Coding Problems in LON-CAPA for Authors/Co-Authors

Originally created by Jonathan Abbott for the benefit of teachers coding problems in LON-CAPA. Updated by Gerd Kortemeyer September 2012. Feel free to adapt and copy.

The goal of this tutorial is to provide a set of instructions to coding problems in LON-CAPA. This is largely a guide to the fundamentals of coding problems.

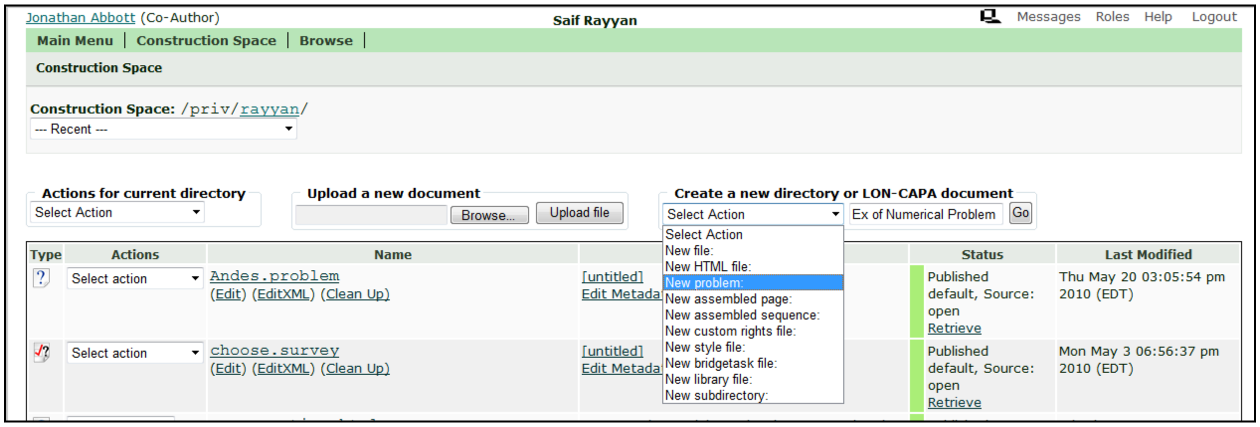
Overview:

1. Initiate Creating a New Numerical Problem from a Template
2. The Interface for Coding Problems
    a. First Representation: “Edit”
    b. Second Representation: “EditXML”
    c. Third Representation: “View”
3. Necessary Fundamentals to Coding Problems
    a. Evaluating the Answer
    b. Basics of Coding in Perl
    c. Randomization
    d. Hints
    e. Publishing
4. Other problem types
    a. Making Symbolic Problem
5. Errors within LON-CAPA with Interpreting Answers
6. Tips
    a. Having hints show up after more than one try
    b. Add multiple parts to questions
    c. Show something after the question is solved or unsolved.
    d. Bold, Italics, horizontal line
7. Link: using&ext()

Initiate Creating a New Numerical Problem from a Template

Let's go through the process of making a numerical problem from a template so you can experiment with the interface and coding.

1. After logging in (and, if you have several roles in the system, selecting the role of Author), you will need to go to the construction space. You can just click construction space in the top left if not already there. The construction space can be used for any multimedia type (images, web pages, movies, etc), but for the purposes of this simple tutorial, we limit ourselves to problems.
2. Let's make a new problem, so use the dropdown as shown in the figure below. Be sure to provide a filename for your problem and then click Go. It is recommended to not use special characters in the filename, as it later translates into a URL, where special characters make for a long, awkward address. Click Continue on the next screen.


Jonathan Abbott (Co-Author) Saif Rayyan

Construction Space

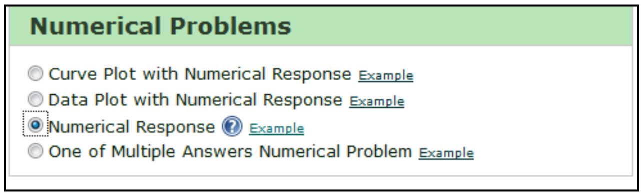
Construction Space: /priv/rayyan/
--- Recent ---

Actions for current directory
Select Action
Upload a new document
Browse...
Upload file
Create a new directory or LON-CAPA document
Select Action

Ex of Numerical Problem

Type	Actions	Name		Status	Last Modified
?	Select action	Andes.problem (Edit) (EditXML) (Clean Up)	[untitled] Edit Metada	Published default, Source: open Retrieve	Thu May 20 03:05:54 pm 2010 (EDT)
✓?	Select action	choose.survey (Edit) (EditXML) (Clean Up)	[untitled] Edit Metada	Published default, Source: open Retrieve	Mon May 3 06:56:37 pm 2010 (EDT)

3. Now you should see a screen with many problem templates. These templates are not problem types, but in fact just starting points for your problems. LON-CAPA problems are constructed from functional elements that can be combined extremely flexibly. Over time, you will become more familiar with them, but let's stick to numerical response down in the bottom right as shown in the figure below. Click Continue.


Numerical Problems

- ○ Curve Plot with Numerical Response Example
- ○ Data Plot with Numerical Response Example
- ● Numerical Response Example
- ○ One of Multiple Answers Numerical Problem Example

4. Now you are at the representation of the template problem the students see. Click on "Edit" or "EditXML" to switch to the other views.

The Interface for Coding Problems

Sooner or later you will actually have to code your own problems. Fortunately, you can easily start with other problems as a template. Using a template is a good start and a great time saver, but let's begin by previewing the interface where you will code problems.

