# Introduction: Getting Started

This guide is designed to help a user who is new to using WeBWorK (WW). In this document, you will find out how to create a class list, make homework sets, edit existing problems, etc. Part 1 covers the basics, Part 2 deals with how to do basic edits on existing problems, Part 3 is a collection of advanced problem editing tips, and Part 4 will lead you into WeBWorK 's online documentation. A Test Set of problems has been prepared as a companion to this guide, containing different kinds of sample problems. We will explain in Part 4 how to access the Test Set.

## Part 1: The Basics

### Section 1.1: Creating a Class List

This section will walk you through creating a class list for your class. Upon logging in to WebWork you will see all the problem sets for your class (unless you have not created any problem sets).

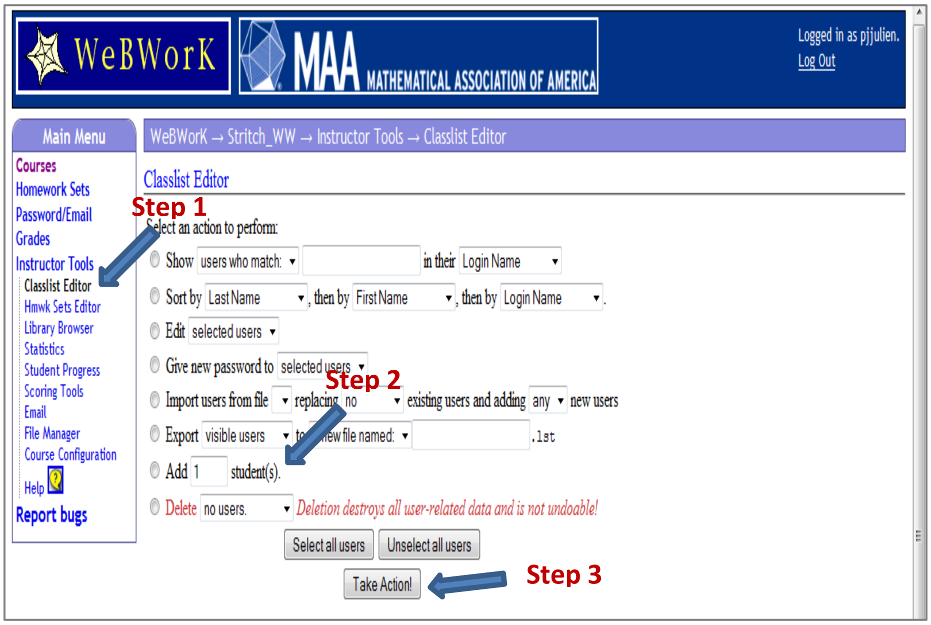
Step 1 Select the **Classlist Editor** (Exhibit 1.1) option in the main menu box. (The Main Menu box is on the left hand side of the screen, directly under the WebWork Log.)

Step 2 Select the **"Add___ student(s)"** (Exhibit 1.1) radio button. (This is the 7th option down.)

Step 3 Type in the number of student that will be in the course and click the **"Take Action!"** button. (Exhibit 1.1)

Step 4 Enter student(s') information and click **"Add Students"**

*Exhibit 1.1*



Classlist Editor

Select an action to perform:

○ Show users who match: ▾ in their Login Name ▾

○ Sort by Last Name ▾, then by First Name ▾, then by Login Name ▾.

○ Edit selected users ▾

○ Give new password to selected users ▾

○ Import users from file ▾ replacing no ▾ existing users and adding any ▾ new users

○ Export visible users ▾ to new file named: ▾ .lst

○ Add 1 student(s).

○ Delete no users. ▾ *Deletion destroys all user-related data and is not undoable!*

## Editing A Class List

### Change Permission Level

By default each new user that is added will be added as a student, but this can be changed. For example, if one of the users is TA then they should have more permissions than a student.

Step 1 Select user(s) that need to have their permission level changed.
Step 2 Select the **"Editing Selected Users"** radio button. (3rd from the top)
Step 3 Click the **"Take Action!"** button.
Step 4 Change the permission level by selecting the drop down menu and the **Change Permission Level**.
Step 5 Select the "**Save Changes"** radio button and then select **"Take Action."**

Different Permission Levels

- Guest
  - Will allow a user to log in.
- Student
  - Will allow a user to work on problem sets that have been assigned to them.
- Login proctor
- Grade Proctor
- TA
  - Will allow a user to assist with the problems. This user will also have limited access to the class list.
- Professor
  - Will allow a user to have full access to the all the WeBWorK features.

## Section 1.2: Creating a Problem Set

There are two ways to create a problem set: under the **Hmwk Sets Editor** menu option or under the **Library Browser** menu option.

### *Option 1: Hmwk Sets Editor*

Step 1 Click the **"Hmwk Sets Editor"** link from the Main Menu box.
Step 2 Select the **"Create a new set named____"** radio button and give the set a name.

