WEBWORK PROBLEM AUTHORING GUIDE FOR WPI ACTUARIAL SCIENCES

Leah Navickis

WORCESTER POLYTECHNIC INSTITUTE

Table of Contents

Table of Figures

Context

My name is Leah Navickis and I am an Actuarial Mathematics major and member of the WPI class of 2019. I created this document as part of my Major Qualifying Project. Many actuarial sciences students take actuarial exams during their time at WPI. WPI actuarial faculty provide many exam study resources for these students, however they expressed the need for an online exam study tool.

The purpose of this project is to begin the process of creating such an online resource in the form of a database of WeBWorK question sets, as well as a Canvas site. I created a pilot resource for the FM exam, one of the first actuarial exams students typically take. This pilot resource will also be used in the Theory of Interest courses offered at WPI. Since this is meant to be a study tool, solutions are always available to the students, so some of the techniques used may not apply to graded courses. This document serves as a guide of how to use WeBWorK, in order to ease the process of creating similar resources for other exams.

1 About WeBWorK

1.1 Purpose of WeBWorK

Simply put, WebWorK is an online resource where professors can create and assign homework sets and students can complete those homework sets. The system can help professors organize and manage their courses, as well as save them time since the problems are graded automatically. Additionally, WeBWorK is open source, so it serves as a space where professors and other contributors can share problems.

1.2 Programing Language

The programing language used to write WeBWorK problems is called PG, which stands for “Problem Generating” (Problem Authoring Background Information, 2015). The Problem Generating syntax is written in Perl, and takes some practice. I have found that the easiest way to write problems is to start with an existing problem template, so that you do not have to write problems from scratch. Examples of the PG language can easily be found online, and I will discuss multiple examples in Section 5.

1.3 WeBWorK Resources

There are many resources for the WeBWorK community to use. In this section, I will discuss the ways I found useful for both learning how to use WeBWorK and writing new problem types including:

1. The WeBWorK Website
2. The WeBWorK Forum
3. The Open Problem Library (OPL)

1.3.1 The WeBWorK Website

There are multiple web pages on webwork.maa.org/wiki that contain valuable information (Main Page: WeBWorK Doccumention Wiki, 2019). You should explore the website before writing problems, and start your search for questions there.

1.3.2 The WeBWorK Forum

The WeBWorK forums are excellent for trouble shooting problems in either your code or in the WeBWorK site. You can find questions people have asked, or post your own question

(Using WeBWorK: General Forums, n.d.). Some of the questions are answered by regular users or by WeBWorK developers.

1.3.3 The Open Problem Library (OPL)

The most useful way to learn how to write WeBWorK problems is to look at code from similar existing problems. Luckily, thousands of problems are freely available to view and use on your WeBWorK course website. You can view problems by logging into your WeBWorK course, and navigating to the library browser page. This is shown later on in Figure 14, but you will want to select "Open Problem Library" instead of "Local Problems" next to "Browse".

1.3 WPI and WeBWorK

WPI is a participating institution in WeBWorK. Many WPI professors use WeBWorK for introductory math and science courses, utilizing freely available homework problems already on WeBWorK. However, not many WPI professors write WeBWorK problems. The purpose of this guide is to serve as a reference for members of the WPI actuarial community in writing WeBWorK problems.

Currently (April 2019), WPI has their own WeBWorK server. You can verify this on the "WeBWorK Sites" page on webwork.maa.org (WeBWorK Sites, 2016). The system administrator is responsible for managing WPI's server. The WeBWorK administrator can make new courses and assign instructors to them, which is why it is important to contact them at the beginning of your project. Instructors have the ability to manage their particular courses, including adding students to the course, creating problem sets, assigning problem sets, and uploading their own WeBWorK problems (Instructors, 2016). So, if you are writing problems, you will need to ask the administrator to create a course for you and register you as an instructor. Students have the ability to complete homework sets assigned to them.

2 Locally Uploading a Problem to WeBWorK

There are multiple ways to write and upload your WeBWorK problems to your local problem database, but in this section I will guide you through the method I found most intuitive and useful. Figure 1 shows a flowchart of the process of uploading a problem to WeBWorK.

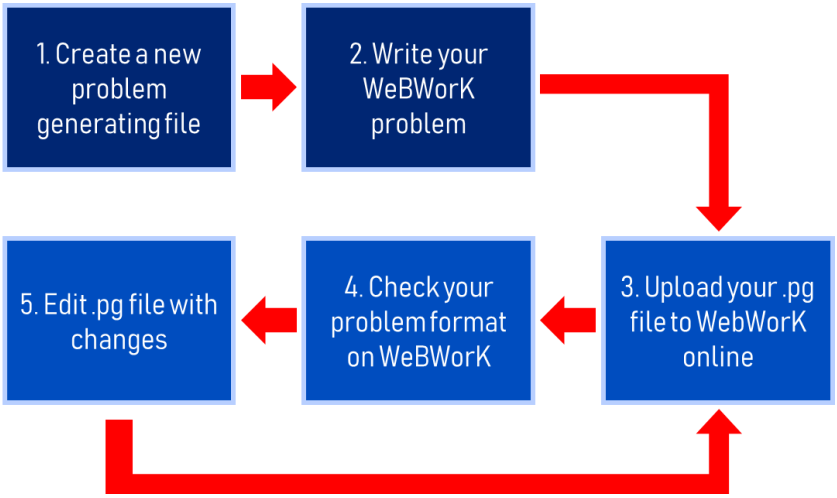


Figure 1 Flow chart depicting the process of uploading a problem to WeBWorK.

2.1 Create a new problem generating file

The first step is to create a text file where you will write your WeBWorK problem. Each WeBWorK problem exists in its own text file, so it is important to use a naming convention and keep your problem files organized. First, open the Notepad application on your computer, or any application that allows you to save a file as a text file. Once you have created a new text file, name and save your file as a .pg file, so that WeBWorK will read it as a problem file. "Pg" stands for "problem generating". In order to do this, press "file" then "Save as" at the top left of the window. Next to "Save as type", select "All files". Then, type your problem name followed by .pg in the "File name" box. The process of saving a file as a problem generation file is shown in Figure 2.

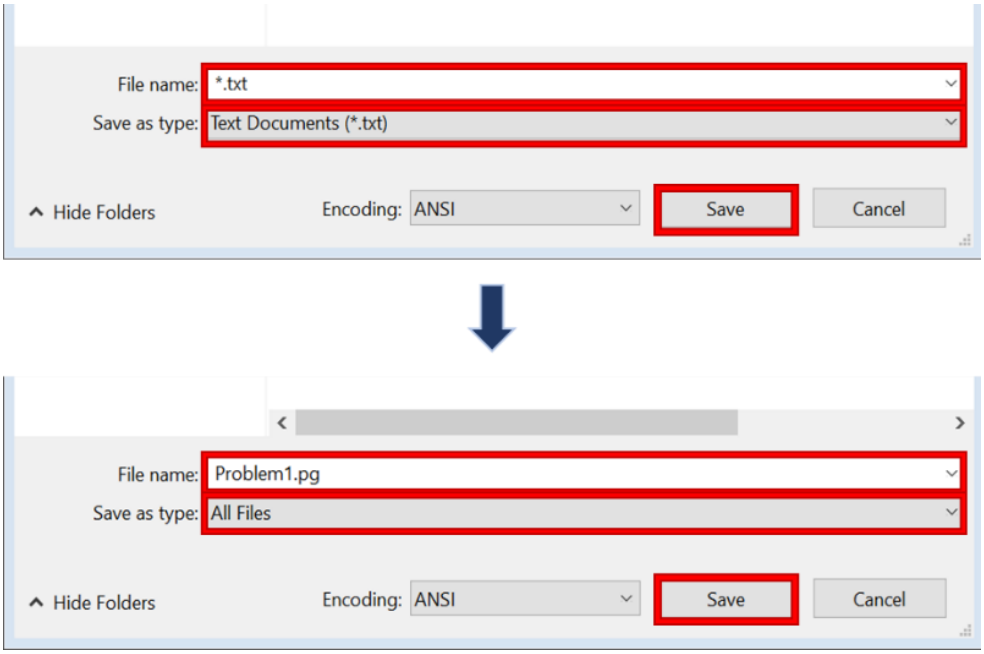


Figure 2 Name your problem generating file.

2.2 Write your WeBWorK problem

You can now write your WeBWorK problem by typing directly in your new file, and making sure you save the file afterwards. It is also recommended to format your code in a readable way. I discuss how to author a WeBWorK problem in detail in Section 4 Writing WeBWorK Problems.

2.3 Upload your .pg file to WeBWorK online

Now you can upload your new WeBWorK file to the file manager in your WeBWorK course. In order to do this, first open your web browser, and navigate to your WeBWorK course, and login. The left half of Figure 3 shows that I am choosing my "FMExam" course, however you want to choose whatever course you want to upload your problem to. The right side of Figure 3 shows where to find the "file manager" tab in your course. If you do not have a "file manager" tab, check to make sure you are registered as an instructor for the course.

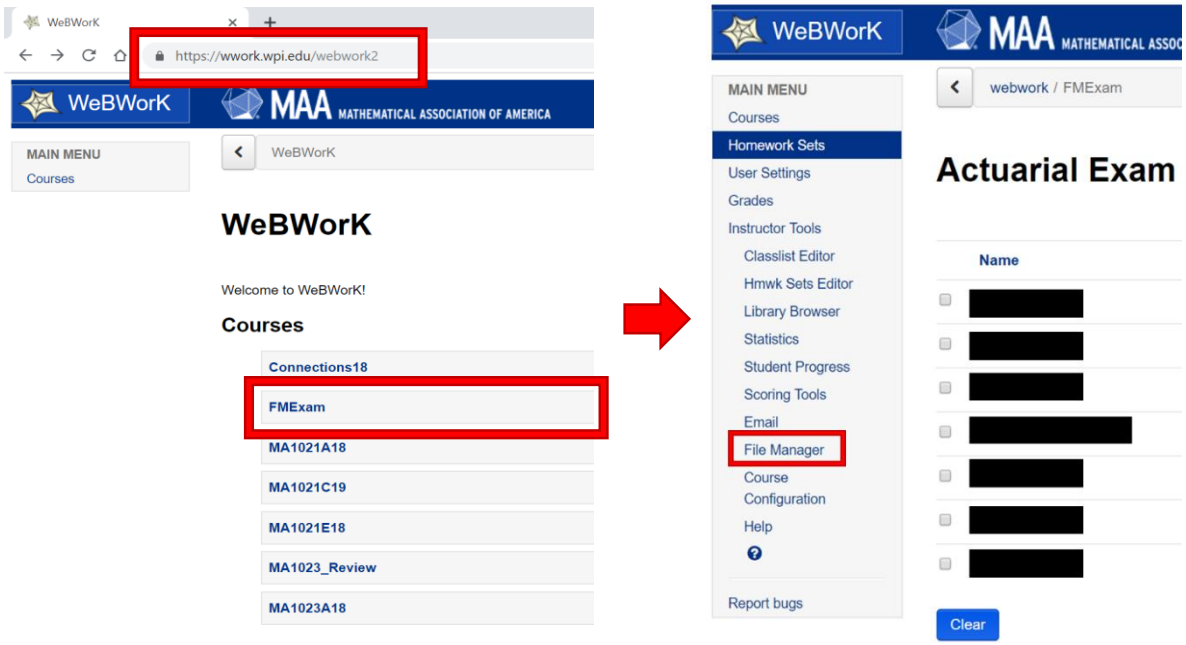


Figure 3 Navigate to file manager.

