Command

The `\ansField` creates a text field named `ans` region; the field holds the correct answer to the problem.

The syntax for `\cbTiming` is

```
\ansField[<efrom_parameters>]
```

The appearance of this field can be modified using `\cBGNoBorder`; the default definition is,

```
\newcommand{\cBGNoBorder}{\Q{1}\BC{}\BG{}\autoCenter{n}
\textSize{0}\textColor{0 0 0}\BG{.75 .75 .75}\Ff\FfReadOnly}
```

This can be redefined as desired.

The dimensions of `ansregion` can be set be `\setDimOf`:

```
\setDimOf{ansregion}{.87in+10bp}{.24in}
```

Notice that a little arithmetic on dimensions is used, this is because the `fc_arith` package inputs the `calc` package.

The `ansregion` has a formatting string, it prepends `Answer:` to the correct answer. You can change this word using the command `\fmtAnswer` the definition of which is

```
\newcommand{\fmtAnswer}{Answer:}
```

3.9. The `\cbOperation` Command

The `\cbOperation` command creates a combo box, named `ChooseOperation`, that lists a menu of arithmetic operations for the user to select.

The syntax for `\cbOperation` is

```
\cbOperation[(eform_parameters)]
```

The default appearance properties are pre-defined as

```
\newcommand{\cBGBorder}{\BC{0 0 0}\BG{.75 .75 .75}
\autoCenter{n}\textSize{0}\textColor{0 0 0}}
```

You can redefine `\cBGBorder`, or you can introduce a few changes through the optional parameter.

The dimensions of `ChooseOperation` field can be set using `\setDimOf`:

```
\setDimOf{ChooseOperation}{1.38in-10bp}{0.24in}
```

The font used in the list box is determined by the value of `\fieldFont`, the default definition is `\newcommand{\fieldFont}{Helv}`.

The strings used in the combo box can be modified by redefining the following commands:

```
\newcommand{\fcAddition}{Addition}
\newcommand{\fcSubtraction}{Subtraction}
\newcommand{\fcMultiplication}{Multiplication}
\newcommand{\fcDivision}{Division}
```

3.10. The `\statsFields` Command

The `\statsField` command creates a number of fields that hold the statistics of the user's attempts at answering arithmetic problems.

The following definitions are used.

```
\setDimOf{fcSF}{0.37in}{0.25in}
\newcommand{\statsFieldOpColor}{1 0 0}
\newcommand{\statsFieldColor}{blue}
```

The first is to set the dimensions of each field, they all have the same dimension. The second sets the color for the operations of $+$, $-$, \times and \div ; the default color for the labeling of these operations is red (`{1 0 0}` in the RGB color space. The `\statsFieldColor` is a color of typeset content and its default is blue. (`\statsFieldOpColor` is used in form fields, while `\statsFieldColor` is \LaTeX ed.)

4. Other Customizations

In this section, we itemize various other customizations not already mentioned.

4.1. Setting the Range and Decimal

The command `\DeclareArithParams`—executed in the preamble only—is used to set the start-up parameters of the Options menu [Figure 3](#) on page 8.⁴

The `\DeclareArithParams` takes a series of key-values. We classify the keys by function: setting the intervals and setting the decimal places.

- **Setting the intervals.** Each of these keys takes an interval of the form $[a, b]$. Because there is a comma in the interval notation, the interval needs to be enclosed in braces, like so `{[a, b]}`.
 - **Addition:** `addT={ [a, b] }, addB={ [a, b] }`
 - **Subtraction:** `subT={ [a, b] }, subB={ [a, b] }`
 - **Multiplication:** `mulT={ [a, b] }, mulB={ [a, b] }`
 - **Division:** `divQ={ [a, b] }, divB={ [a, b] }`
- **Setting the decimal places.** The value of these keys take any of three values none, 1, or 2. In all cases, the default is none.
 - **Addition:** `addDecT=none|1|2, addDecB=none|1|2`
 - **Subtraction:** `subDecT=none|1|2, subDecB=none|1|2`
 - **Multiplication:** `mulT=none|1|2, mulDecB=none|1|2`
 - **Division:** `divDecQ=none|1|2, divDecB=none|1|2`

The default definitions are

⁴If the `nomenu` option is taken, the parameters are still used, but user has no way of changing them.


```

\DeclareArithParams
{%
  addT={ [0,100] }, addB={ [0,100] },
  subT={ [0,100] }, subB={ [0,100] },
  mulT={ [0,100] }, mulB={ [0,10] },
  divQ={ [0,10] }, divB={ [0,10] },
}

```

The decimal points are all set to none, by default.