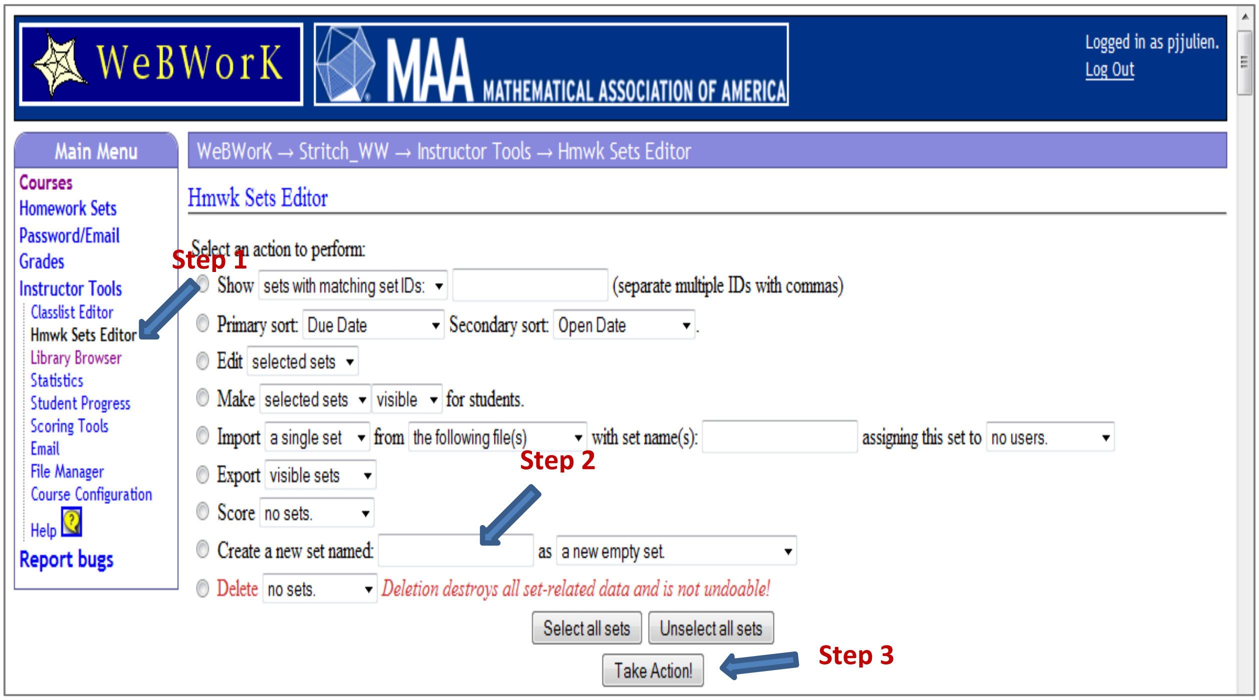
- Selecting "**<u>as a new empty set</u>"** will create an empty set with the name that you gave it.
- Selecting **"as <u>a duplicate of the first selected set</u>"** will create a problem set that is an exact replica of a problem set that has been previously selected.

Step 3 Select the **"Take Action"** button.
Step 4 Once the problem set has been created, it is possible to select problems to put into the new problem set.

- Click the **Library Browser** menu option and select problems to add to your problem set.

- Problems specifically for the CMA book can be found by selecting the CMA button. (See below for the step-by-step way to select and add problems.)

*Exhibit 1.2: Option 1 for creating a problem set*



Hmwk Sets Editor

Select an action to perform:

○ Show sets with matching set IDs: [ ] (separate multiple IDs with commas)

○ Primary sort: Due Date Secondary sort: Open Date.

○ Edit selected sets

○ Make selected sets visible for students.

○ Import a single set from the following file(s) with set name(s): [ ] assigning this set to no users.

○ Export visible sets

○ Score no sets.

○ Create a new set named: [ ] as a new empty set.

○ Delete no sets. *Deletion destroys all set-related data and is not undoable!*

Step 1

Step 2

Step 3

### *Option 2: Through Library Browser*

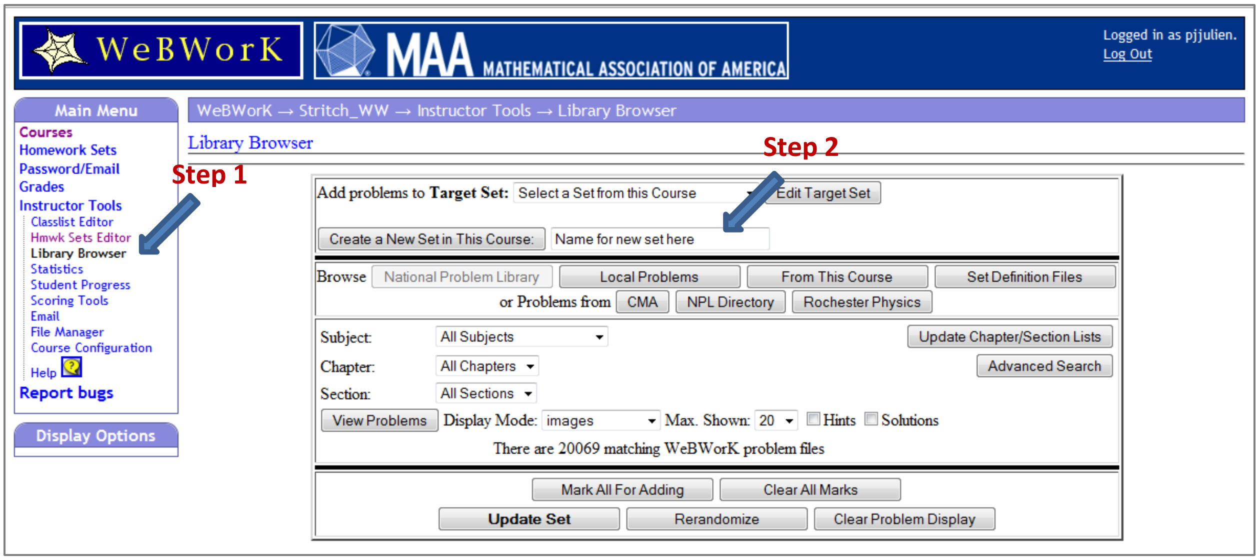
Step 1 Click the **"Library Browser"** link from the Main Menu box.

