Other Question types

Create multi-select questions

Multi-select (M-S) questions require respondents to identify one or more correct answers in a list of possible answers. Unlike multiple choice (MC) questions, multi-select questions enable you to choose a grading method and allow users to select more than one answer.

To create a multi-select question

1. Navigate to **Quizzes**, **Surveys**, or **Self Assessments**.
2. Click the **Question Library** tab.
3. Click the **New** drop-down menu, select **Multi-Select Question (M-S)**.

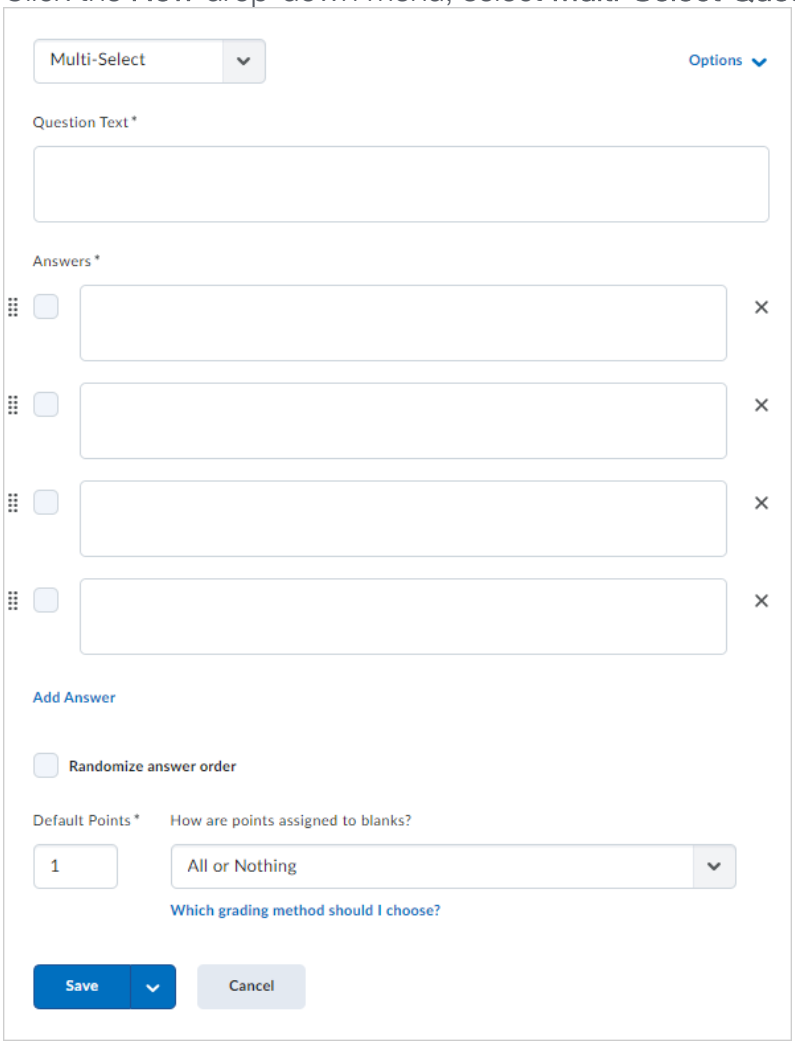

Figure: Enter the required information for your multi-select question.

4. In the **Question Text** field, enter your question. As you select the field, a pop-up toolbar displays for formatting the text and adding images, links, and

graphical equations. As you complete the field, the preview pane displays how it appear to students.

5. Under Answers, enter possible answers. Select the check box for each correct answer.
6. To add optional information to the multi-select question, click Options and choose one or more of the following: Add Feedback, Add Hint, Add Short Description, and Add Enumeration.
   Enter information as required.
7. If you want to allow each student to submit random answers, select the Randomize answer order.
8. Specify a value for Default Points and select one of the following How are points assigned to blanks? options:
   - All or Nothing: Students earn full points only when all correct answers are selected and all incorrect answers are left blank.Students cannot earn partial points.
   - Correct Selections: Points are evenly distributed across all answers. Students earn partial points for each answer correctly selected and left blank.
   - Correct Answers, Limited Selections: Points are evenly distributed across correct answers only. The number of selections allowed is limited to the number of correct answers. Students earn partial points for each correct answer selected.
   - Right Minus Wrong Selections: Points are evenly distributed across all answers. Students earn partial points for each answer correctly selected and left blank. Students lose points for answers incorrectly selected or left blank. Students cannot receive less than 0 points.
9. Click Save.