Below is a complete example of setting all parameters.

```

\DeclareArithParams
{%
  addT={ [2,20] }, addB={ [5,30] },
  addDecT=none, addDecB=2,
  subT={ [1,5] }, subB={ [5,10] },
  subDecT=1, subDecB=2,
  mulT={ [1,12] }, mulB={ [1,10] },
  mulDecT=1, mulDecB=2,
  divQ={ [1,4] }, divB={ [1,10] },
  divDecQ=1, divDecB=2,
}

```

4.2. Customization Strings

The following English strings are used in Flash Card. These can all be redefined.

The following is a message that appears in an alert box when your time is up.

```
\newcommand{\timeUpMsg}{Your Time is UP!}
```

These next two are the default messages in the alertbox text field.

```

\newcommand{\rightMsg}{Right!}
\newcommand{\wrongMsg}{Wrong!}

```

The text on the StartAgain and NewProblem.

```

\newcommand{\startAgainMsg}{Start Again}
\newcommand{\newCardMsg}{New Card}

```

The formatting of the ansfield.

```
\newcommand{\fmtAnswer}{Answer:}
```

Labels that appear on the key pad.

```

\newcommand{\kpBack}{Back}
\newcommand{\kpEnter}{Enter}

```

The default setting for the TimeDelay combo box.

```
\newcommand{\fcNoTiming}{No Timing}
```

The listing of arithmetic operations in to combo box titled ChooseOperation.

```
\newcommand{\fcAddition}{Addition}
\newcommand{\fcSubtraction}{Subtraction}
\newcommand{\fcMultiplication}{Multiplication}
\newcommand{\fcDivision}{Division}
```

The text of the menu that, by default, appears in the upper left-corner.

```
\newcommand{\fcSettings}{Settings}
\renewcommand\fcOptionsMenuMenuItemTitle{Options}
\renewcommand\fcToggleKeypadMenuItemTitle{Toggle Keypad}
\renewcommand\fcMouseKPMenuItemTitle{Touch Friendly Keypad}
\renewcommand\fcTouchKPMenuItemTitle{Mouse Friendly Keypad}
\renewcommand\fcAboutFCTitle{About PDF Flash Cards}
```

Some text strings that are part of the \statsField set.

```
\newcommand{\toggleKeypad}{Toggle Keypad}
\newcommand{\operation}{Operation}
\newcommand{\numCorrect}{Number Correct}
\newcommand{\numAttempted}{Number Attempted}
\newcommand{\percentCorrect}{Percentage Correct}
\newcommand{\timedScores}{Timed Scores}
```

Some strings in the timed scores dialog box (Figure 8).

```
\def\fcCharWidth{22}
\def\fcInstr{"Points are awarded, based on time,
for successfully solving a problem:"}
\def\fcTimeElapsed{"Time elapsed"}
\def\fcPoints{"Points"}
\def\fcLessThanV{"Less than 5 sec"}
\def\fcLessThanVPoints{6}
\def\fcBtwnVAndX{"Between 5 and 10 sec"}
\def\fcBtwnVAndXPoints{5}
\def\fcBtwnXAndXV{"Between 10 and 15 sec"}
\def\fcBtwnXAndXVPoints{4}
\def\fcBtwnXVAndXX{"Between 15 and 20 sec"}
\def\fcBtwnXVAndXXPoints{3}
\def\fcBtwnXXAndXXV{"Between 20 and 25 sec"}
\def\fcBtwnXXAndXXVPoints{2}
\def\fcGtrXXV{"Greater than 25 sec"}
\def\fcGtrXXVPoints{1}
```