Step 2 Enter the name for the problem set and select the **"Create a New Set in this Course."**

Step 3 Once the problem set has been created, it is possible to select problems to put into the new problem set.

- Problems specifically for the CMA book can be found by selecting the CMA button. (See below for the step-by-step way to select and add problems.)

*Exhibit 1.3 Option 2 for Creating a Problem Set*



Step 1

Step 2

Add problems to **Target Set:** Select a Set from this Course

Name for new set here

Subject: All Subjects

Chapter: All Chapters

Section: All Sections

Display Mode: images

Max. Shown: 20

There are 20069 matching WeBWorK problem files

## Section 1.3: Adding Problems to a Problem Set

Step 1 Click the **Library Browser** menu option.

Step 2 Select from the "**Add problems to Target Set"** drop down menu the problem set you want to add problems to.

Step 3 Select the location that you want the problems to come from.

- The problems for the Calculus: Modeling & Application text book are under the **CMA** button.
- If you want to view all the problems in the library select the **National Problem Library** button. See more on the **National Problem Library** in **Part 4: Documentation.**

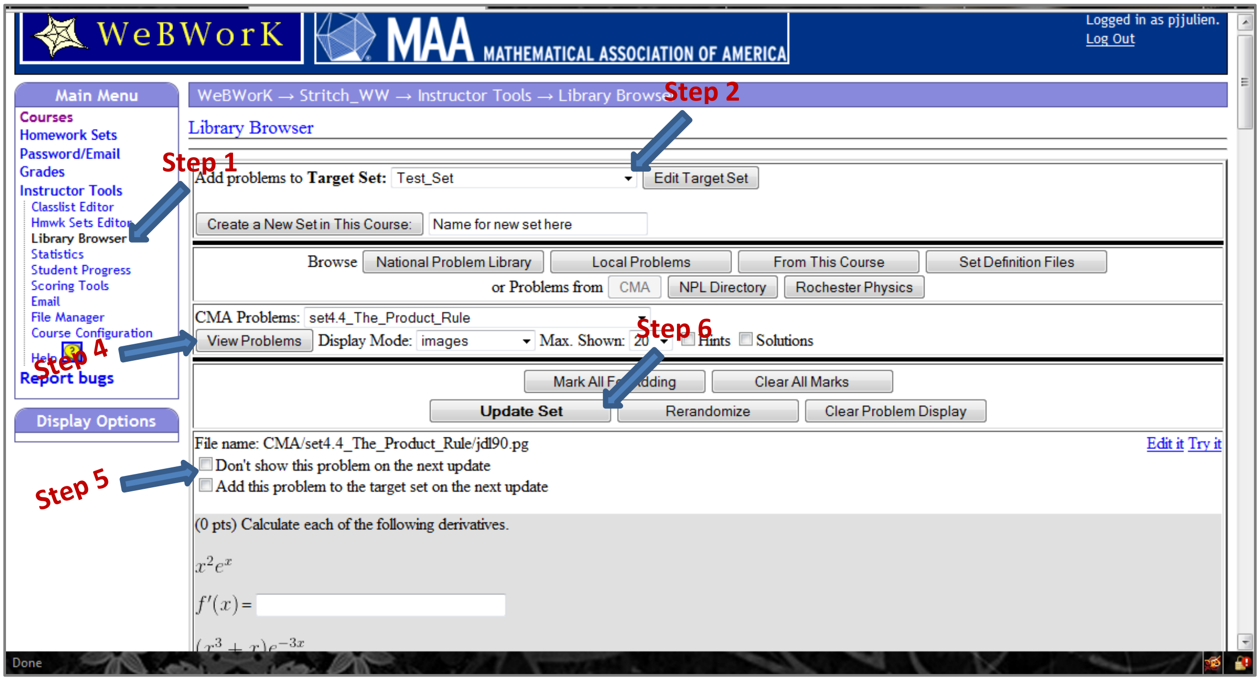
Step 4 Next you want to select the chapter and section the problems should come from. Then click **View Problems.**

Step 5 Reviewing the list of problems for the corresponding chapter and section, you have two options for each problem:

- The ***first*** check box: if selected, the problem will not show up after the **Update Set** button is clicked.
- The ***second*** check box: if selected, the problem will be added to the selected problem set.

Step 6 After the desired check boxes are selected, click the **Update Set** button at the bottom or top of the page.