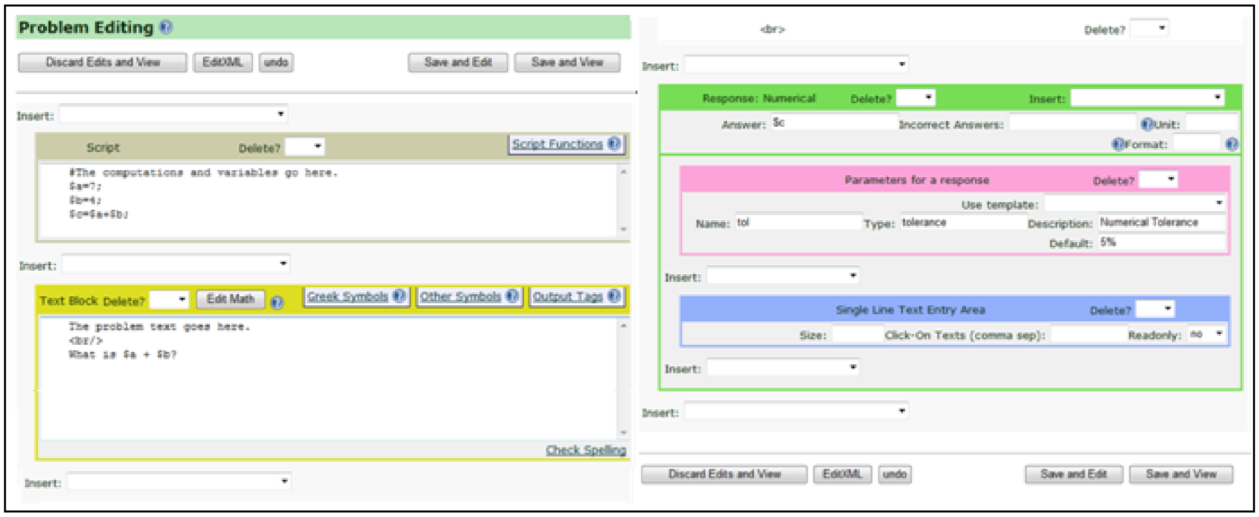
Below you see three "representations" of the same problem. One representation is colorful with a bunch of boxes, the second representation is simply code, and the last representation is the final display of how the student sees the problem.

In practice, you will need to use all three representations, so let's quickly go over each view.

First Representation: "Edit"

LON-CAPA calls the colorful representation with boxes as simply "Edit." Each colorful box is each a functional element or group of functional elements with their relevant inputs. For instance:

- The *Script* is where you can define variables and do computations.
- The *Text Block* is where you can directly add text to the problem.
- The *Response* is where you can ask for and then evaluate the student's response.



```
#The computations and variables go here.
$a=7;
$b=4;
$c=$a+$b;
```

```
The problem text goes here.
<br/>
What is $a + $b?
```


Answer: $c Incorrect Answers: Unit: Format:

Use template:
Name: tol Type: tolerance Description: Numerical Tolerance
Default: 5%

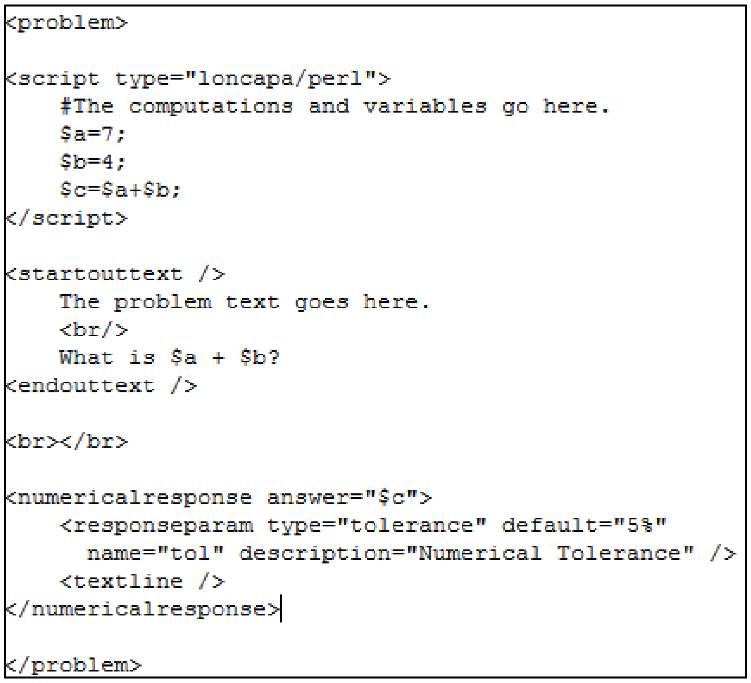
Size: Click-On Texts (comma sep): Readonly: no

Second Representation: "EditXML"

The second representation is simply XML code, but you can see how it matches the first representation.

- The code starts with <problem> and then ends with </problem>.
- In-between <script> and </script>, you can define variables and do computations.
    - Note: the language used in this area is a mix between LON-CAPA and Perl. (We'll talk about more code shortly).