Create multi-short answer questions

Multi-short answer (MSA) questions require respondents to answer a multi-solution question and input their answers into individual input boxes. Respondent's answers are checked against each possible answer stored in the answer fields. Multi-short answer questions differ from short answer (SA) questions in that the multi-short answer question enables you to create multiple answer boxes which all relate to one answer set; short answer questions also support multiple answer boxes, but each requires a distinct set of possible answers. The short answer question type is ideal if you need to create a multi-part question that cannot share the same answer pool.

For example, the question "Name 3 state capitals" displays three input boxes to users. Each answer users submit is checked against 51 possible correct answers stored in the answer fields and each answer field has a weight of 33.3%.
Note: D2L recommends that the required number of answers corresponds with the

number of input boxes provided. An MSA question's maximum point value is reflected by a 100% weight. As a best practice, D2L recommends that each possible answer's weight calculation equals 100% divided by the number of answers required by the question.

To create multi-short answer questions

1. Navigate to **Quizzes**, **Surveys**, or **Self Assessments**.
2. Click the **Question Library** tab.
3. Click the **New** drop-down menu, select **Multi-Short Answer Question (MSA)**.
4. In the **General** area, enter your multi-short answer question details.
5. In the **Options** area, do the following:
    - In the **Input boxes** field, enter the number of required answer fields you want your question to have. You can assign the number of rows and columns for each answer field.
    - In the **Answer** fields, enter the correct answers for your question, the weight for each answer, and how you want the answers to be evaluated.

      Note: You can set different weights if some solutions are more correct than others.

    - To assign more possible answers for the question, click **Add Answer**.

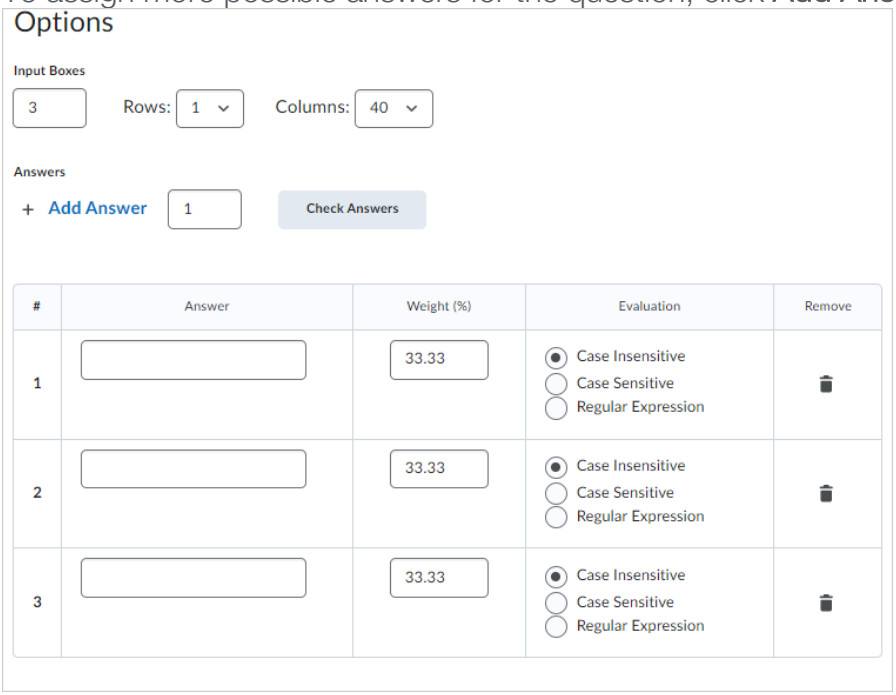

Figure: Set up your multi-short answer questions in the **Options** area.

6. Click **Save** to create your multi-short answer question.

Create fill in the blanks questions

Fill in the blanks (FIB) questions require respondents to fill in one or more missing words for an incomplete sentence, statement, phrase, list, or key terminology.

Note: As a best practice, D2L recommends that answers in blank fields be no more than one or two words to ensure auto-grading accuracy. Your listed order of blank and text fields corresponds with the sequence displayed to users. An FIB question's maximum point value is reflected by a 100% weight. As a best practice, the combined weight of your answers should equal 100%. If your FIB question has multiple blank fields and each blank field has several possible answers, D2L recommends the combined weight of each blank field's most correct answer equals 100%.

To create fill in the blanks questions

1. Navigate to Quizzes, Surveys, or Self Assessments.
2. Click the Question Library tab.
3. Click the New drop-down menu, select Fill in the Blanks Question (FIB).
4. In the General area, enter your fill in the blanks question details.
5. In the Question Text area, do the following:
    - Enter the question text in the Text #1 area.
    - In the Blank #1 area, enter answer information, including the answer to the text, the weight you want to assign the answer (you can set different weights if some solutions are more correct than others), and how you want to evaluate the answer.
    - To assign more possible answers for the blank, click Add Answer.
    - To add more blanks to the question, click Add Blank.