*Exhibit 1.4 Adding Problems to a Problem Set*



Add problems to **Target Set:** Test_Set

Create a New Set in This Course: Name for new set here

CMA Problems: set4.4_The_Product_Rule

Display Mode: images Max. Shown: 20 □ Hints □ Solutions

File name: CMA/set4.4_The_Product_Rule/jdl90.pg

□ Don't show this problem on the next update

□ Add this problem to the target set on the next update

(0 pts) Calculate each of the following derivatives.

$x^2e^x$

$f'(x)=$

$(x^3+x)e^{-3x}$

## Section 1.4: Assigning Problem Set to Students

Now it's time to assign the Problem set to students.

Step 1 Click the **"Hmwk Sets Editor"** link from the Main Menu box.

Step 2 Find the desired Problem Set and select the link in the **"Edit Assigned User"** column. (This is the fourth column from the left. It should say "0/out of the # of students")

Step 3 Select the users for whom the problem set should be assigned to and click the **Save** button.

# Part 2: Minor Tweaks

This section will give tips for making minor tweaks to existing problems.

## Section 2.1: Erasing part of a problem

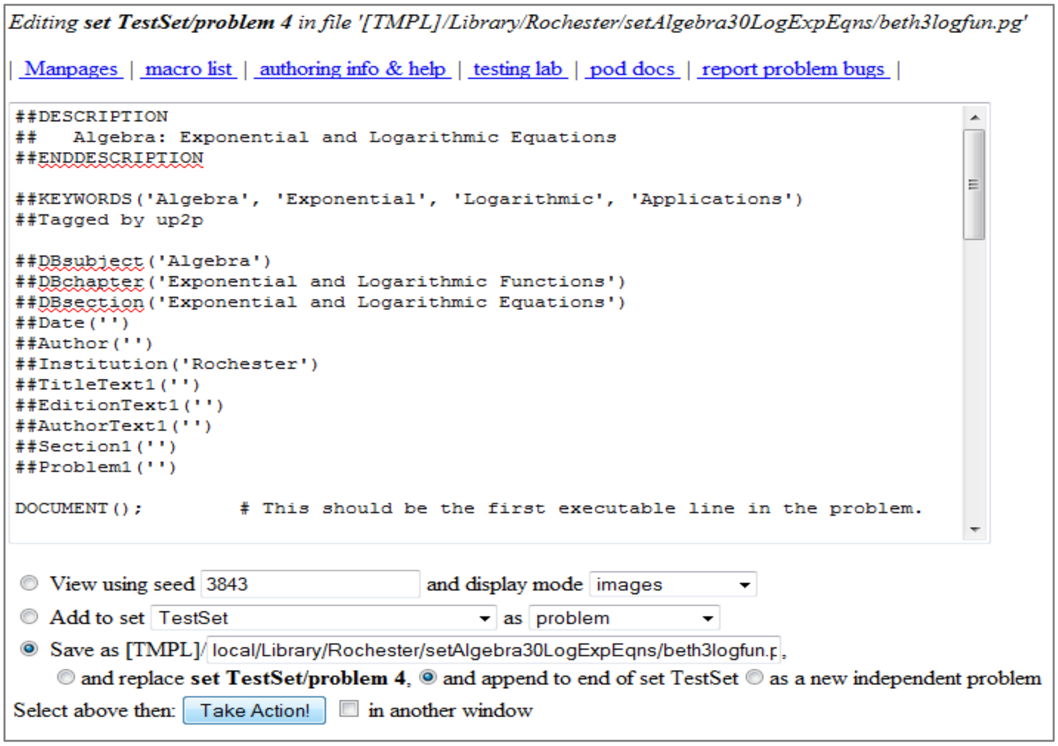
In order to edit an existing problem, you must create a duplicate of the problem locally.

Step 1 When looking at the problem you want to edit there will be an **"Edit this problem"** link.

Step 2 Select the radio button that says "**Save as [TMPL]/local/*Problem name***" and the second of the three options that are under this one. It will say, **"and append to end of set *(Set Name)."***

Step 3 Click the **Take Action** button.

*Exhibit 2.1 Creating a duplicate problem locally*



*Editing **set TestSet/problem 4** in file '[TMPL]/Library/Rochester/setAlgebra30LogExpEqns/beth3logfun.pg'*

| Manpages | macro list | authoring info & help | testing lab | pod docs | report problem bugs |

```
##DESCRIPTION
##   Algebra: Exponential and Logarithmic Equations
##ENDDESCRIPTION

##KEYWORDS('Algebra', 'Exponential', 'Logarithmic', 'Applications')
##Tagged by up2p

##DBsubject('Algebra')
##DBchapter('Exponential and Logarithmic Functions')
##DBsection('Exponential and Logarithmic Equations')
##Date('')
##Author('')
##Institution('Rochester')
##TitleText1('')
##EditionText1('')
##AuthorText1('')
##Section1('')
##Problem1('')

DOCUMENT();        # This should be the first executable line in the problem.
```

○ View using seed 3843 and display mode images

○ Add to set TestSet as problem

● Save as [TMPL]/ local/Library/Rochester/setAlgebra30LogExpEqns/beth3logfun.p,

○ and replace **set TestSet/problem 4**, ● and append to end of set TestSet ○ as a new independent problem

Select above then: □ in another window

This will add a new problem to the end of your problem set named **local/Problem Name.** A green bar will appear and will notify you that the new problem has been added. Once the problem is local it can be edited. Also keep in mind that if you want to rename the problem, you can repeat these steps. Just delete the current name that is in the text box and put a new name in.