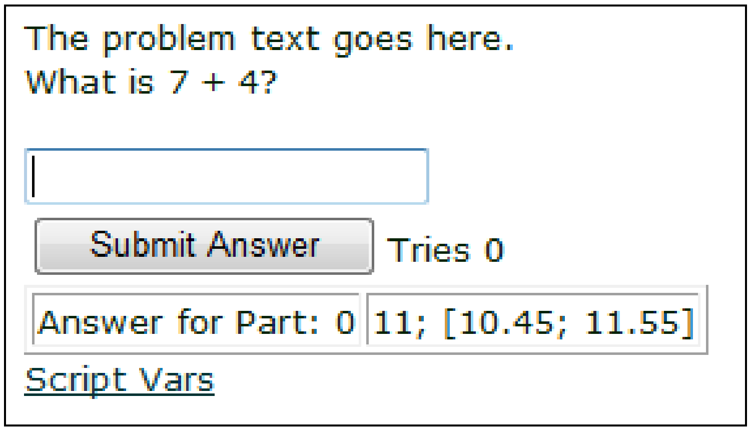
- In-between <startouttext /> and <endouttext />, you can directly add text to the problem.
- In-between <numericalresponse answer="$c"> and </numericalresponse>, you can ask for and then evaluate the student's response, as well as add settings ("parameters").



```
<problem>

<script type="loncapa/perl">
    #The computations and variables go here.
    $a=7;
    $b=4;
    $c=$a+$b;
</script>

<startouttext />
    The problem text goes here.
    <br/>
    What is $a + $b?
<endouttext />

<br></br>

<numericalresponse answer="$c">
    <responseparam type="tolerance" default="5%"
      name="tol" description="Numerical Tolerance" />
    <textline />
</numericalresponse>

</problem>
```

Third Representation: "View"

- You should not see anything from the code in "script" as that was all computer computation.
- You just see the problem text and an answer box.
- Since you are an instructor, you also enter information about the correct answer and what range of student answers will be considered acceptable.



The problem text goes here.
What is 7 + 4?

Answer for Part: 0	11; [10.45; 11.55]

Necessary Fundamentals to Coding Problems

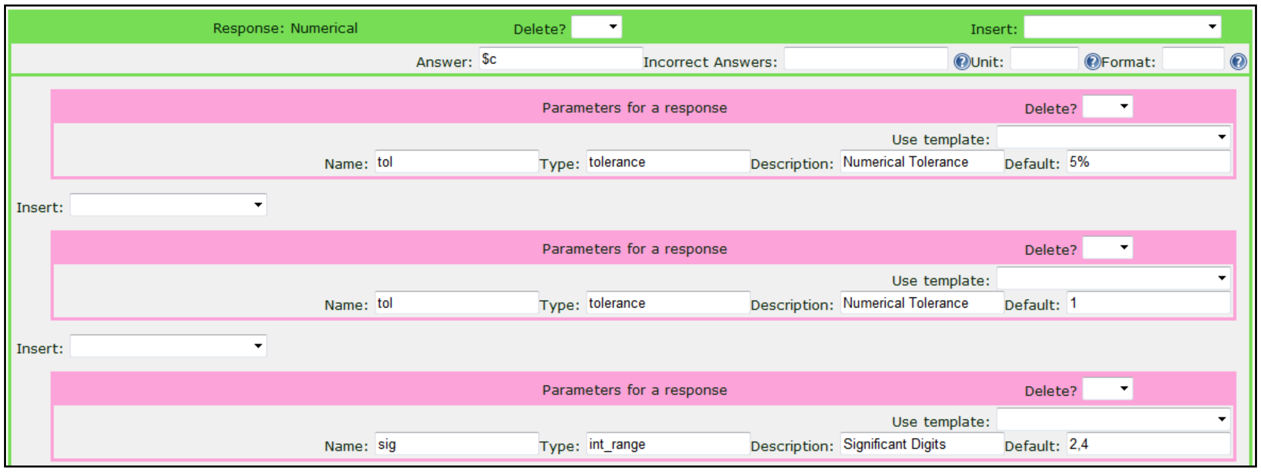
Evaluating the Answer

Let's go over how to adjust how LON-CAPA judges the student's answer. The correct answer simply needs to be given in the "Answer" field, and can be number of a variable. The physical unit of the answer should be given in the "Unit" field; LON-CAPA will accept any student answer that has the right dimension and do the unit conversions automatically. The "Format" field can be used to provide appropriate computer answer display – for example, entering "3s" would display the computer answer with three significant digits. Again, the blue help symbols get you a window with relevant information.

Since floating point calculations may always have some round-off/calculator error, it's wise to give range of input that is considered correct. As shown by Figure 6, you can specify the percent (i.e. +/- 5%) the student's answer can deviate from the computer's answer or you can specify the range (i.e. +/- 1).

You can also see in figure below that you can use require significant figures. Note that 2,4 means the answer must have between 2 and 4 significant digits.

To insert more "Parameters for a response" just use the insert dropdown in the numerical block. Then use the "Use template" to add more criteria.



Response: Numerical Delete? Insert:

Answer: $c Incorrect Answers: Unit: Format:

Parameters for a response Delete?

Use template:

Name: tol Type: tolerance Description: Numerical Tolerance Default: 5%

Insert:

Parameters for a response Delete?

Use template:

Name: tol Type: tolerance Description: Numerical Tolerance Default: 1

Insert:

Parameters for a response Delete?

Use template:

Name: sig Type: int_range Description: Significant Digits Default: 2,4

Basics of Coding in Perl

Let's start covering the basics of the Perl code.