The following are the strings used for the naming the clusters of the Settings > Options menu.

```

\newcommand\fcAdditionName{\fcAddition}
\newcommand\fcSubtractionName{\fcSubtraction}
\newcommand\fcMultiplicationName{\fcMultiplication}
\newcommand\fcDivisionName{\fcDivision}

```

If the Settings control is available, but the MenuFC environment was not used to create any menu items, an alert box appears when the user presses the Settings control with the following message on it.

```

\newcommand\fcMenuFCMsg{No menu items to display}

```

Various tool tips that appear on various controls.

```

\newcommand\toggleKeyPadBtnColor{0 0 1}
\newcommand\toggleKeyPadBtnTooltip{Click to toggle keypad,
  shift-click to toggle between mouse and touch keypads}
\newcommand{\fcSettingsTooltip}{Click for a dropdown menu
  of menu choices}
\newcommand{\cbTimingTooltip}{Select a time challenge from
  the dropdown menu}
\newcommand{\cbOperationTooltip}{Choose an arithmetic
  operation to practice}
\newcommand{\timeScoresTooltip}{Click to see how points
  are assigned}

```

The strings to localize the Timed Scores dialog box. You may have to increase the value of `\fcOptTextWidth` from 80 (pixels) to something larger if your language strings are much longer than their English counterparts.

```

\newcommand\fcOptTextWidth{80}
\newcommand\fcOptTopRange{Range Top}
\newcommand\fcOptBottomRange{Range Bottom}
\newcommand\fcOptTopRangeDiv{Range Quotient}
\newcommand\fcOptBottomRangeDiv{Range Divisor}
\newcommand\fcOptTo{ to }
\newcommand\fcOptAllowNegNumber{Allow Negative Subtraction}
\newcommand\fcOptDecimal{Decimals:}
\newcommand\fcOptDecimalNone{none}

```

All this customization commands are listed in `fc-strings.txt`. There they can be redefined and copy and pasted into the file `fc_custom.def`.

Customization. Any change in language strings can be put in the file `fc_custom.def`. Such a file is input, if found, at the end of the package.

5. Suggested Layout

The files `fc-acrobat.tex` and `fc-noacrobat.tex` contain the original layout of PDF Flash Card. We include that layout in this manual for completeness sake.

```

%
% Design your own title
%
\begin{center}
{%
  \LARGE\bfseries\color{blue}PDF Flash Cards\[\1ex]
  Elementary Arithmetic
}
}

%
% The arithmetic problem, \arithProb: top, bottom and operation.
% This command is REQUIRED. This command generates three text fields
% stacked so that form a standard arithmetic
%
\arithProb

%
% \alertbox is a text field where a right or wrong message is
% written--REQUIRED
%
% \startAgain clears the statistics field, re-initializes a
% variables--REQUIRED
%
% \inputRegion is where the user enters his/her answer--REQUIRED
%
% \Keypad allows user to enter answer with mouse--OPTIONAL
%
% \newCard random selects a new arithmetic problem (add, sub, mul, div)
% depending on the combo box \cbOperation, described below--REQUIRED
%
% These components can be moved around to a new design, though I don't
% know what that would be. I have no imagination for design myself.
%
\mbox{\vbox{\smash{\alertbox}\startAgain}\fcSep
  \inputRegion\fcSep\vbox{\smash{\raisebox{4bp}{\Keypad}}\newCard}}

\medskip
%
% \cbTiming is combo box use to set a time limit on answer the
% problem--OPTIONAL
%
% There is also a package option, notimescores, that removes timing
% calculations from the PDF, no statistics, no alerts. The option
% notimescores makes the combo box created by \cbTiming into a readonly
% field.
%
% \ansField is the field the user enters his/her answer into---REQUIRED
%
% \cbOperation a combo box to select what operation to use---REQUIRED

```

```
%
\mbox{\cbTiming\fcSep\ansField\fcSep\cbOperation}

\medskip

%
% This calculation computes the width of the previous row of fields, and
% sets \fcWidth, a dimension in this package, to that width.
%
\settowidth{\fcWidth}{\cbTiming\fcSep\ansField\fcSep\cbOperation}
%
% \statsFields is a collection of text fields to display user
% statistics---OPTIONAL
%
\makebox[\fcWidth][s]{\statsFields}

\end{center}
```