Now you can erase the undesired part of the problem.

Step 4 Go to the new problem. (It will be the last problem of the problem set.)

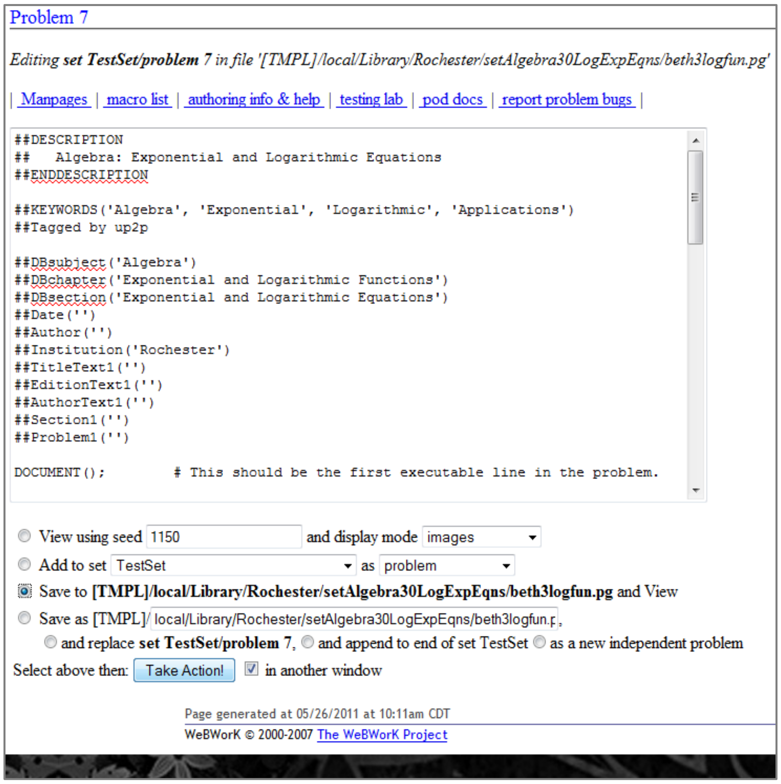
Step 5 Select the **"Edit this Problem"** link.

Step 6 Locate the unneeded section of the problem and delete it.

Step 7 The select the **“Save to *[PROBLEM NAME]* and View”** radio button. (This is the third radio button.)

Step 8 Click the **Take Action** button.

*Exhibit 2.2 Saving changes*



Problem 7

*Editing* ***set TestSet/problem*** *7 in file '[TMPL]/local/Library/Rochester/setAlgebra30LogExpEqns/beth3logfun.pg'*

| Manpages | macro list | authoring info & help | testing lab | pod docs | report problem bugs |

```
##DESCRIPTION
##   Algebra: Exponential and Logarithmic Equations
##ENDDESCRIPTION

##KEYWORDS('Algebra', 'Exponential', 'Logarithmic', 'Applications')
##Tagged by up2p

##DBsubject('Algebra')
##DBchapter('Exponential and Logarithmic Functions')
##DBsection('Exponential and Logarithmic Equations')
##Date('')
##Author('')
##Institution('Rochester')
##TitleText1('')
##EditionText1('')
##AuthorText1('')
##Section1('')
##Problem1('')

DOCUMENT();        # This should be the first executable line in the problem.
```

○ View using seed 1150 and display mode images

○ Add to set TestSet as problem

● Save to **[TMPL]/local/Library/Rochester/setAlgebra30LogExpEqns/beth3logfun.pg** and View

○ Save as [TMPL]/ local/Library/Rochester/setAlgebra30LogExpEqns/beth3logfun.p ,

○ and replace **set TestSet/problem** 7, ○ and append to end of set TestSet ○ as a new independent problem

Select above then: Take Action! ☑ in another window

Page generated at 05/26/2011 at 10:11am CDT

WeBWorK © 2000-2007 The WeBWorK Project

# Part 3: More Advanced Tweaks

## Section 3.1: Randomizing Problems

Now that you know how to delete and add parts to existing problems, a more complicated (but very useful) tweak is also available: randomizing certain numbers in problems. In a randomized problem, each student gets a slightly different problem from the rest. This way students can help one another solve problems instead of just copying answers from one homework set to another.

For instance, it is possible to modify the blank problem template so that the student can enter a value for 1, 3 or the square root of 2 instead of pi—and not only that, but each student will get a different value to enter.