- Put the Perl code in the "Script" box from the Edit screen or in-between <script type="loncapa/perl"> and </script> on the EditXML screen.
- Write yourself notes in the format #yourtext. Writing meaningful notes inside the code (that the computer otherwise ignores) will help you to remember what you were trying to do.
- Define variables in the format $variable. Always refer to the variables with the "$".

    - You can define variables using other variables in the form $newvariable=$a+$b;
- End every line of code with a semicolon “;”.

Randomization

To randomize values in Perl use the format $variablename =&random(low,high,interval) as shown below. Randomization is crucial to preventing cheating and randomization allows for more flexibility with questions. There are additional randomization routines. In fact, in the colorful representation, the blue help box “Script Functions” gives you an overview of all functions that are available.

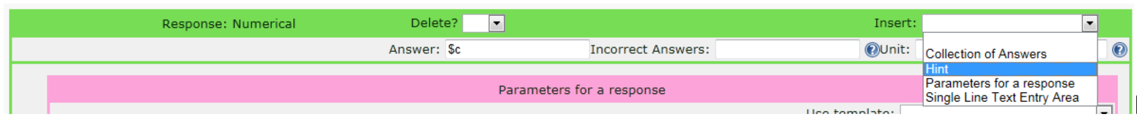
```
<script type="loncapa/perl">

#To randomize variables use the form
# &random(low,high,interval).

$a=&random(1,10,1);

</script>
```

Hints

Hints are very confusing to first learn because they have a lot of options and take a while to input. Let’s first cover some nomenclature:

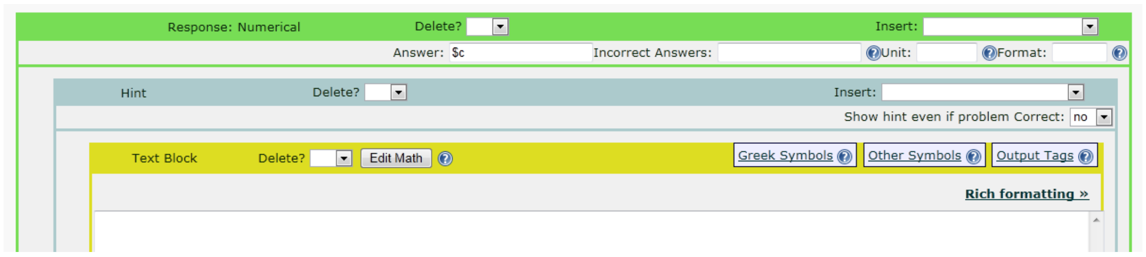
- A “hint” that shows up by default is a text box that shows up after a student has attempted a problem.
- A “hint condition” is where you define criteria (like a specific answer) and LON-CAPA evaluates if the criteria are true.
- A “conditional hint” and the “Text/HTML Block” are the response and output to the above criteria being true.

So let’s walk through making a default hint (which shows just when you want text to appear after a student attempts the problem.

1. Create a numerical response question and get to the “Edit” representation of the problem.
2. Now in the green Response: Numerical box, insert a “hint” using the right dropdown box.
3. 

N
ow go to the bottom of the page and click “Save and Edit.” When the page reloads you will have your default hint.

4. Enter what text you want to appear in the text block. If the “show hint even if problem Correct” dropdown is set to “no,” the text block will only show if the student gets the answer wrong.

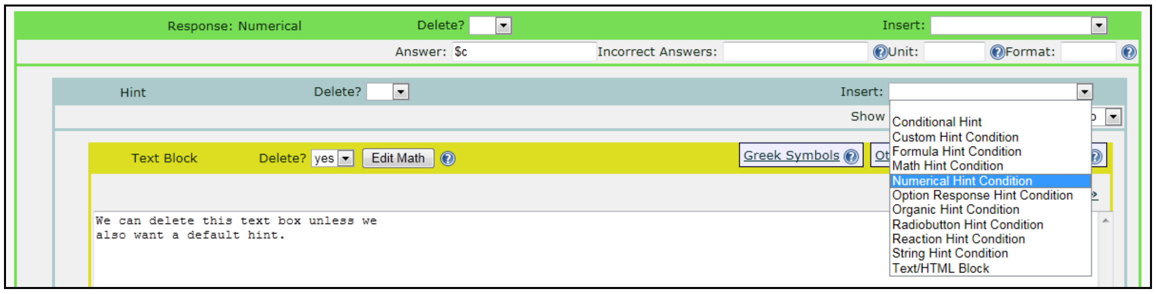


Response: Numerical Delete? Insert:
Answer: $c Incorrect Answers: Unit: Format:
Hint Delete? Insert:
Show hint even if problem Correct: no
Text Block Delete?

5. To delete a hint (or anything else), just set the appropriate dropdown box to true and then click save and edit.

Let’s now make a conditional hint. This may be slightly tedious and confusing at first so let’s summarize what we are trying to do: first we make criteria for when the hint will be shown and then we make the actual hint.

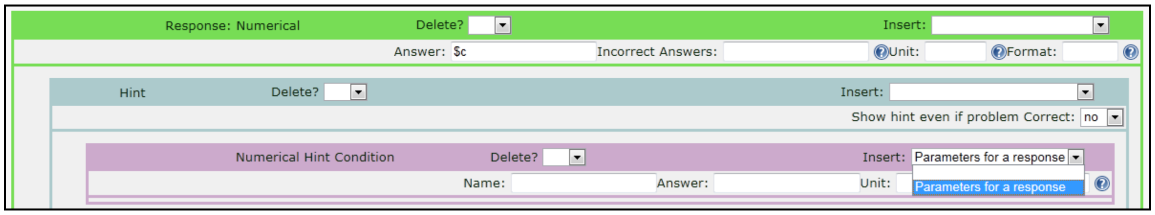
1. From where we let off with the default hint, you will want to now insert from the hint dropdown box a “Numerical Hint Condition.” Consider also deleting the text block by setting the text block dropdown to yes. Click “Save and “Edit.”