Next, you must create a new folder to put your file in, so that WeBWorK will have a location to retrieve your file from. In order to do this, press the "New Folder" button shown in Figure 5, and name your new folder as shown in Figure 4. Do not forget to press "New Folder" in Figure 4, otherwise your folder will not be created.

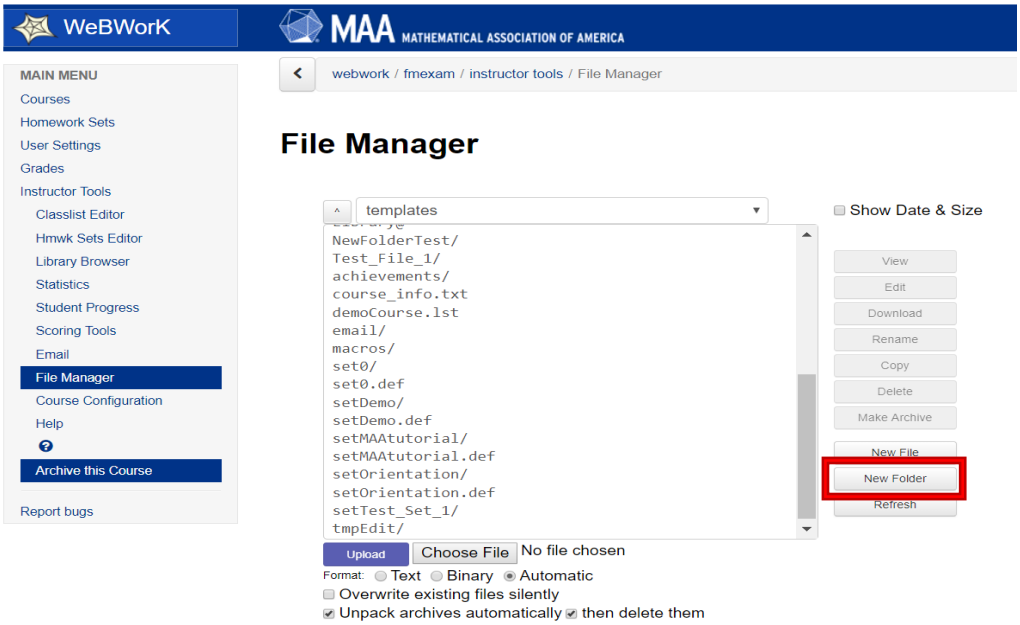


Figure 4 Create a new folder.

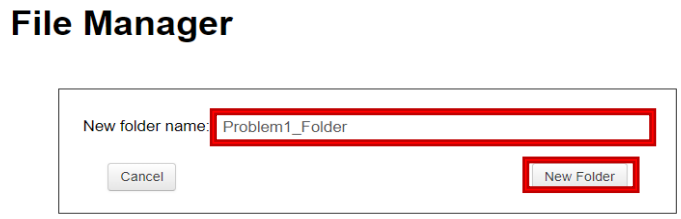


Figure 5 Name your new folder.

Now you may begin the process of uploading your problem. Press “Choose File” as shown in Figure 6.

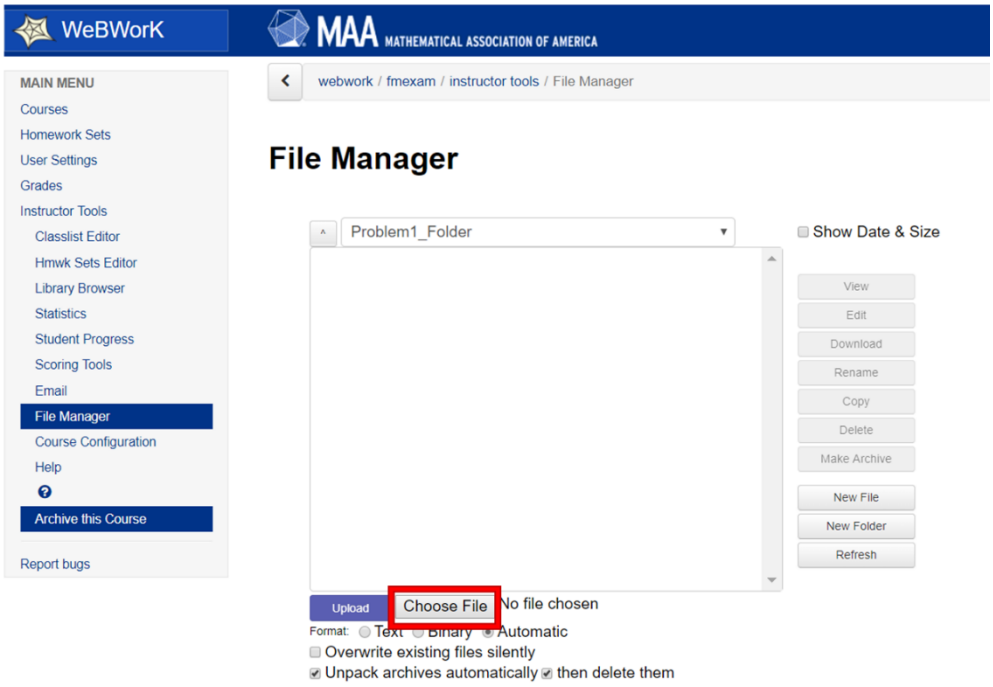


Figure 6 Select “Choose File”.

After pressing this button, a file window will appear, as shown in Figure 7. Now navigate through your files, find your problem, select it, and press “Open”. These actions are also shown in Figure 8Figure 7.

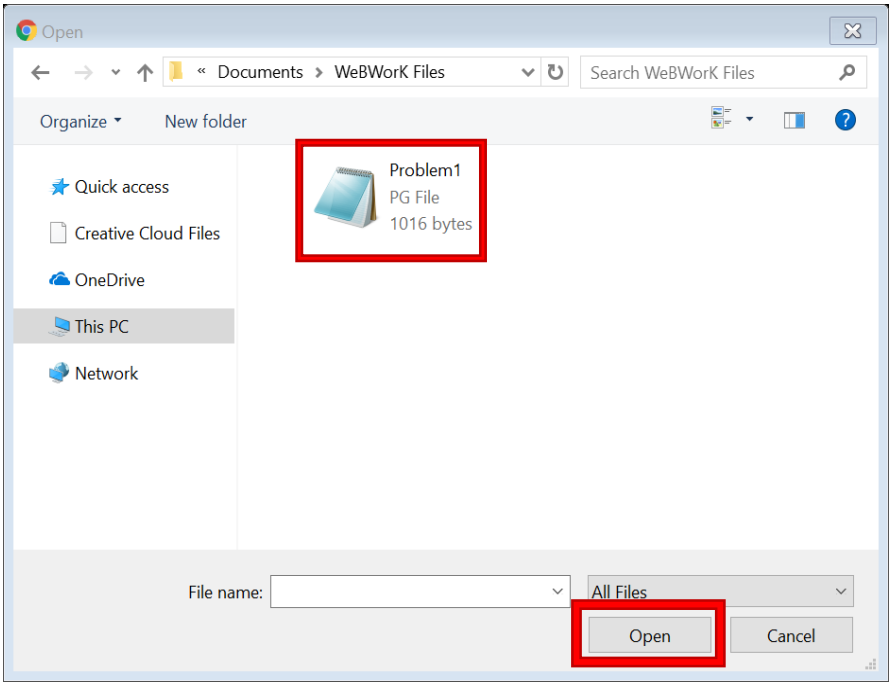


Figure 7 Select your PG file.

After pressing “Open”, the file window should close automatically, bringing you back to the File Manager screen shown in Figure 8. Make sure your problem file name now appears next to the “Choose File” button, indicated by the arrow in Figure 8, then press “Upload”. The settings at the bottom of the screen should not need to be changed.

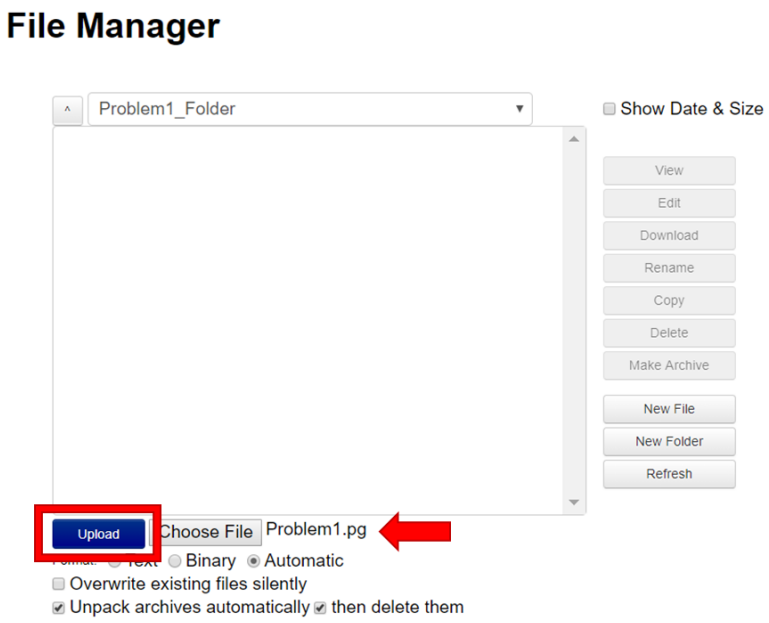


Figure 8 Upload your PG file.

