How to create a Quiz question using STACK

This refers to Moodle 3.11 from August 2022 onward

NOTE: Knowledge of Maxima CAS is required to create questions.

Overview of STACK

STACK is a powerful question type that is integrated into Moodle quizzes. It uses a specially configured version of Maxima and features of the Maxima Computer Algebra System (CAS) to support the creation of mathematical questions and analyse students’ responses. Essentially it can achieve the following:

- Allow students to enter algebraic expressions rather than a value
- Accept multiple responses based on intermediate steps. Evaluate these responses including latter steps even if initial ones were incorrect.
- Provide high quality algorithmically determined feedback based on the students’ responses.

There are three main parts to creating a STACK question: Constructing the questions, creating the answer validation, and building the Potential response tree.

Choose the STACK question type

Choose a question type to add

Choose a STACK question type

Constructing a question

- Enter the "Question name". This is a mandatory field and good practice is to make the name unique and meaningful.
- Enter the "Question variables". These are optional but good practice. Note: A variable name should have at least 2 characters, finish with a semi colon and use a colon rather than the equals sign. e.g. var1: 6; Variables can also be random. e.g. var2: rand (6)
Optional: Random group is optional if you want different questions to use the same random seeds.

- Enter “Question text”. Type in the question text that is presented to the students. It accepts input in both standard text and Maxima CAS. To insert variables into the question requires use of \{@...@\} notation. e.g. What is \{@var1@\} + \{@var2@\}? Note: In the question text box, there are variables which are automatically generated. \texttt{Input} refers to the response box and assigns the student input to the variable \texttt{ans1}. This will be used later in the form. The \texttt{validation} variable is a mathematical interpretation of \texttt{ans1}. These should be left as is unless you want to use a different variable than \texttt{ans1}. If you require a question with multiple inputs then additional input and validation variables are needed.

- Optional: “Specific feedback”. By default it shows the feedback from the response tree. This is a more advanced feature.

- Enter “Default mark” for the question.

- Optional: Add “General feedback”. Depending of quiz settings, this is shown to all students after they have finished. You can put a fully worked out response here.

- Optional. Add an "ID number". This allows you to assign your own ID system to your questions.

- Enter “Question note”. This is compulsory if you use random variables. It has two arguments: questions = answer and is used for checking versions of random questions. it is recommended you first define your question and answer as variables in the "Question variable" field and then enter the Question note in the form \{[@questionvariable@] = [@answervariable@]\}. As long as something is entered the question will work but unless you enter a valid formula you will not be able to check the different variants.

Validating the answer

- Chose the "Input type". Algebraic input is a good general type and accepts algebraic as well as numeric input. The other Input types are advanced features.

- Type in your "Model answer". This can be an expression or good practice suggests defining the correct answer as a Question variable and you can enter the Question variable as the model answer.
Optional: Chose the settings you require for the answer settings.

**Potential response tree**

The potential response tree is where you build the automated feedback to the students. This guide will focus on creating and managing simple behaviour of the nodes.

The Potential response tree is created by through a series of true/false conditions. If a student’s response matches a condition the tree follows the green arm of the branch or if it doesn’t match the tree follows the red arm. When the logic reaches a [stop] command the accompanying feedback is given.

**Node**

Answer test: This is the type of comparison the node will compare the student’s answer against a test condition such as algebraic equivalence, number of significant figures, partial fractions etc. There are a large variety of conditions which are beyond the scope of this document.

SAns: This is where you enter the variable for the students answer and will generally be ans1 from the question text field. If you have set up the question with multiple inputs you will need to chose the correct student answer variable as you will then be creating multiple Potential response trees.

TAns: This is the reference answer you wish to compare the student’s answer against. Typically for the first node it will be the same as the model answer. Many of the reference answers will be typical mistakes students make and so the TAns will be pre-determined incorrect answers.

Test options. Many of the Answer tests require options to be defined. For example, if you are testing for significant figures you will need to define this in the Test options.

Quiet. This option determines whether the feedback is displayed. No means it is not quiet and hence displayed.

**Node when true/false**

These fields control what happens when the true/false condition is met in the node. Most controls are self explanatory or can be accessed through the ? icon. The most important field is Next.

Next field. This determines what happens next after the condition has been met and the score awarded. The feedback is only given if the [stop] is an option. The Next field is used to construct the tree. Choosing another node here automatically appends that node to the current one.