Response: Numerical Delete? Insert:
Answer: $c Incorrect Answers: Unit: Format:
Hint Delete? Insert:
Show
Text Block Delete? yes

```
We can delete this text box unless we
also want a default hint.
```

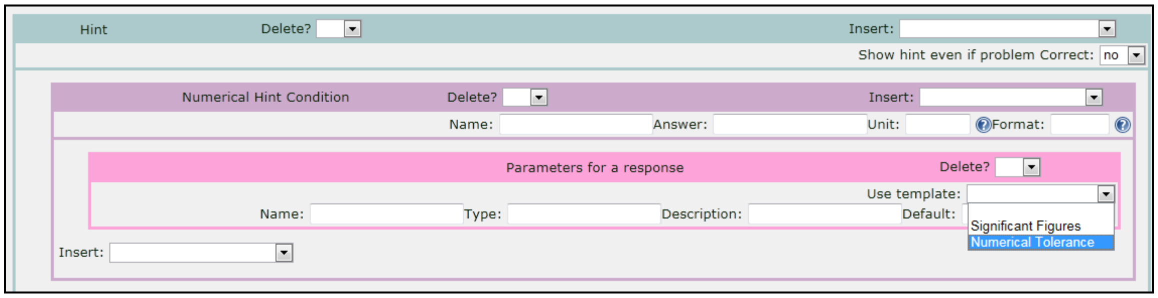
2. Now go back and insert “Parameters for a Response” in the Numerical Hint Condition dropdown. Click “Save and
Edit.”



Response: Numerical Delete? Insert:
Answer: $c Incorrect Answers: Unit: Format:
Hint Delete? Insert:
Show hint even if problem Correct: no
Numerical Hint Condition Delete? Insert: Parameters for a response
Name: Answer: Unit:

3. Next use the template “Numerical Tolerance” to add appropriate criteria for response. Click “Save and

Edit."



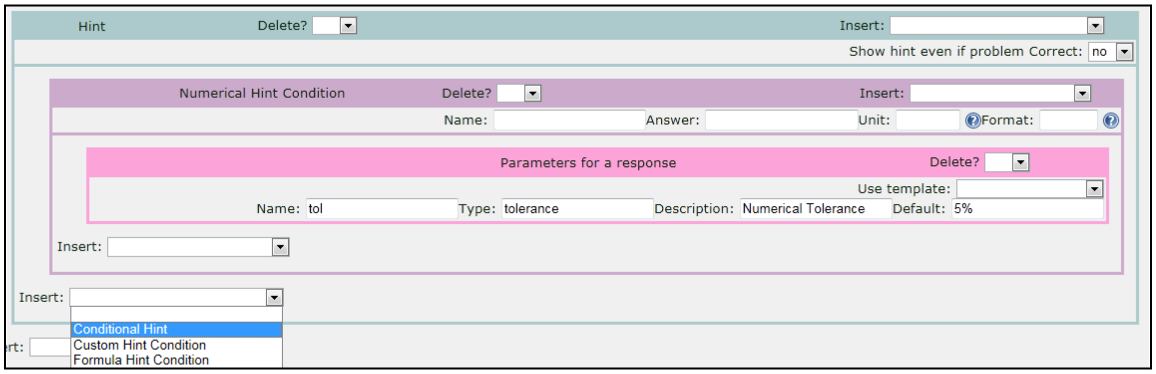
Show hint even if problem Correct: no

Name: Answer: Unit: Format:

Use template:

Name: Type: Description: Default:

4. Then insert a conditional hint.



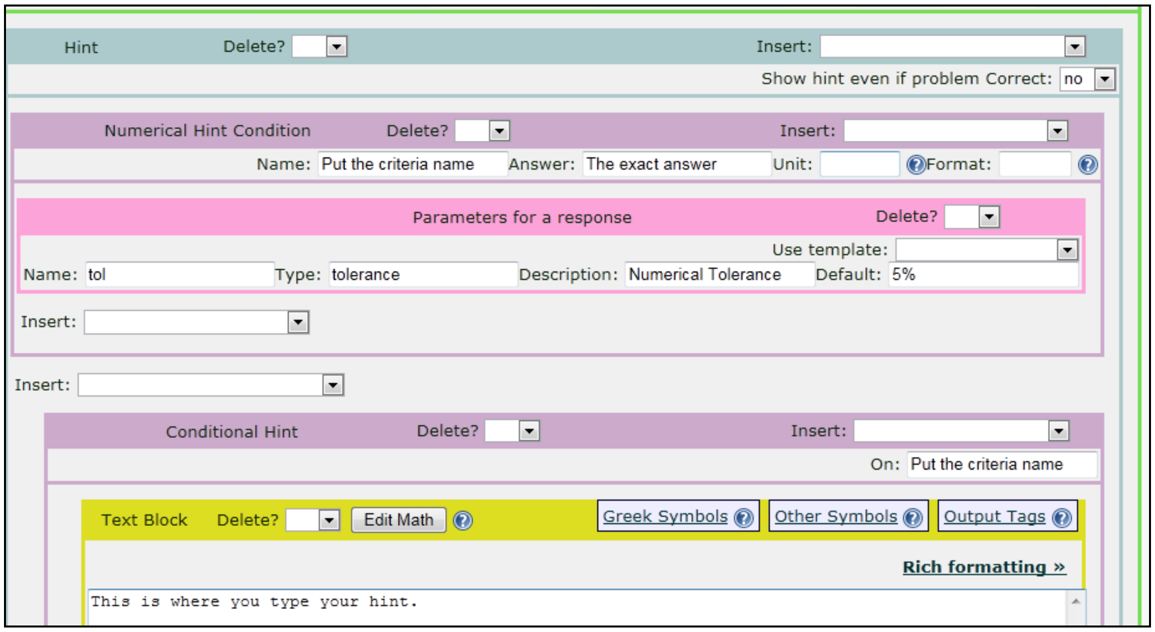
Show hint even if problem Correct: no

Name: Answer: Unit: Format:

Use template:

Name: tol Type: tolerance Description: Numerical Tolerance Default: 5%

5. Finally we have everything to make our hint. So let's decide what we should put as our hint.



Show hint even if problem Correct: no

Name: Put the criteria name Answer: The exact answer Unit: Format:

Use template:

Name: tol Type: tolerance Description: Numerical Tolerance Default: 5%

On: Put the criteria name

```
This is where you type your hint.
```

6. From this picture you can see some of the "fields" where you need to put information. Let's cover them starting with the most important fields.
   a. Name the criteria. You have to name each individual criterion first in the Numerical Hint Condition box and then in the Conditional Hint box.
   b. Specify the answer. You need to put in the exact answer. Note: you can use a variable that you defined in the script section of the code. (This is very helpful if you had randomization.)

c. Type in the text to display. Note: you can also add in other items such as pictures or plots by using the insert dropdown for the conditional hint.

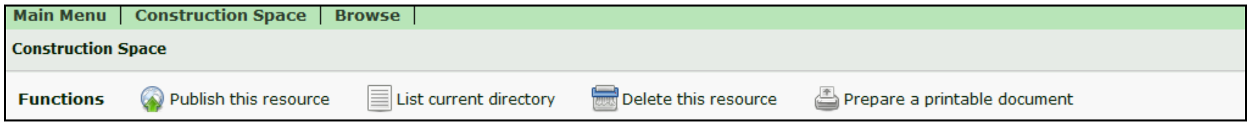
d. Note that any number of hint boxes can be triggered by the same criterion. Once a criterion is named and defined, it can trigger additional functionality later. Hint criteria can also be logically combined (e.g., “this and that”).

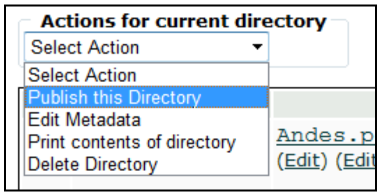
7. Remember to click “Save and View” to test out your hint.

Publishing

Once you finish a problem you will need to publish it before you (or anyone else) can use it.

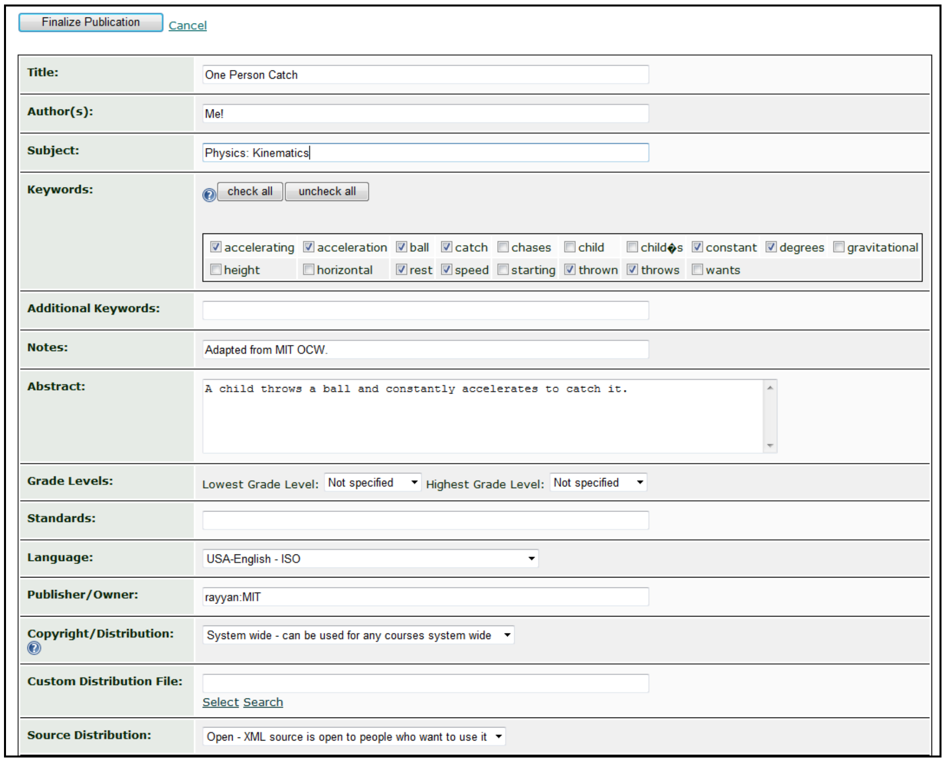
You can either publish the problem when in the construction space of the particular problem or you can publish a whole directory of problems when in the construction space of the directory.