Be sure to check that your problem was uploaded into the correct folder, and that you get a “File uploaded successfully” message such as in Figure 9. Any images incorporated into the question should also be uploaded into the same folder as the question. Once you get the hang of writing questions, you will find it more convenient to upload multiple problems in this step, so that you may proof-read a bunch of problems at once.

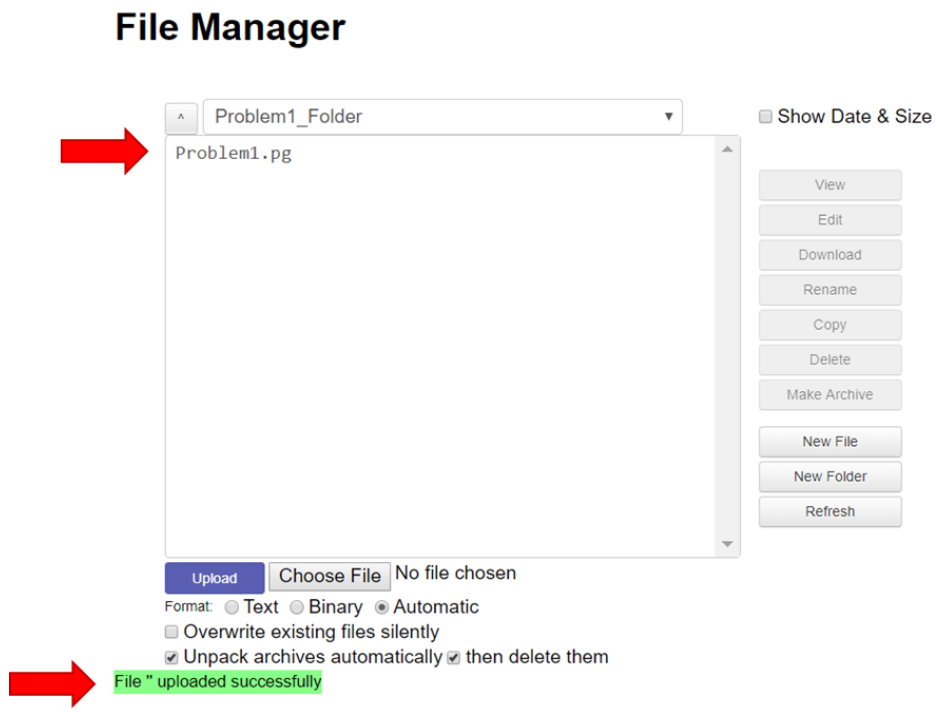


Figure 9 Check that your problem wsa uploaded.

2.4 Check your problem format on WeBWorK

Since it is likely that your first few problems will have some mistakes in them, you must now view your problem in WeBWorK. To check if you coded your problem correctly, you need to first create a blank problem set. This can be done by navigating to the “Hmwk Sets Editor” Tab on the left of the window, and pressing “Create”. These two buttons are shown in Figure 10.

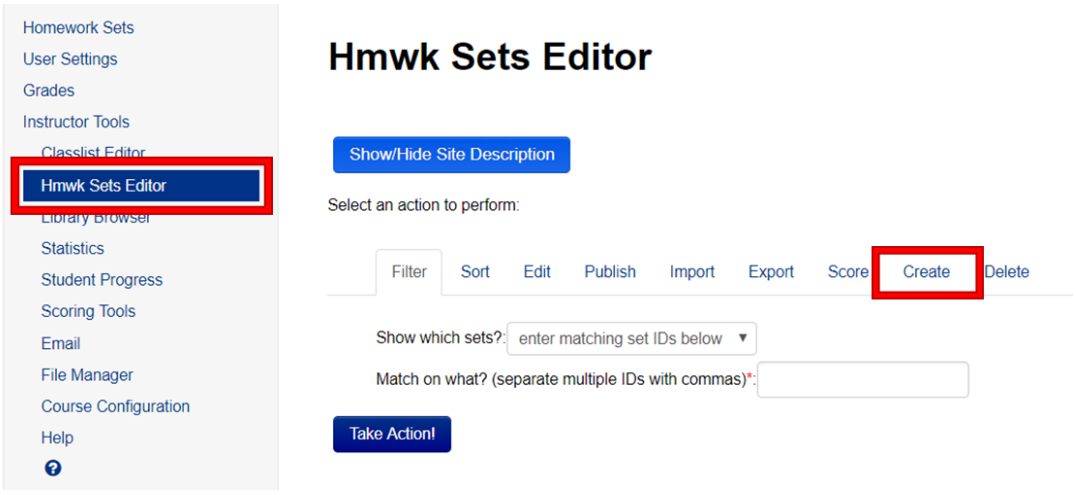


Figure 10 Create a new problem set.

When you press create, the screen should look like Figure 11. You can now select that you want to create a new empty set and the name of the new set. To continue, press “Take Action!”

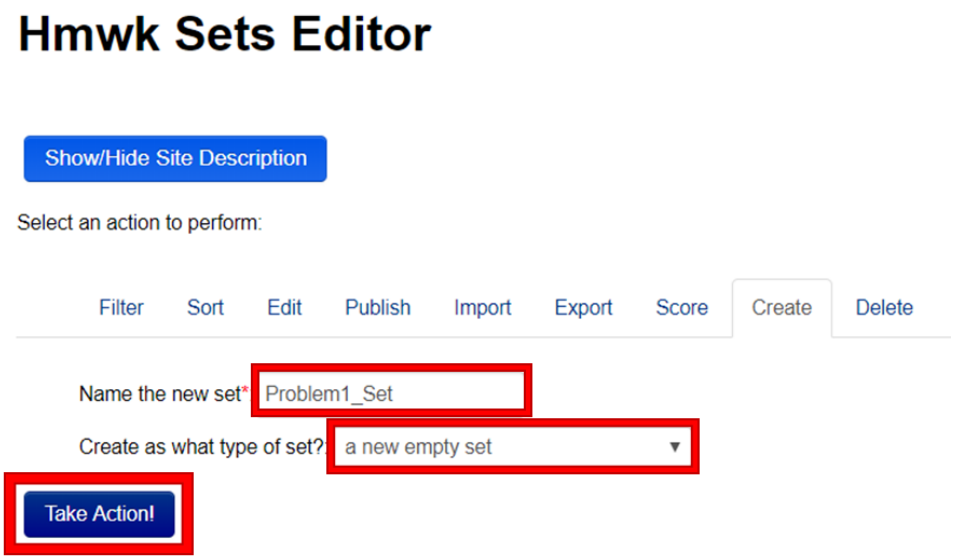


Figure 11 Name your new problem set.

You should then be taken to a different screen with confirmation notifications on the bottom as shown in Figure 12.

Figure 12 Check that your problem set was created.

Next, we can finally view the problem by pressing Library browser in the left-hand menu, as shown in Figure 13.

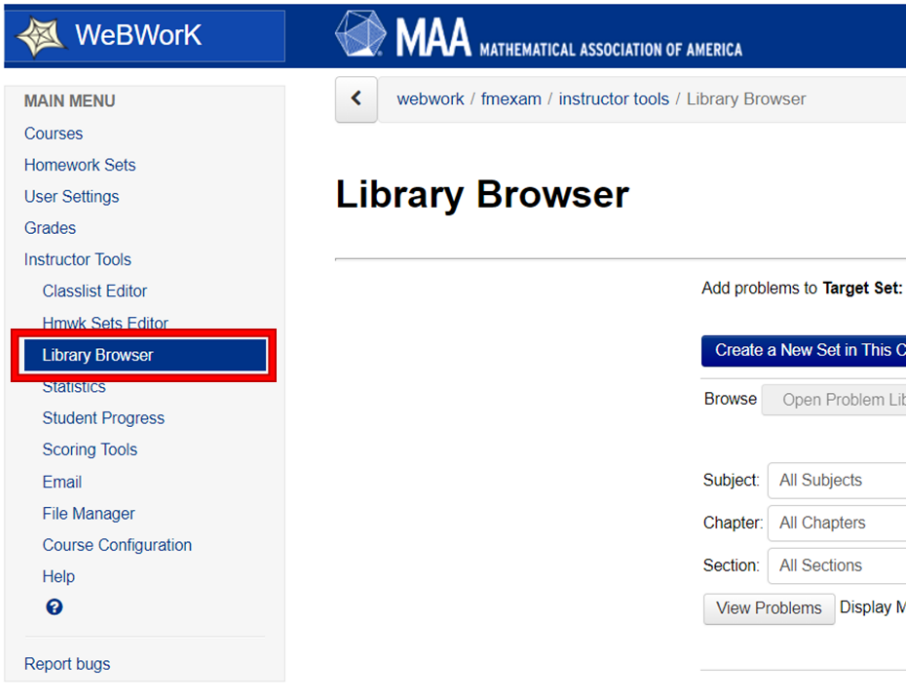


Figure 13 navigate to "Library Browser".

Now, in order to view your problem, you must pretend that you want to add your new problem to the problem set you created in Figure 11. So, you must select the correct options on the Library Browser page, as shown in Figure 14. Next to "Target Set", select the new problem set you created. Then, ensure "Local Problems" is selected next to "Browse", because your problem is stored locally in your course. Next to "Local Problems" select the folder that your problem is in, and finally press "View Problems". If your problem is written correctly, it should appear on the screen, as shown in Figure 14.

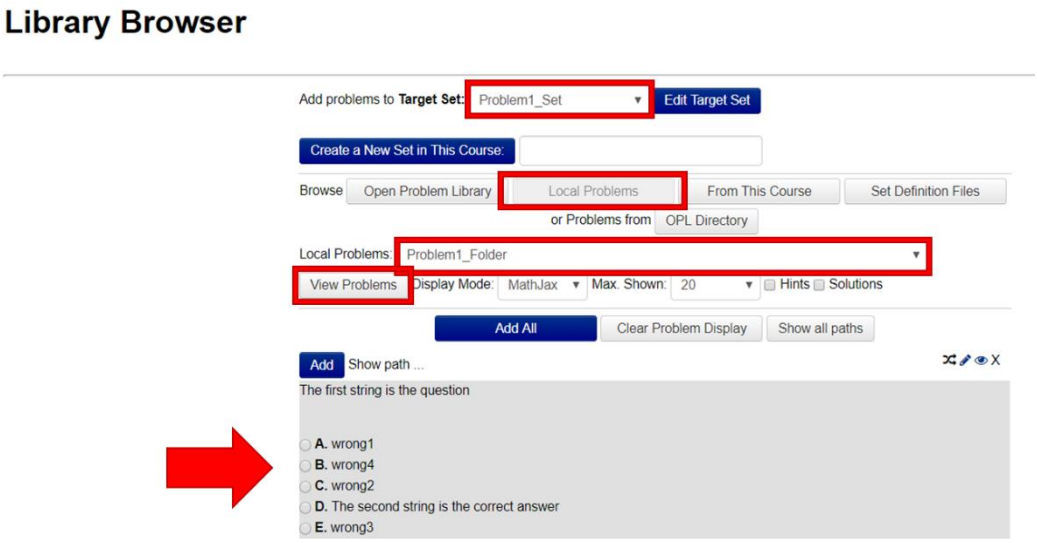


Figure 14 View your new problem on WeBWorK in the "Library Browser".

