

# Using WeBWorK

## A Short Introduction

Shaun Ault

**What is WeBWorK?** WeBWorK is a free and open-source online homework delivery system maintained by the Mathematical Association of America (MAA). <http://webwork.maa.org/>

Instructors using WeBWorK may find problems on the Open Problem Library (OPL), or have access to custom problems available for the course, or even write/modify their own problems. Generally each problem has some random parameter(s) built into it so that different students will see slightly different problems.

- **OPL.** The Open Problem Library is categorized by subject, chapter, and section (for example, *Calculus – single variable; Differentiation; Definition of the Derivative*).
- **Local Problems.** These are problems that have already been created by a problem author for use in your courses. For example, with the support of an Affordable Learning Georgia grant (ALG), over 300 problems were created for use in the no-cost version of Analytic Geometry & Calculus I at Valdosta State University. Instructors of this course create homework sets from this problem bank in their own sections.
- **Creating your own problems.** Not recommended for the typical instructor. Custom problems can be created by coding them in PGML (problem generation markup language) / PERL. Mathematical expressions are written in a basic form of LaTeX embedding with the PGML.

### Getting Started:

1. Go to <https://webwork01.valdosta.edu/webwork2/myTestCourse/> and login with the credentials: testuser / webwork01
2. Click **Library Browser** (on left side, under **Instructor Tools**).
3. Type a name for a new set, *without spaces* (note, you may use the underscore \_ to indicate spaces), and then click **Create a New Set in This Course**. Notice that your set name is now in the box after “Add problems to **Target Set:**”.
4. Explore the Open Problem Library. After choosing the Subject, Chapter, and Section, click **View Problems**. If you want to add any to your test set, then click **Add** for each problem, or **Add All**.
5. The tiny pencil icon on the upper right of each problem takes you to the **edit** page, where you can view and modify the PGML code for the problem.
6. Click the tiny eye icon to try out the problem as a student would see it.

### Local Problems:

1. In the **Library Browser**, instead of using OPL problems, you may have local problems specific to your course/institution. Click on **Local Problems** to see a problem bank created specifically for VSU no-cost Calculus.
2. Click the drop-down menu labeled, *Local Problems: Select a Problem Collection*. Select a section and click **View Problems**.
3. All local problems are stored in the **File Manager**. Additional problems written locally must be uploaded to a folder in the File Manager before they can be accessed in WeBWorK. Often, a designated WeBWorK Administrator in your department will handle this part.

### Modifying and Writing Problems (not for the faint of heart):

1. It helps to copy an existing problem that is close to the one you want to write.
2. Note, problems written or edited within one course are not visible to other courses. Best practice is to maintain the problems as separate files outside of the WeBWorK environment.
3. Typical problem development proceeds as follows:
  - Copy an existing problem to a new file, ExerciseXXX.pg (identifying the problem appropriately in the filename – remember, *no spaces in any names*).
  - Edit the file using any text editor.
  - From your test course on WeBWorK, go to **File Manager**, find or create the folder into which you want to put this problem.
  - Click **Browse** to find the file, and then click **Upload**. You may wish to check the box, *Overwrite existing files silently*, especially as you intend to work on the same problem and upload it many times.
  - Go back to the **Library Browser** (I generally have this pulled up in a separate tab or window), find the Local Problems folder containing your problem, and click **View Problems**. Later, if you make changes and re-upload your problem file, then you can click View Problems again to refresh the list with the current version of the problem(s).
  - Click the tiny icon of two crossing arrows (upper right of each problem) in order to randomize the problem. It's best to click this a number of times. Sometimes a particular choice of the random parameters causes an error that you would need to address before students see the problem (*Rule of Thumb*: If some choice of random parameters causes an error, then at least one student *will* find that error and complain about it!)
  - Click on the tiny eye icon (upper right of each problem) to try it out. This is the main testing ground to verify that you've formatted the problem correctly and that it works as expected (i.e., correct answers are indeed accepted as correct). Robust testing of the problem includes entering many incorrect answers as well.
  - Once you are satisfied with the problems, you may save them all to a single file by clicking **Make Archive** in the **File Manager**. Then archives can be downloaded, shared to others, and uploaded into another WeBWorK course.

### Typical Problem Structure:

Problems written in PGML consist of a few distinct parts, as shown below.

```
#Problem APEX 2.3.12
```

Commented line with information about the problem (optional)

```
DOCUMENT();
```

Start of the problem file.

```
# Load whatever macros you need for the problem
```

Macros are necessary for certain features. More information about what each macro does can be found on the MAA WeBWorK website.

```
loadMacros(  
  "PG.pl",  
  "PGbasicmacros.pl",  
  "PGchoicemacros.pl",  
  "PGanswermacros.pl",  
  "PGauxiliaryFunctions.pl",  
  "MathObjects.pl"  
);
```

```

## DBsubject(Calculus - single variable)
## DBchapter(Differentiation)
## DBsection(Derivatives involving
multiple rules (no product, quotient, or
chain rule))
## Institution('Valdosta State
University')
## Author('S. V. Ault')
## TitleText1('APEX Calculus')
## AuthorText1('Hartman')
## EditionText1('3.0')
## Section1('2.3')
## Problem1('12')

```

These lines provide database information. Most are optional. The only required fields are DBsubject, DBchapter, and DBsection.

```

$showPartialCorrectAnswers = 1;

$a3 = random(2,20,1)*random(-1,1,2);
$a2 = random(2,20,1)*random(-1,1,2);
$a1 = random(2,20,1)*random(-1,1,2);
$a0 = random(1,20,1)*random(-1,1,2);

```

The start of the problem code. In this section, variables are set that affect the problem behavior and display. All random parameters should be defined here.

```

$b2 = 3*$a3;
$b1 = 2*$a2;
$b0 = $a1;

```

```

$sans = "$b2 x^2 + $b1 x + $b0";

```

```

TEXT(beginproblem());

```

The start of the displayed problem text.

```

BEGIN_TEXT
Compute the derivative of the given
function.
$PAR
\[
g(x) = $a3 x^3 + $a2 x^2 + $a1 x + $a0
\]
$PAR
\{g'(x) = \}\{ ans_rule(15) \}

```

Everything between BEGIN\_TEXT and END\_TEXT will be interpreted as part of the problem statement. A number of markup commands are available (such as \$PAR for paragraph). Mathematical expressions can be written in LaTeX, using  $(\ )$  for inline, and  $\[ \]$  for centered expressions.

```

END_TEXT

```

Answer fields are coded using ans\_rule as shown (there are also other kinds of answer formats, including radio buttons, dropdown lists, etc.).

```

ANS( fun_cmp( $ans ));

```

The ANS command generally handles answers checking.

```

SOLUTION(EV3(<<'END_SOLUTION'));
$BR$BBOLD Solution:$EBOLD
$PAR
Using the Sum/Difference Rule and Power
Rule,
\[
g'(x) = $a3(3x^2) + $a2(2x) + $a1 x + 0
= $ans.
\]
END_SOLUTION

```

(Optional) Solutions can be provided to students using the SOLUTION / END\_SOLUTION environment. Typically a student cannot see the solutions until after the due date for the problem set has passed.

```

ENDDOCUMENT();

```

Ends the problem file.