Usually you will just publish the problem after you have created it. So let’s walk through publishing a problem.

1. From the construction space of the problem click “Publish this resource.”
2. Then fill out the metadata. Metadata is information about data (in this case it is about the problem.
    a. Some of the most important fields are the title, keywords, copyright distribution, and source distribution.
    b. The title helps when importing the problem in a course. It is the default entry in the table of contents of a course, and it by default appears in the browser title bar when the problem is used in a course.
    c. The keywords help users search for the problem.
    d. The copyright distribution determines who can access the problem.
    e. The source distribution determines who can access the code to the problem.


Finalize Publication Cancel

Title:	One Person Catch
Author(s):	Me!
Subject:	Physics: Kinematics
Keywords:	check all / uncheck all ☑ accelerating ☑ acceleration ☑ ball ☑ catch ☐ chases ☐ child ☐ child�s ☑ constant ☑ degrees ☐ gravitational ☐ height ☐ horizontal ☑ rest ☑ speed ☐ starting ☑ thrown ☑ throws ☐ wants
Additional Keywords:	
Notes:	Adapted from MIT OCW.
Abstract:	A child throws a ball and constantly accelerates to catch it.
Grade Levels:	Lowest Grade Level: Not specified Highest Grade Level: Not specified
Standards:	
Language:	USA-English - ISO
Publisher/Owner:	rayyan:MIT
Copyright/Distribution:	System wide - can be used for any courses system wide
Custom Distribution File:	Select Search
Source Distribution:	Open - XML source is open to people who want to use it

Other Types of Problems

Making Symbolic Problems

Symbolic problems are questions that use algebraic expressions instead of numbers in the answers. For instance, you use variables such as *h* for height and *t* for time instead of specifying 4 feet and 6 seconds as you would use in numerical problems.

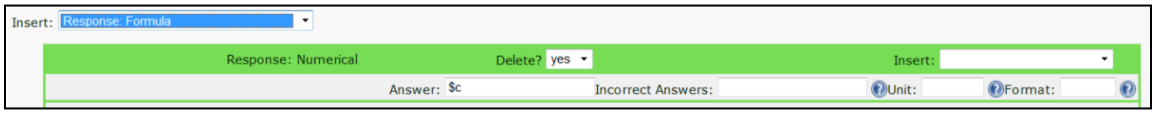
To make a symbolic problem, it is exactly the same as numerical problems, but you have different ways of testing the response. One way of testing the response is to use Maxima, which is a computer algebra system.

Sometimes a better approach is random sampling. By plugging in random values for the variables in both the student's answer and the computer's answer, it's easy to verify that both answers are equivalent. We'll cover how to use both Maxima and random sampling below.

Steps to making a symbolic problem:

1. Start by creating a problem (consider using numerical problem template). Replace the values with symbols.

2. From the Edit representation of the problem, delete the “numerical response” box and add a “formula response” box. Click Save and Edit.

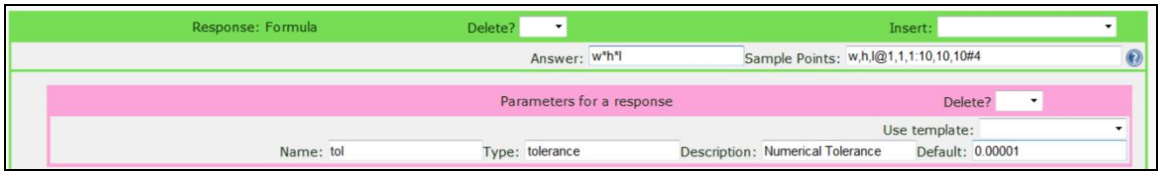


Insert: Response: Formula

Response: Numerical — Delete? yes — Insert:

Answer: $c — Incorrect Answers: — Unit: — Format:

3. In the “formula response” box, add the correct answer. Here, I typed in w*h*l.



Response: Formula — Delete? — Insert:

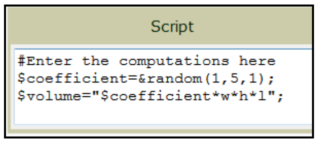
Answer: w*h*l — Sample Points: w,h,l@1,1,1:10,10,10#4

Parameters for a response — Delete?

Use template:

Name: tol — Type: tolerance — Description: Numerical Tolerance — Default: 0.00001

   a. You can also try naming a response in the script as a string (i.e. $volume=“w*h*l”;) and then you can insert this in the answer field. The advantage to this is that the answer can be customized easier to fit a problem. Ex: maybe you randomize coefficients or ask for other answers such as surface area.



Script

```
#Enter the computations here
$coefficient=&random(1,5,1);
$volume="$coefficient*w*h*l";
```

   b. You can put variable names in the answer field. Ex: $coefficient*w*h*l.