If you have uploaded multiple problems in the folder, all of the problems should appear here as well. If you are satisfied with your problem, you can add it to the problem set by pressing the "Add" button as shown in Figure 15.

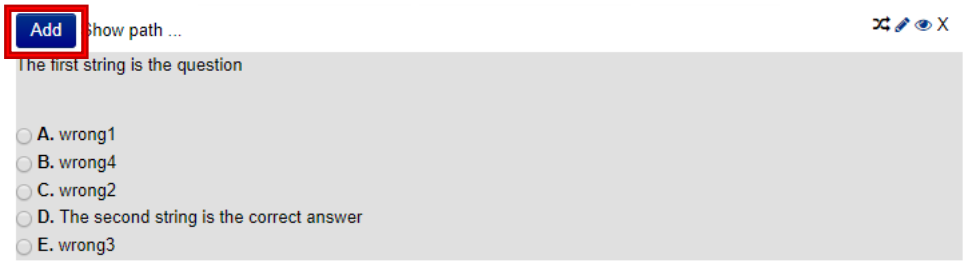


Figure 15 Add your problem to your problem set.

2.5 Edit .pg file with changes

Unfortunately, there are a lot of potential mistakes you could have made in your WeBWorK code. This may cause many formatting issues, including code showing up in your problem, images not being displayed properly, or even producing an error message, as shown in Figure 16.

Figure 16 Potential error message

In order to correct your mistakes, you must edit your .pg files directly on your device, then re-upload them to your online folder. In other words, repeat the steps shown in Figure 6 through Figure 14, with the exception that you will not have to create a new problem set this time. Then, proof-read your problems once again, and repeat this process until your problems appear how you want them to. Once you are satisfied with your problems, you can then add them to homework sets by pressing the "Add" button, as I mentioned in Figure 15.

3 Assigning Problem Sets to Students

3.1 Viewing Problem Sets

In Figures 10-12, we created a problem set, and added a problem to it in Figures 13-15. You can view the problem set you created by clicking on the name of your problem set under "Hmwk Sets Editor", as shown in **Error! Reference source not found.** You can also view Homework sets under the "Homework Sets" tab, but you cannot edit. Figure 17 also highlights the three main categories you can edit: Set Dates, Problems, and Assigned Users.

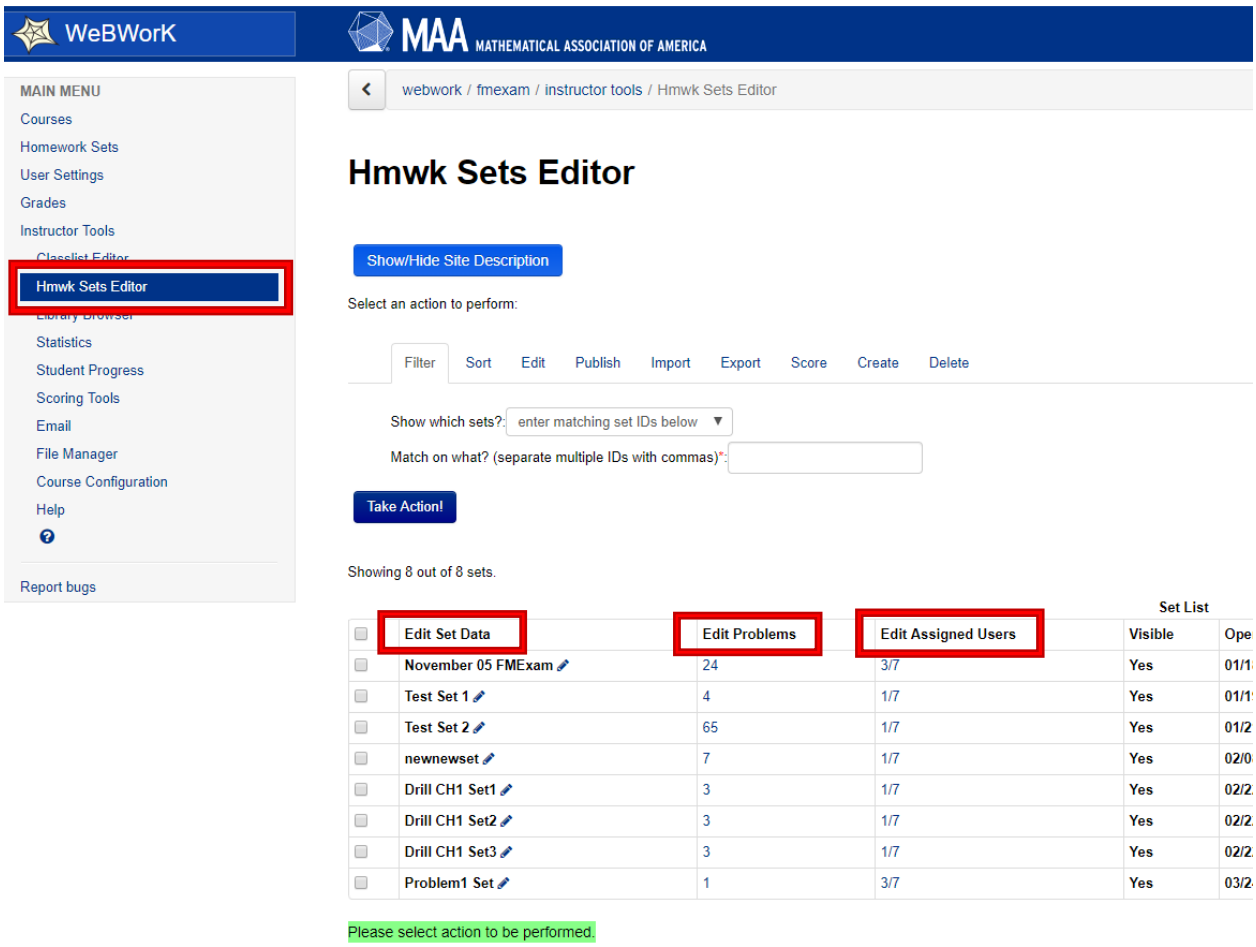


Figure 17 Navigate to "Hmwk Sets Editor".

3.2 Edit Set Dates

To set the due dates corresponding to your set, first press the pencil next to the corresponding set name as indicated in Figure 18.

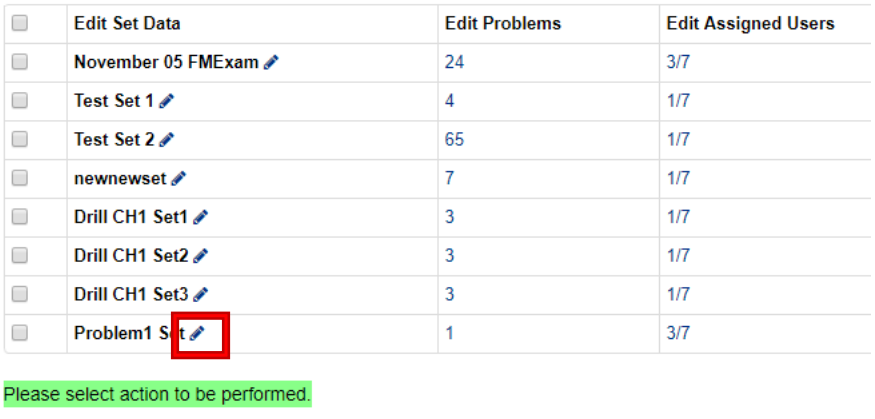


Figure 18 Select pencil icon to choose due dates.

The next set of steps are shown in Figure 19. In order to input the dates and times, you can manually type the date and times in the text boxes, or you can press the calendar buttons and choose the date that way. First, choose the "Open Date", which determines when the students will be able to start working on the problem set. Then choose the "Close Date" which is effectively the due date, and then the "Answer Date" determines when the students will be able to view the solutions. Be sure to press "Done" after choosing dates by using the calendar buttons. Finally, press "Take Action!" in order to save your work.

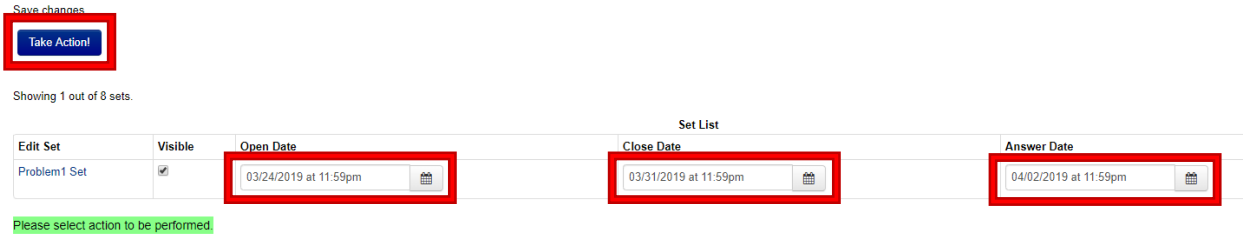


Figure 19 Choose "Open Date", "Close Date", and "Answer Date" for your peoblem set.