- To add more text to the question, click **Add Text**.

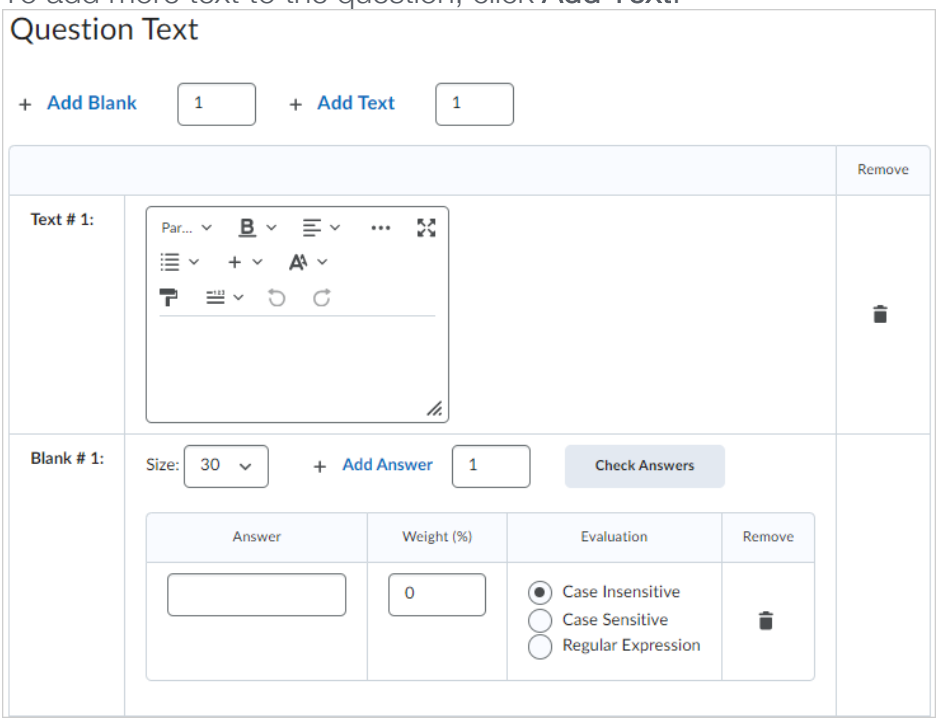

Figure: Set up your fill in the blanks questions in the **Options** area.

Note: Click **Preview** to review how the question is displayed to users.

6. Click **Save** to create your fill in the blanks question.

Create matching questions

Matching (MAT) questions require respondents to choose from a set of possible match choices from drop-down lists and correctly pair them with related items. This question type enables you to assess users' recognition of information and demonstrate comprehension of specific relationships.

To create matching questions

1. Navigate to **Quizzes**, **Surveys**, or **Self Assessments**.
2. Click the **Question Library** tab.
3. Click the **New** drop-down menu, select **Matching Question (MAT)**.
4. In the **General** area, enter your matching question details.
5. In the **Choices** area, do the following:
    - Select the **Grading** method for the question.
    - In each **Value** field, enter a choice.

- To add additional values for the question, click **Add Choice**.