What follows is a modified version of the blank problem template, with the altered parts highlighted. Explanation of how to apply this to more advanced problems follows.

```
##DESCRIPTION
##  Algebra problem: true or false for inequality
##ENDDESCRIPTION

##KEYWORDS('algebra', 'inequality', 'fraction')

## DBsubject('Algebra')
## DBchapter('Fundamentals')
## DBsection('Real Numbers')
## Date('6/3/2002')
## Author('')
## Institution('')
## TitleText1('Precalculus')
## EditionText1('3')
## AuthorText1('Stewart, Redlin, Watson')
## Section1('1.1')
## Problem1('22')

########################################################################

DOCUMENT();

loadMacros(
  "PGstandard.pl",     # Standard macros for PG language
  "MathObjects.pl",
  #"source.pl",        # allows code to be displayed on certain sites.
  #"PGcourse.pl",      # Customization file for the course
);

# Print problem number and point value (weight) for the problem
TEXT(beginproblem());

# Show which answers are correct and which ones are incorrect
```

```
$showPartialCorrectAnswers = 1;

#################################################################
#
#  Setup
#
#
Context("Numeric");
$a=random(1,3,5);
```

The line of code above will generate either a 1, a 3, or a 5 wherever $a is placed in the problem. There is no limit to the number of random numbers that can be used, but each random number generator must be labeled differently. If you wanted two random numbers in a problem, for instance, the first would be $a, and the second might be something like $b=random(2,4,6).

```
#################################################################
#
#  Text
#
#
Context()->texStrings;
BEGIN_TEXT

Enter a value for \($a\) $BR $BR
\{ans_rule(40)\}

END_TEXT
Context()->normalStrings;

#################################################################
#
#  Answers
#
#
ANS(num_cmp($a));
```

Since $a is a number, the answer should be in ANS(num_cmp()); form. If the answer were a function, it would be in ANS(fun_cmp()); form.

Step 1 Create a blank problem, or add a problem similar to the one you wish to modify to your problem set. (It will be the last problem of the problem set.)

Step 2 Select the **"Edit this Problem"** link.

Step 3 Locate the relevant sections of the problem and revise them, adding in random elements to the text of the problem itself as needed.

Step 4 The select the **"Save to *[PROBLEM NAME]* and View"** radio button. (Third radio button.)

Problems 2, 3 and 4 of the Test Set, which is a companion to this guide, all give examples of randomized problems, some with multiple randomized numbers. The problem "MoreAdvancedTweaks.pg" is the problem above.

## Section 3.2: Variables

Another useful tweak is adding variables themselves to a WebWork problem. WebWork will recognize **x** and **y** easily enough, but for problems with more than two variables, or for problems with variables that aren't **x** and **y**, a different method must be adopted. Fortunately, this method is far simpler than randomizing.

The problem given below is two-part. The first part of the problem asks for a number, the second, for a generalized formula involving the variable **c**. Writing these variables into the problem does not pose significant challenges, but defining the answer in terms of **c** can:

```
##DESCRIPTION
##  Algebra problem: true or false for inequality
##ENDDESCRIPTION

##KEYWORDS('algebra', 'inequality', 'fraction')

## DBsubject('Algebra')
## DBchapter('Fundamentals')
## DBsection('Real Numbers')
## Date('6/3/2002')
## Author('')
## Institution('')
## TitleText1('Precalculus')
## EditionText1('3')
## AuthorText1('Stewart, Redlin, Watson')
## Section1('1.1')
## Problem1('22')

########################################################################

DOCUMENT();

loadMacros(
  "PGstandard.pl",     # Standard macros for PG language
  "MathObjects.pl",
  #"source.pl",        # allows code to be displayed on certain sites.
  #"PGcourse.pl",      # Customization file for the course
);

TEXT(beginproblem());
$showPartialCorrectAnswers = 1;
```

```
TEXT(EV2(<<EOT));

What is the average value of the function \(f(x)=7\) for \(x\) greater than or equal to \(-2\) and less than or equal to \(3\)? \{ ans_rule(35) \}  $BR $BR
What is the average value of a constant function \(f(x)=c\) on the interval \([a,b]\)?

EOT
$ans = 7;
ANS(num_cmp($ans));

TEXT(EV2(<<EOT));
What is the average value of a constant function \(f(x)=c\) on the interval \([a,b]\)? \{ ans_rule(35) \}
EOT
ANS(fun_cmp( "c", 'var'=>['c'] ) );

ENDDOCUMENT();
```

This problem is in the Test Set and is named "Defining Variables.pg." (This is #5 in the Test Set.) Any number of variables may be added by tweaking the code very slightly. If you wanted an answer involving **a**, **x**, and **c**, for example, all you'd need to do would be to add these to the 'var': ANS(fun_cmp( "**a**+**x**+**c**", 'var'=>['**a**','**x**','**c**'] ) ).

## Section 3.3: Helpful coding tips

The following is an excerpt from the WebWork Newbie Guide. The link to this guide can be found below.

1. $showPartialCorrectAnswers = 0;
   - This corresponds to not revealing which of the answers may be correct. For example, if the student answers correctly two of the four questions, WebWork will respond by saying the student has 50% of the parts of the problem correct. If you set $showPartialCorrectAnswers = 1;, WebWork will indicate which of the four answers are correct.
2. $a = random(-5,5,1);
   a. random(begin, end, incr) produces a pseudo-random number between <begin> and <end> in increments of <incr>. The default increment is 1, so the statement $a = random(-5, 5, 1) can be abbreviated $a = random(-5, 5);
3. BEGIN_TEXT ... END_TEXT
   - Before we get going, note that comments are *not* allowed between the BEGIN_TEXT and END_TEXT delimiters, so place your comments outside. Anything you write in there will be rendered on the screen for students to see. You can have multiple BEGIN_TEXT and END_TEXT blocks within a program, so this is not a significant limitation.