3.3 Edit Problems

To edit problems, begin from the "Hmwk Sets Editor" tab again, but this time press the number that appears in the "Edit Problems" column of the corresponding set's row. This number, shown in **Error! Reference source not found.**, represents the number of problems in the c

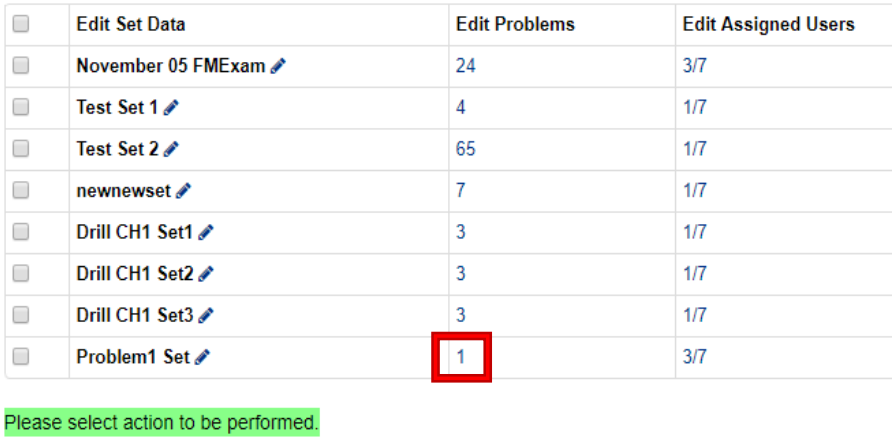


Figure 20 Select number icon to edit problems.

orresponding set.

You can also change the dates here as shown in figure 21, as well as other settings such as "Visible to Students", "Hide Hints", and "Assignment type". These settings should be left on their default unless you know otherwise.

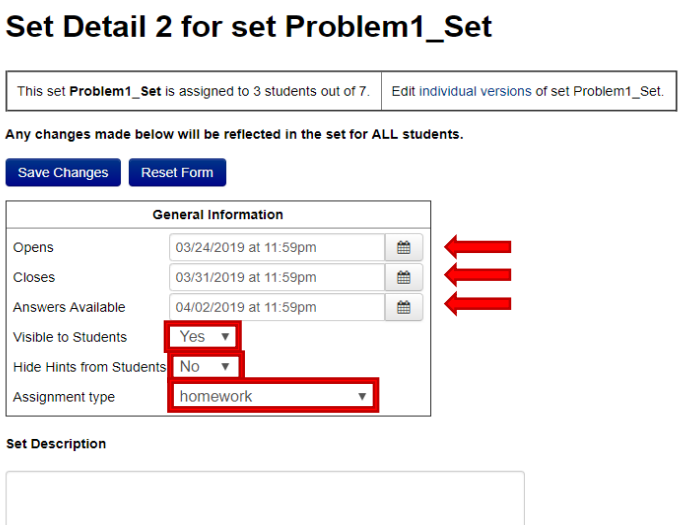


Figure 21 Adujst assignment settings.

If you scroll down the page, you will find even more settings such as “Set Description” and “Set Headers” shown in Figure 22. Additionally, you can view each problem in the set by pressing “Render All”. You may also delete or make other edits to problems, such as assign wrights and the maximum number of times each student can attempt a problem.

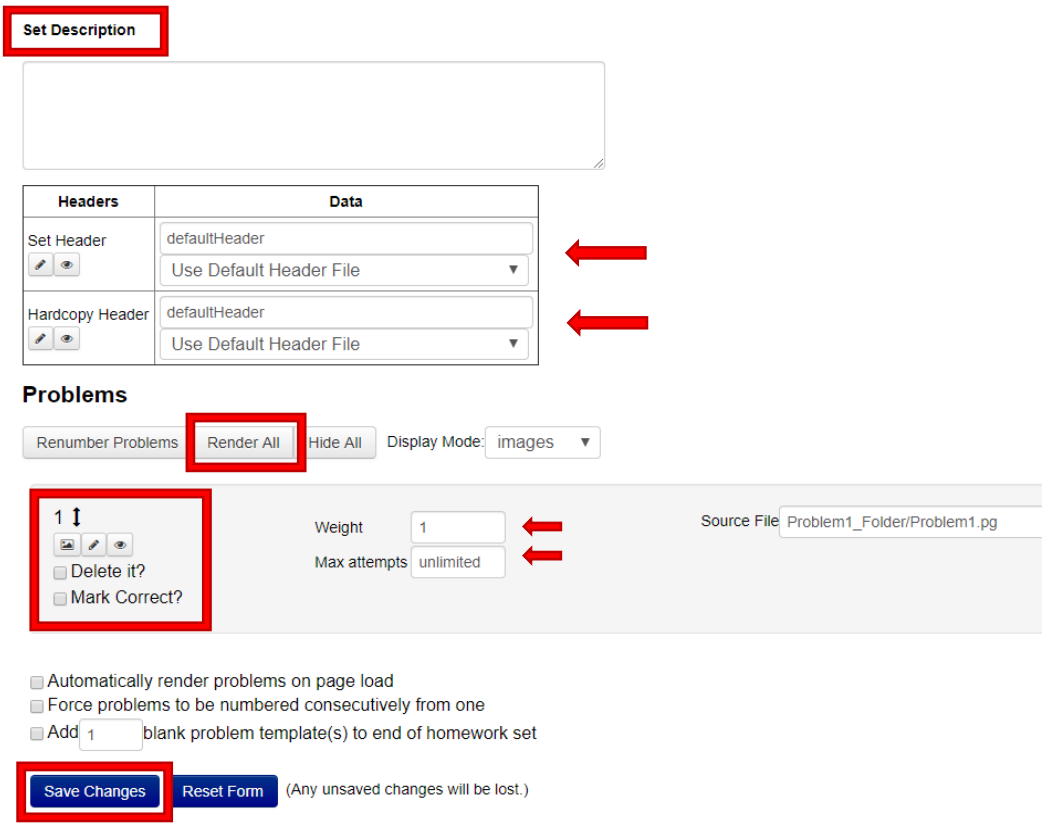
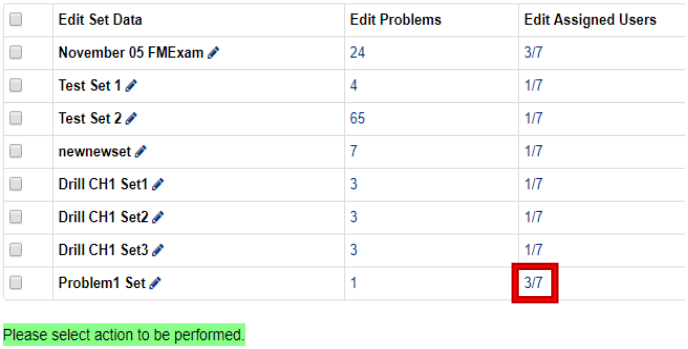


Figure 22 More assignment settings.

3.4 Edit Assigned Users (Assigning Homework Sets to Students)

Once you have created a homework set on WeBWorK and adjusted all of the other settings, you can assign it to the students in your class. Once, again, start from the "Hmwk Sets Editor" tab, and press the fraction under the "Edit Assigned Users" column. This fractions describes how many students the assignment is assigned to. These buttons are indicated in Figure 23.



	Edit Set Data	Edit Problems	Edit Assigned Users
☐	November 05 FMExam	24	3/7
☐	Test Set 1	4	1/7
☐	Test Set 2	65	1/7
☐	newnewset	7	1/7
☐	Drill CH1 Set1	3	1/7
☐	Drill CH1 Set2	3	1/7
☐	Drill CH1 Set3	3	1/7
☐	Problem1 Set	1	3/7

Please select action to be performed.

Figure 23 Select fraction icon under "Edit Assigned Users".

This will bring you to the "User Assigned" page shown in Figure 24. You can then either press "Assign to All Current Users" or manually select which students to assign the set to by checking the boxes next to their names. Do not forget to press "Save".

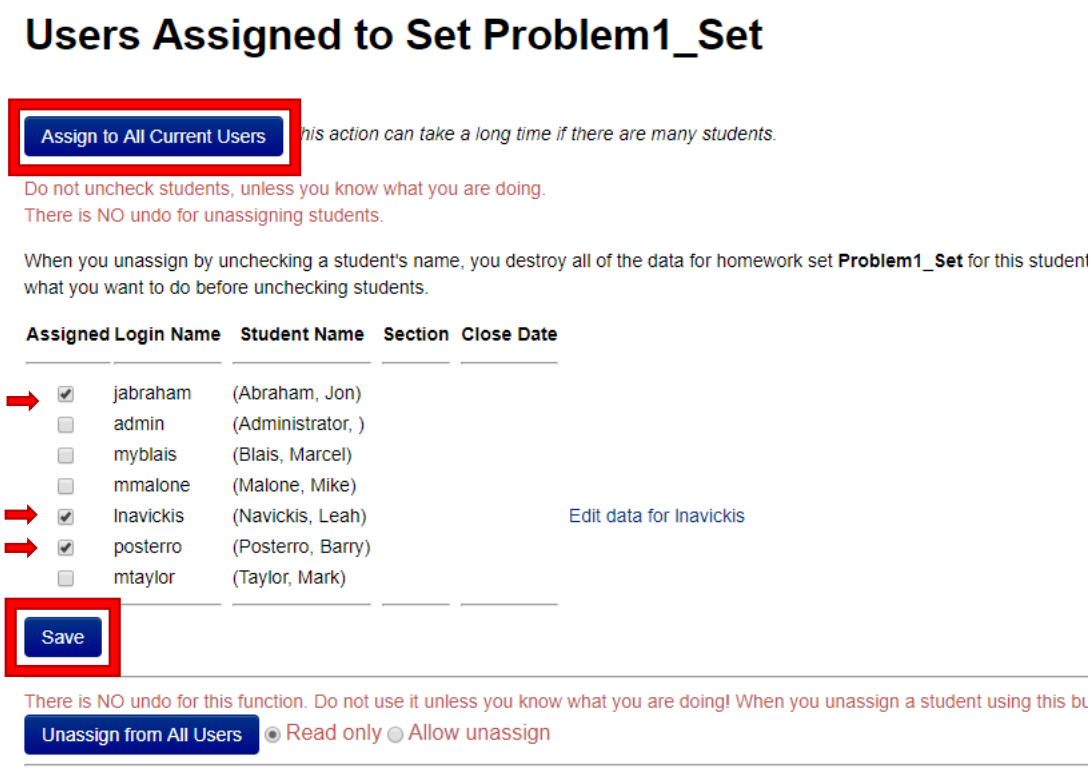


Figure 24 Assign your problem set to students on WeBWorK.

3.5 Assigning Problem Sets on Canvas

You can now create an assignment on Canvas to direct the students to the correct WeBWorK problem set you created in section 3.4. First, log into canvas and select the canvas course you want to add the assignment to. Then navigate to the "Assignments" tab shown in Figure 25, and press the "+Assignment" button.