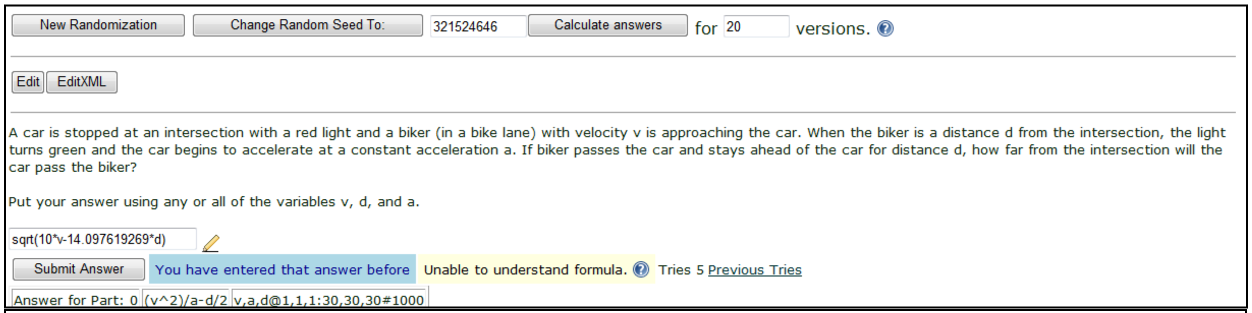
4. If you want to use Maxima, just left leave the sample points field blank.
5. if you want random sampling, type the following in the Sample Points field:
   a. State all of the variables separated by commas
   b. Type @
   c. State the range of values you would like to sample the variables in the following format:
      i. w_low,h_low,l_low:w_high,h_high,l_high
   d. Type #
   e. State how many samples you want to test

   You can also define single points.
6. As always you can change the parameters for a response.

Tip: if you forget the format for the sample points, just click the circled question mark to the right of the Sample Points field.

Errors within LON-CAPA with Interpreting Answers

LON-CAPA’s internal engine uses real numbers, not complex numbers. LON-CAPA thus for example has.difficulty with negative square roots. When sampling, be sure that no expected answers will have negative square roots. You can see the “mysterious” error in LON-CAPA below:


New Randomization | Change Random Seed To: | 321524646 | Calculate answers | for 20 versions.

Edit | EditXML

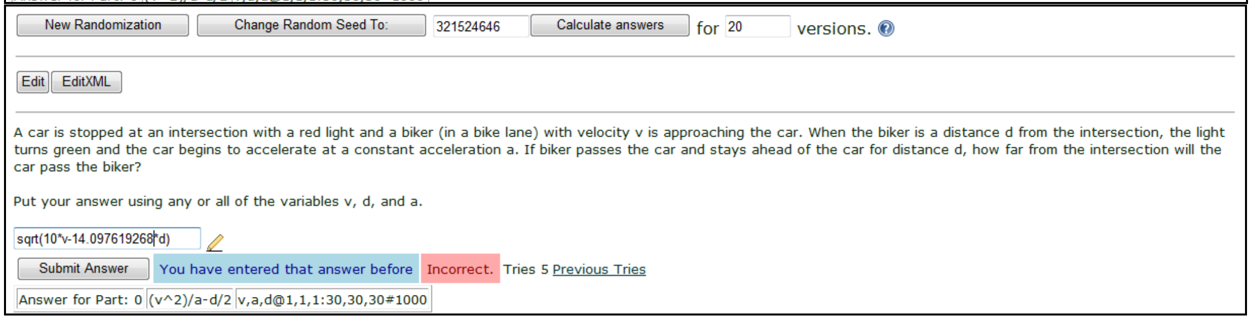
A car is stopped at an intersection with a red light and a biker (in a bike lane) with velocity v is approaching the car. When the biker is a distance d from the intersection, the light turns green and the car begins to accelerate at a constant acceleration a. If biker passes the car and stays ahead of the car for distance d, how far from the intersection will the car pass the biker?

Put your answer using any or all of the variables v, d, and a.

sqrt(10*v-14.097619269*d)

Submit Answer | You have entered that answer before | Unable to understand formula. | Tries 5 Previous Tries

Answer for Part: 0 | (v^2)/a-d/2 | v,a,d@1,1,1:30,30,30#1000


New Randomization | Change Random Seed To: | 321524646 | Calculate answers | for 20 versions.

Edit | EditXML

A car is stopped at an intersection with a red light and a biker (in a bike lane) with velocity v is approaching the car. When the biker is a distance d from the intersection, the light turns green and the car begins to accelerate at a constant acceleration a. If biker passes the car and stays ahead of the car for distance d, how far from the intersection will the car pass the biker?

Put your answer using any or all of the variables v, d, and a.

sqrt(10*v-14.097619268*d)

Submit Answer | You have entered that answer before | Incorrect. | Tries 5 Previous Tries

Answer for Part: 0 | (v^2)/a-d/2 | v,a,d@1,1,1:30,30,30#1000

In these two above pictures LON-CAPA is able to understand the bottom input, but not the top. Ironically the ratio of 10/14.0976… split between accepting the response and not. Anything more negative did not work. As you can see from the sampling certain samples would have negative square roots for the response given.

Tips

Having hints show up after more than one try.

1. In the script box, add $tries=&EXT("user.resource.resource.$external::part.tries");
2. Then make a block condition with the following format:

   <block condition="$tries > 1">
   <startouttext />This is the hint text<endouttext />
   </block >
3. Or, you can allow the course coordinator to adjust this for the whole course by *not* using a block condition.

Add multiple parts to questions

Use part tags <part> and </part>

Show something after the questions is solved or unsolved:

Use solved tags: <solved> and </solved>

<unsolved> and </unsolved> works similarly.

Bold, Italics, horizontal line

Use <b> and </b> for bold

Use<i> and </i> for italics

Use <u> and </u> for underline

Use <hr> and </hr> or just <hr/> for a horizontal line.

Link: using &ext()

In LON-CAPA, visit /res/msu/albertel/test/ext_examples.problem to see a large table of useful parameters you can access within a problem. Many of these are nicely packaged already into functions, which you can find in the “Script Functions” help window.