- Now what appears between these two delimiters is meant to be rendered for the student in a number of ways. In general, this code is interpreted and a LaTeX document is produced. For students who select to receive a hard copy, this document is TeXed and either a postscript or .pdf output is produced for the student to view and print.
- Alternatively, the LaTeX document is run through LaTeX2HTML to produce a nice looking web page on which the student can actually do the homework problem.
- What you write in the text blocks will become either LaTeX code or HTML code. Thus, the code you write here must be translated into the appropriate language, and you must learn to use certain generic control sequences to achieve your goals. Thus for example, a paragraph break in LaTeX is denoted \par, while in
- HTML it is denoted <P>. In the .pg language of WebWork it is denoted $PAR which gets translated as appropriate. A list of control sequences is found in PGbasicmacros.pl. They include:
    - $PAR — paragraph break
    - $BR — line break
    - $LQ, $RQ — left and right quotes: ‘‘ and ’’
    - \( \) — begin and end math mode as in \( f(x) = \sin(x^2) \)
    - \[ \] — begin and end display math mode
    - $LTS, $GTS, $LTE, $GTE — ¡, ¿, ¡=, ¿= (inequalities)
    - $US — underscore (without this LaTeX wants to be in math mode)
    - $SPACE — a space (\ in LaTeX, &nbsp; in HTML)
    - $BBOLD, $EBOLD — begin and end bold face type
    - $HR — horizontal rule across the page
    - $LBRACE, $RBRACE — left and right braces: {, }
    - $LB, $RB — synonymous with $LBRACE, $RBRACE

4. \{ans_rule(10)\}
    - This is clearly a very important construction since it is how you provide a blank into which students type their answers. The length should provide adequate space for the answer you expect. Be mindful: if you type ans_rule(10) all that will be displayed is the text ‘‘ans_rule(10)’’. You have to bracket the construct with (escaped) braces ( \{ans_rule(10)\} ) so that the evaluation routine (EV3) interprets it as an answer box.
5. The construct $a % $b gives the integer remainder of dividing $a by $b, with the remainder in the interval [0, $b-1].
6. ANS(std_str_cmp($ans));
    - This is the standard answer evaluator that handles strings. It is case insensitive and any white space is treated as a single space. For example, "Apache really rocks" will be treated the same as "APACHE ReAlLY ROCKS." It is the most common string function invoked. You can peruse .../system/courseScripts/PGanswermacros.pl for more details.
7. ANS(std_num_cmp(($m/$n)*(($b - $a)**($m/$n - 1))));
   ANS(std_num_cmp($b));
   ANS(std_num_cmp(($b - $a)**($m/$n)));

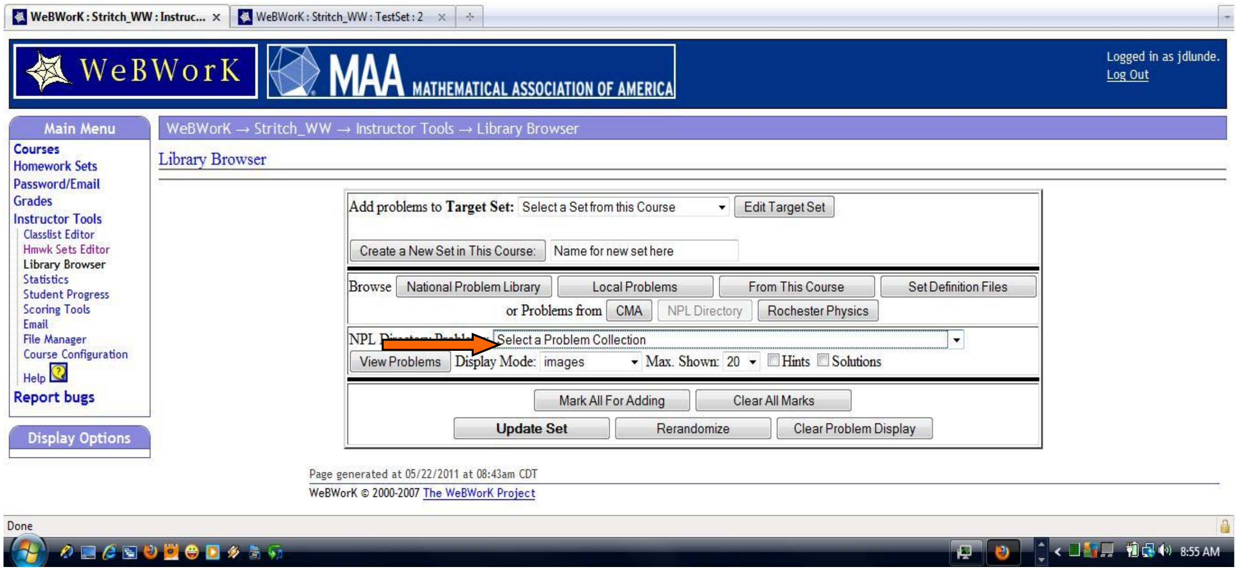
- These are the standard answer evaluators that handle numeric answers. These compare the answer the student has provided with the correct one and accept the student answer as correct if it is within .1% of the right answer. You can change the tolerances which WebWork will use, which is quite useful if your answers are very large or very small. These answer evaluators will also accept simple functions as part of the answer.
  - Example: If the answer is 1, then exp(0) would be an acceptable answer. You can also require answers in a more restrictive form.