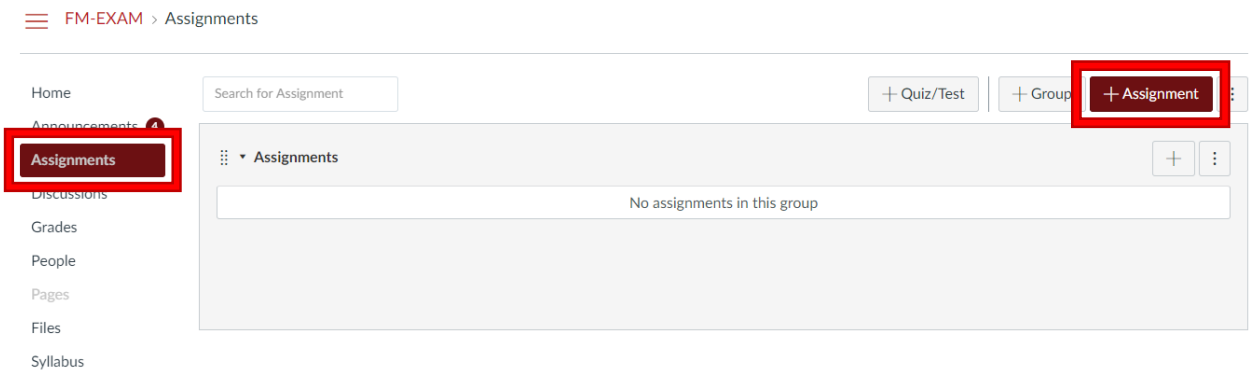


Figure 25 Create a new Canvas assignemnt.

Now your page will look like Figure 26, and be sure to give your assignment a clear name in the textbox indicated.

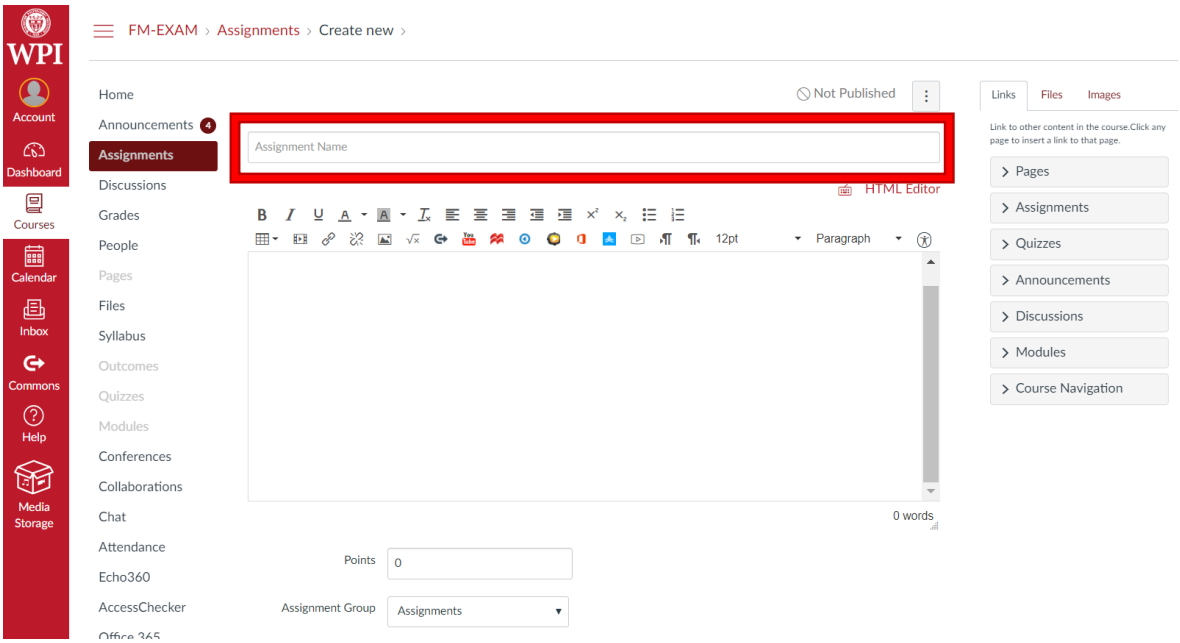


Figure 26 Name Canvas assignemnt.

Now scroll down the page until the "Submission Type" and "Assign" sections are visible on your screen, as shown in Figure 27. In order to provide a link to the WeBWorK assignment for the students, you must first select "External Tool" as the "Submission Type", and add the link to the WeBWorK assignment in the "Enter or find an External Tool URL" field. The URL you need to use is specified in Figure 28. Be sure to select "Load This Tool In A New Tab" so that

the students are not exited from canvas. Next, fill out the “Assign to” and Due date settings in the Assign section. Finally, select “Notify students that is content has changed” so that the students know you have posted an assignment, and press “Save & Publish” to make the assignment available to the students.

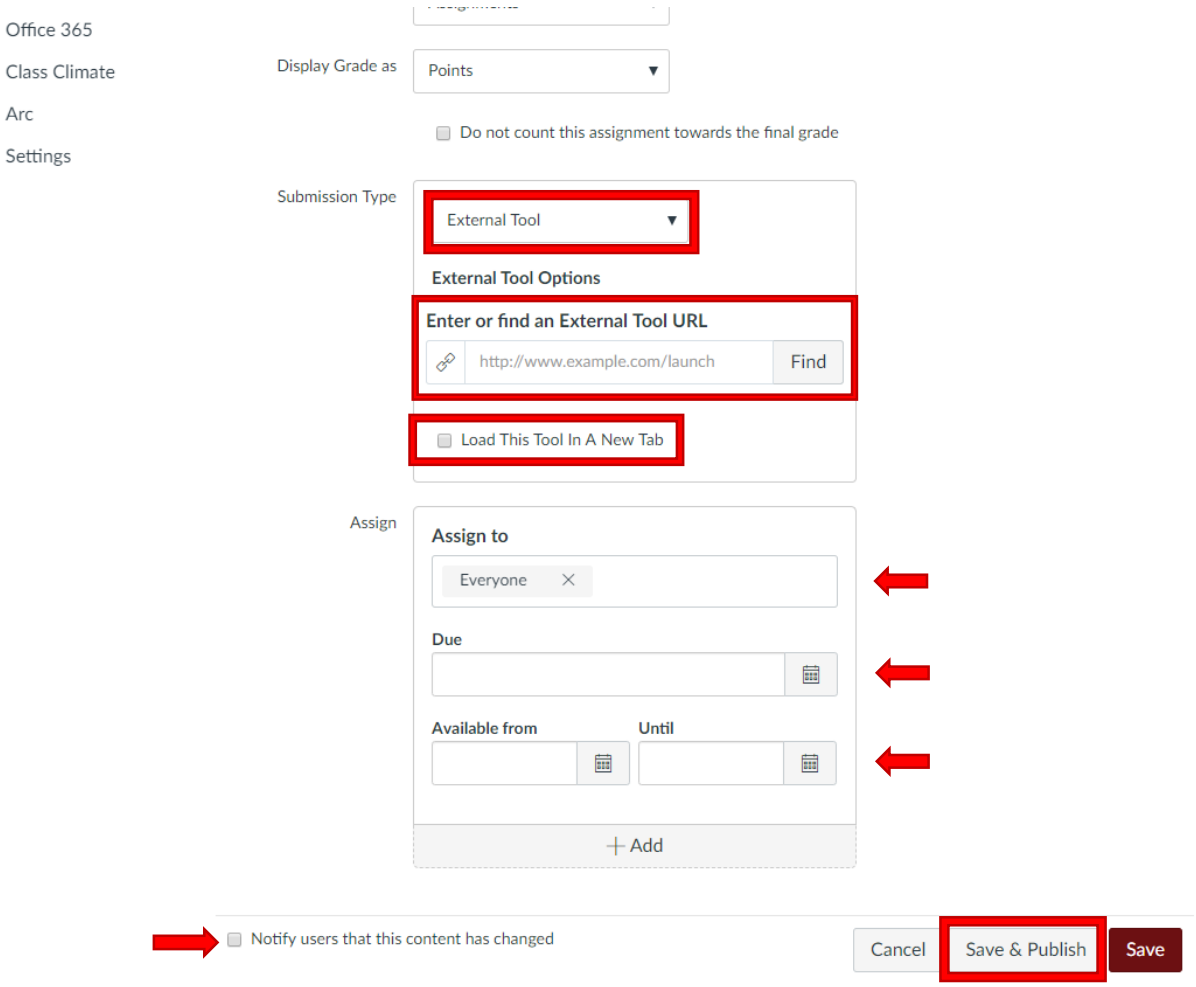


Figure 27 Adjust Canvas assignemnt settings.

Figure 28 depicts the exact page of WeBWorK you should take the URL from. For example, the URL I am using is: https://wwork.wpi.edu/webwork2/FMExam/Problem1_Set/ because that is where my problem set is actually located (you only need to include up to the problem set name for the URL). You can navigate to this page by going to the “Homework Sets” tab and pressing the corresponding homework set.

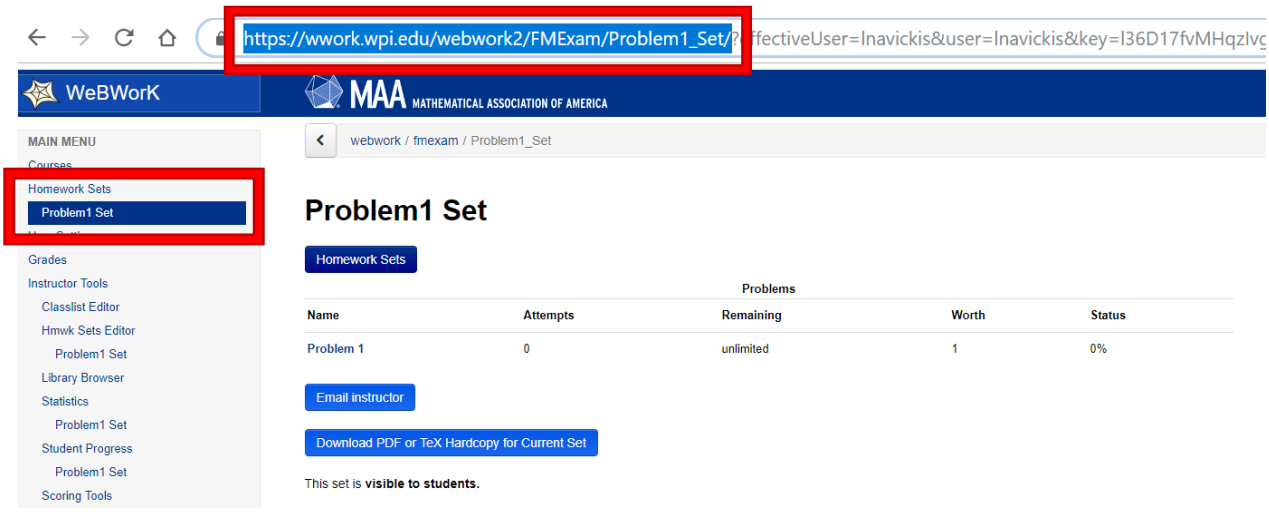


Figure 28 Copy WeBWorK problem set link.

After publishing your assignment in Figure 27, you will now be able to view the assignment you have added in Canvas, shown in Figure 29.

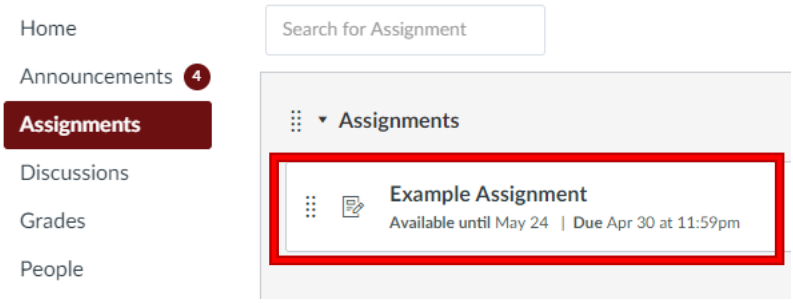


Figure 29 View your new Canvas assignment.

Now, if you press "Load Example Assignment in a new window", you should be taken to the screen shown in Figure 28, because it is what you liked the assignment to. This button is shown in Figure 30.

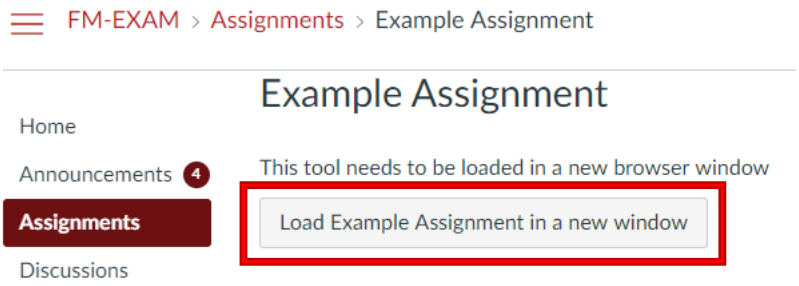


Figure 30 Check link from Canvas to WeBWorK.

4 Writing WeBWorK Problems

4.1 Basic Problem Structure

A description of the basic structure of a WeBWorK problem can be found online on WeBWorK's website (SampleProblem1: A First WeBWorK Sample Problem, 2011). I also provide full problem examples later in Section 5.

As you add more problems to your WeBWorK course, it becomes increasingly difficult to find the problems you are looking for. This is why it is important to comment an adequate description at the beginning of each problem code. An example of a descriptive heading is shown in Figure 31. These comments (denoted by #) are not read by WeBWorK, they are intended to provide context for whoever is reading the problem code.

```
DOCUMENT();
# Classification('Type of Problem')
# This is a simple annuity problem.

## KEYWORDS('Annuities', 'Interest')
## Exam('FM Exam', 'Date', 'Problem Number')
## Institution('WPI')
## Author('Leah Navickis', 'WPI Actuarial')
## Date('MM-DD-YYYY')
```

Figure 31 Example problem heading.

For the FM Exam, I used 10 different classifications, shown in Figure 32 for the problems. I indicated their category in the "Classification()" section at the top of each problem, as shown in Figure 31. I also created a folder in the FM Exam course for each classification, and added the problems to their corresponding folder, so that the professors will have an easier time finding the problems in the future. I suggest doing the same in any future projects.

Classifications	
1	present_future_acc_value
2	annuities
3	loans_bonds
4	yield_curves
5	IRR_NPV
6	rates_of_return
7	duration_convexity
8	immunization
9	swaps
10	spots_forwards

Figure 32 FM Exam problem classifications.

4.2 Useful WeBWorK Functions

There are many functions used in WeBWorK so that your problem is displayed correctly. Figure 33 contains some useful code.

Description	Code
$\ddot{s}_{\overline{n}\rceil j}$	\(\ddot s_{\overline{n}\rceil j}\)
$\ddot{a}_{\overline{n}\rceil j}$	\(\ddot a_{\overline{n}\rceil j}\)
$s_{\overline{n}\rceil j}$	\(s_{\overline{n}\rceil j}\)
$a_{\overline{n}\rceil j}$	\(a_{\overline{n}\rceil j}\)
$s_{\overline{n}\rceil}$	\(s_{\overline{n}\rceil}\)
$a_{\overline{n}\rceil}$	\(a_{\overline{n}\rceil}\)
display image	\{ image("image.png", width=>500) \}
subscript	_{put expression here}
superscript	^{put expression here}
exponent	**
fraction	\frac{numerator}{denominator}
new paragraph	$PAR
add an equation	\(equation\)
multiple choice format	$mc = new_multiple_choice()
create a variable	$variable

Figure 33 Useful WeBWorK syntax.

5 Relevant Problem Types and Examples

In this section, I will discuss the most relevant problem types to the FM Exam WeBWorK course. All exam questions were written as multiple choice questions, and all drill questions were written as simple dynamic questions. For each type of question, I provide a brief description of the problem type, list the code, and then show an image of how the question appears as to the student.

5.1 Multiple Choice

This type of multiple choice question utilizes the $mc variable, randomly orders the answer choices, and has only one correct answer. View this type of problem below:

```
DOCUMENT();

# Mutlitple Choice Question Template

# DESCRIPTION
# Add description here.
# ENDDESCRIPTION

loadMacros("PGbasicmacros.pl",
        "PGchoicemacros.pl",
        "PGanswermacros.pl",

);
TEXT(beginproblem());

$showPartialCorrectAnswers = 0;

# Make a new multiple choice object.
$mc = new_multiple_choice();

# Insert question and answer
$mc -> qa ("The first string is the question",
        "The second string is the correct answer",
);
$mc ->extra(#add the incorrect options here
        "wrong1",
        "wrong2",
        "wrong3",
        "wrong4",
);
# Print the question using $mc->print_q
# Use $mc->print_a to print the list of possible answers.
# This need to be done inside BEGIN_TEXT/END_TEXT.

BEGIN_TEXT

\{$mc -> print_q \}
$PAR
\{$mc -> print_a\}
END_TEXT
# Enter the correct answers to be checked against the answers to the
students.
ANS( str_cmp( $mc->correct_ans )) ;

BEGIN_SOLUTION

Type Solution Here
```

```
END_SOLUTION

ENDDOCUMENT();
```

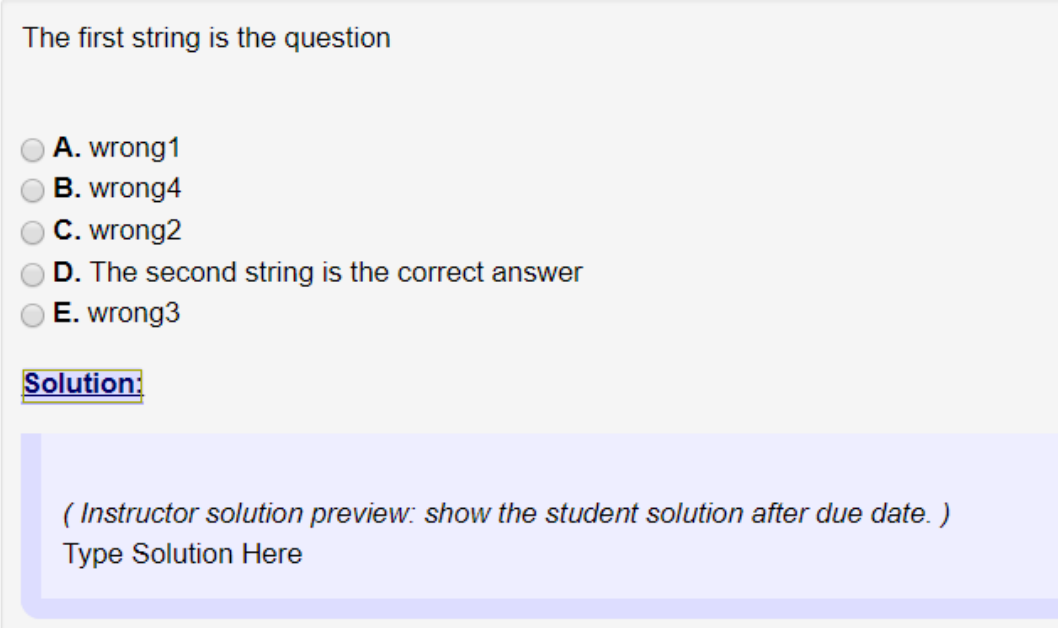


Figure 34 Student View: Multiple Choice Question

5.2 Simple Dynamic

In a simple dynamic question, you can create variables that will change depending on who is viewing the problem. For example, in the problem below, each student will get different values of $a, other than 1600:

```
DOCUMENT();

# Simple Dynamic Question Template

# DESCRIPTION
# Add description here.
# ENDDESCRIPTION

loadMacros(
  "PGstandard.pl",
  "PGchoicemacros.pl",
  "PGasu.pl",
  "PGcourse.pl"
);

TEXT(beginproblem());

$a = random(10,20,1)*100;
$ans = $a+100;
```

```
TEXT(EV2(<<EOT));
What is \($a+100\)?

$BR
Answer  = \{ans_rule(20)\}
$BR
$BR
EOT

SOLUTION(EV3(<<'EOT'));
$BBOLD SOLUTION: $EBOLD
$BR
\($ans=$a+100\).

EOT

ANS(num_cmp($ans));

ENDDOCUMENT();
```