It is very important that in WebWork code, you use ** and not ^ to denote exponents. Students may use either in answering a question, but Perl wants the ** in the code of the problem (Shemanske, 2002).

## Part 4: Documentation

There are many additional resources available for teachers and authors of problems. For teachers, probably the most useful resource is the National Problem Library (NPL), a searchable database which collects many thousands of problems for instructor use.

To access the NPL, click on **Library Browser** on the left sidebar of WebWork. From there, it is easy to select an existing problem set from one of the drop-down menus available under **Select a Problem Collection**.



Logged in as jdlunde.

WeBWorK → Stritch_WW → Instructor Tools → Library Browser

Library Browser

Add problems to **Target Set:** Select a Set from this Course

Name for new set here

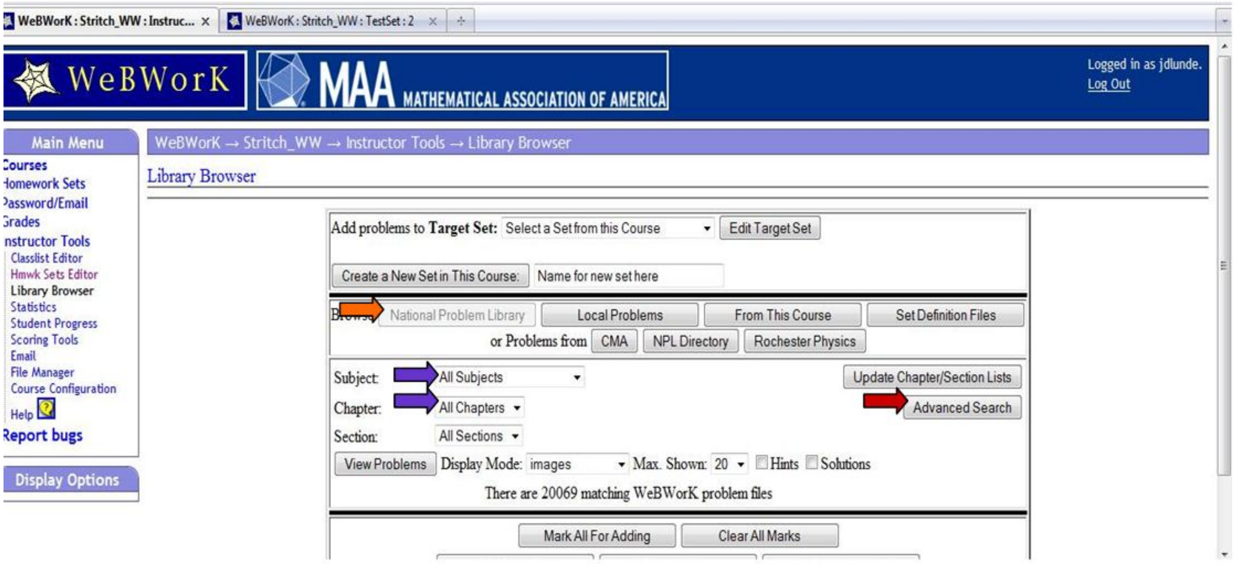
NPL Directory Problems: Select a Problem Collection

Display Mode: images  Max. Shown: 20

Page generated at 05/22/2011 at 08:43am CDT

WeBWorK © 2000-2007 The WeBWorK Project

To search for a specific problem, return to the **National Problem Library** screen, and click on **Advanced Search**. Problems can also be searched for on the basis of a **subject** or a **chapter** of a specific text.



Logged in as jdlunde.

WeBWorK → Stritch_WW → Instructor Tools → Library Browser

Library Browser

Add problems to **Target Set:** Select a Set from this Course

Name for new set here

Subject: All Subjects

Chapter: All Chapters

Section: All Sections

Display Mode: images  Max. Shown: 20

There are 20069 matching WeBWorK problem files

Problems can then be added to sets as usual.

Be aware that the search function for WeBWorK only allows the searching of one-word concepts, since it adds an "and" between terms. Searching for "Taylor Polynomials," for instance, generates no results, but searching for "Taylor" finds over 30, and searching for "Polynomials" finds over 150 results, as of this writing.

WebWork also has extensive documentation meant to help instructors write more complex problems, including problems involving graphs and sophisticated multipart problems. The documentation for WebWork may be found here: http://WebWork.maa.org/documentation.html.

## Course hosting and access to CMA and Test Set

The first FAQ on the documentation page is “How can I apply for hosting?”. Follow that link to another, “Requests for hosting…”, which brings up an online request form. There is a free-response box on the form for “additional information” – be sure to notify MAA that you are using the Calculus: Modeling and Application text and would like your course to have access to the CMA Library. In the drop-down for CMA, the first entry is Set0_Test_Set, where you will find the sample problems refered to above.

## Useful links

WebWork Newbie Guide

Basic syntax for Perl

Write/modify Problems The tutorial link on this page is no long being maintained but this information still can be useful. The new link is: Learning how to Author WebWork problems

## Work Cited

Shemanske, T. R. (2002, July 12). *WebWork Newbie Guide.* Retrieved May 24, 2001, from WebWork Newbie Guide: http://www.math.dartmouth.edu/~trs/expository-papers/tex/newbie-1.7/WebWork_newbie.pdf