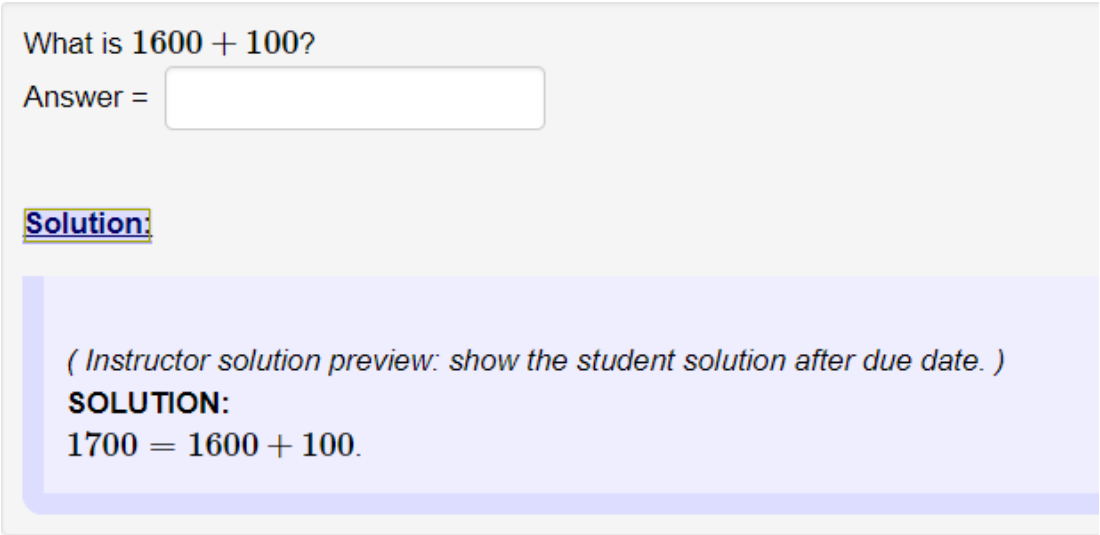


Figure 35 Student View: Simple Dynamic Question

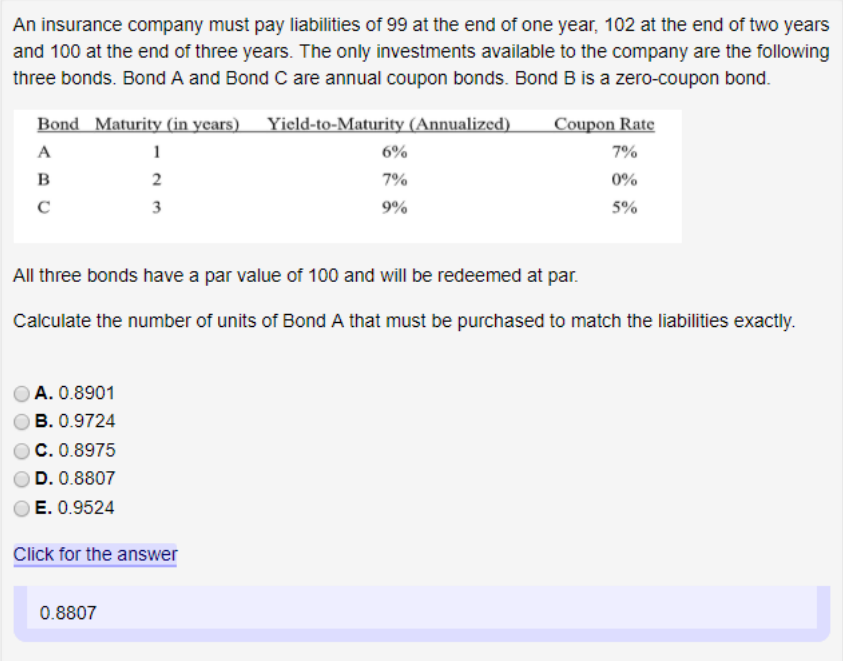
5.3 With Image

Knowing how to insert an image in q problem will come in handy. You can do this by using the image() function in the text of the problem, and by uploading the image directly into your WeBWorK folder with the question. An example of how to add an image in a problem is shown below:

```
DOCUMENT();
```

```
loadMacros(
      "PGstandard.pl",
      "PGbasicmacros.pl",
         "PGchoicemacros.pl",
         "PGanswermacros.pl",
      "PGasu.pl",
      "PGcourse.pl",

);
TEXT(beginproblem());

$showPartialCorrectAnswers = 0;

# Make a new multiple choice object.
$mc = new_multiple_choice();

# Insert question and answer
$mc -> qa ("Calculate the number of units of Bond A that must be
purchased to match the liabilities exactly.",
        "0.8807",
);
$mc ->extra(#add the incorrect options here
        "0.8901",
        "0.8975",
        "0.9524",
        "0.9724",
);
# Print the question using $mc->print_q
# Use $mc->print_a to print the list of possible answers.
# This need to be done inside BEGIN_TEXT/END_TEXT.

BEGIN_TEXT
An insurance company must pay liabilities of 99 at the end of one
year,
102 at the end of two years and 100 at the end of three years. The
only
investments available to the company are the following three bonds.
Bond
A and Bond C are annual coupon bonds. Bond B is a zero-coupon bond.
$PAR
\{image("PE17im69.png", width=>500)\} $PAR
All three bonds have a par value of 100 and will be redeemed at par.
$PAR
END_TEXT

BEGIN_TEXT

\{$mc -> print_q \}
$PAR
\{$mc -> print_a\}
END_TEXT
```

```
BEGIN_TEXT
$PAR
$PAR
\{
knowlLink("Click for the answer",
value=>'0.8807')
\}
$PAR
END_TEXT

# Enter the correct answers to be checked against the answers to the
students.
ANS( str_cmp( $mc->correct_ans )) ;

ENDDOCUMENT();
```



An insurance company must pay liabilities of 99 at the end of one year, 102 at the end of two years and 100 at the end of three years. The only investments available to the company are the following three bonds. Bond A and Bond C are annual coupon bonds. Bond B is a zero-coupon bond.

Bond	Maturity (in years)	Yield-to-Maturity (Annualized)	Coupon Rate
A	1	6%	7%
B	2	7%	0%
C	3	9%	5%

All three bonds have a par value of 100 and will be redeemed at par.

Calculate the number of units of Bond A that must be purchased to match the liabilities exactly.

- **A.** 0.8901
- **B.** 0.9724
- **C.** 0.8975
- **D.** 0.8807
- **E.** 0.9524

Click for the answer

0.8807

Figure 36 Student View: Question with Image

5.4 With Hints

This problem is an example of how to add hints. The hint in this problem appears as "Click here for the answer", however you can name your hint whatever you want. In this project, I elected to use hints to show solutions instead of solutions (as shown in section 5.1), because the students can view hints without the professors setting a due date for the problems. It is okay for the students to have access to the solutions because it is a study tool, so grading is not important. Check out the code below:

```
DOCUMENT();
```

```
loadMacros(
      "PGstandard.pl",
      "PGbasicmacros.pl",
         "PGchoicemacros.pl",
         "PGanswermacros.pl",
      "PGasu.pl",
      "PGcourse.pl",

);
TEXT(beginproblem());

$showPartialCorrectAnswers = 0;

# Make a new multiple choice object.
$mc = new_multiple_choice();

# Insert question and answer
$mc -> qa ("Calculate \(\delta\).",
        "0.0396",
);
$mc ->extra(#add the incorrect options here
        "0.0388",
        "0.0392",
        "0.0404",
        "0.0414",
);
# Print the question using $mc->print_q
# Use $mc->print_a to print the list of possible answers.
# This need to be done inside BEGIN_TEXT/END_TEXT.

BEGIN_TEXT
Bruce deposits 100 into a bank account. His account is credited
interest at an annual nominal
rate of interest of 4% convertible semiannually. $PAR

At the same time, Peter deposits 100 into a separate account. Peter's
account is credited interest
at an annual force of interest of \(\delta\). $PAR

After 7.25 years, the value of each account is the same. $PAR
END_TEXT

BEGIN_TEXT

\{$mc -> print_q \}
$PAR
\{$mc -> print_a\}
END_TEXT

BEGIN_TEXT
$PAR
$PAR
```

```
\{
knowlLink("Click for the answer",
value=>'0.0396')
\}
$PAR
END_TEXT

# Enter the correct answers to be checked against the answers to the
students.
ANS( str_cmp( $mc->correct_ans )) ;

ENDDOCUMENT();
```

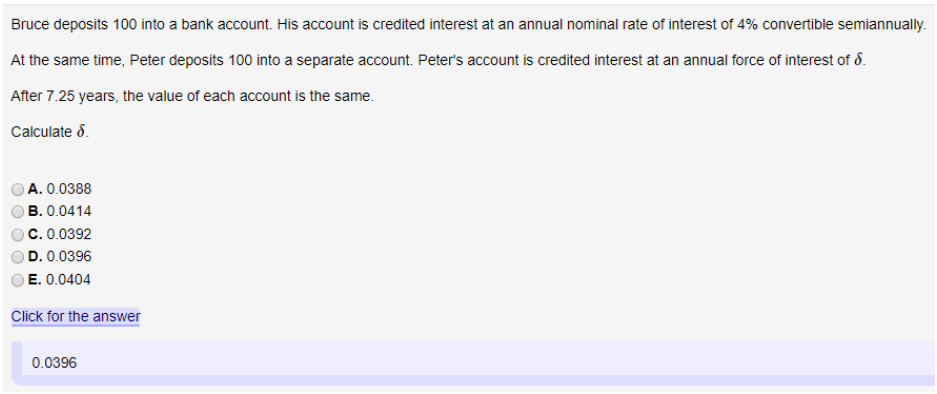


Figure 37 Student View: Question with Hint

References

Instructors. (2016, May 2). Retrieved from webwork.maa.org/wiki: http://webwork.maa.org/wiki/Instructors#.XLJJcuhKg2w

Main Page: WeBWorK Doccumention Wiki. (2019, January 6). Retrieved from webwork.maa.org/wiki: http://webwork.maa.org/wiki/Main_Page#.XLJG_ehKg2w

Problem Authoring Background Information. (2015, September 28). Retrieved from webwork.maa.org: http://webwork.maa.org/wiki/Problem_Authoring_Background_Information#.XLIvvehKg2w

SampleProblem1: A First WeBWorK Sample Problem. (2011, May 25). Retrieved from webwork.maa.org: http://webwork.maa.org/wiki/SampleProblem1#.XLIpvuhKg2w

Using WeBWorK: General Forums. (n.d.). Retrieved from webwork.maa.org: http://webwork.maa.org/moodle/mod/forum/index.php?id=3

WeBWorK Sites. (2016, June 22). Retrieved from WeBWork: http://webwork.maa.org/wiki/WeBWorK_Sites#.XK-AHOhKg2w