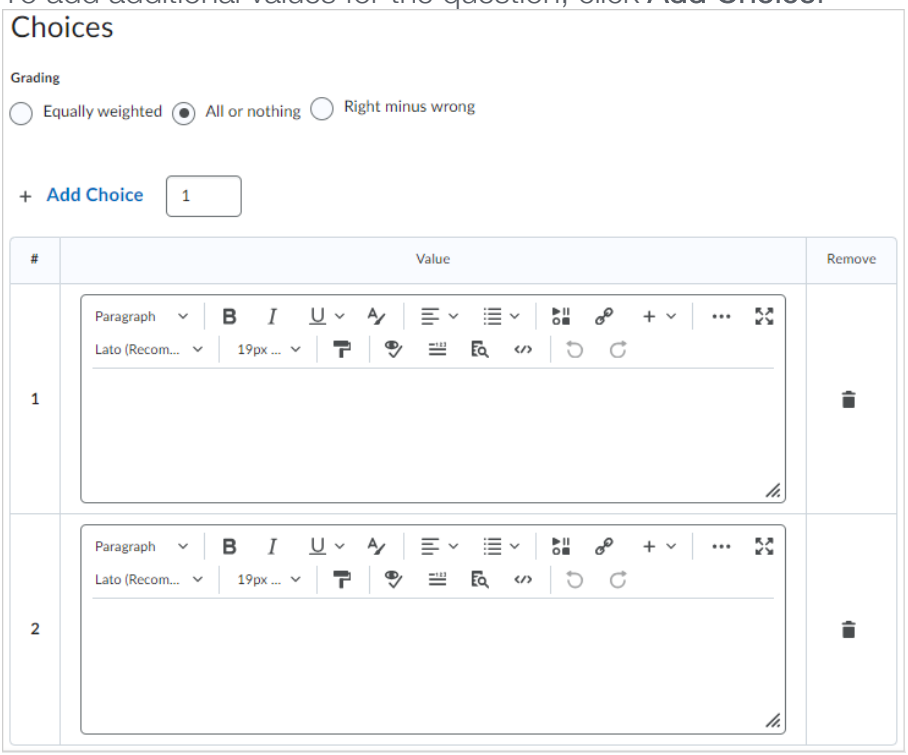

Figure: Create the choices you want to provide to users.

6. In the **Matches** area, do the following:
   - In each **Value** field, enter a matching answer to the choices you created.
   - To add additional matches for the question, click **Add Match**.
   - From the drop-down list for each matching value, select the corresponding **Correct Choice**.

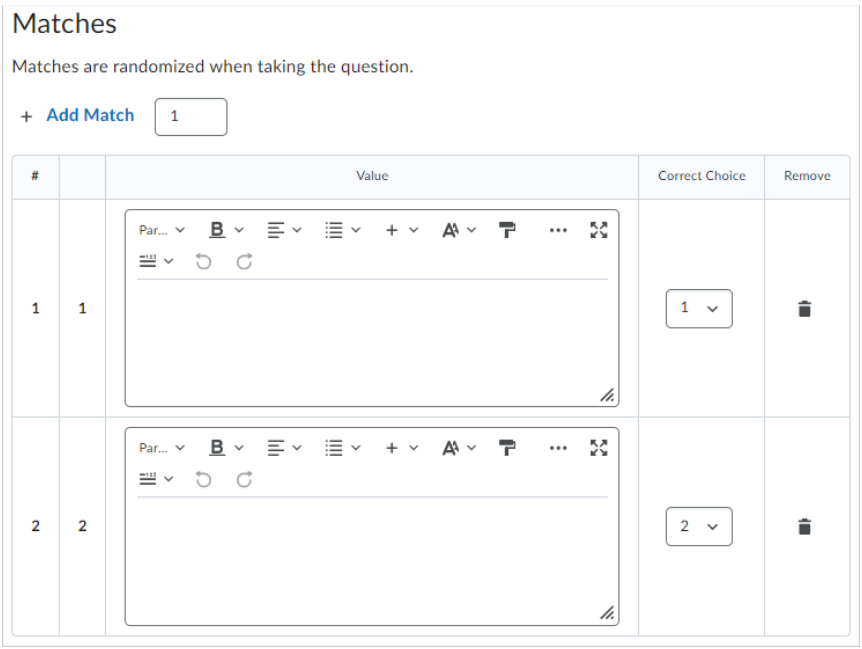

Figure: Create the matching answers to the choices you created.

7. To verify your question, click **Preview**.
8. Click **Save** to create the matching question.

Create ordering questions

Ordering (ORD) questions require respondents to arrange a series of items into a correct sequence or order.

To create ordering questions

1. Navigate to **Quizzes**, **Surveys**, or **Self Assessments**.
2. Click the **Question Library** tab.
3. Click the **New** drop-down menu, select **Ordering Question (ORD)**.
4. In the **General** area, enter your ordering question details.
5. In the **Options** area, do the following:
    - Select your **Grading** method:
        - **Equally weighted** - The total point value is divided equally among all possible correct matches. Users receive equally weighted points for each correct answer.
        - **All or nothing** - Users receive full points for the question if they select all of the correct answers and none of the incorrect answers. Users receive zero points if they miss any correct answers or select any incorrect answers.
        - **Right minus wrong** - Users receive points equal to the number of right answers they choose minus the number of

incorrect answers they choose. To determine how much each answer is worth, the system takes the total number of points assigned to the question and divides it by the total number of answer choices. For example, if a question is worth 10 points and has 5 answer choices, each correct answer is worth 2 points, and each incorrect answer is worth - 2 points (10/5 = 2). If a user gives 3 correct answers and 2 incorrect answers, 2 is the total number of points received for the question [(3*2)+(2*-2)]. Users can receive a minimum of zero on a question; they cannot receive a negative mark.

- In each **Value** field, enter a choice. To add more values, click **Add Item**.
- From the **Correct Order** drop-down list for each value, set the order of the values.
  Note: The first value in the correct order should be **1**.

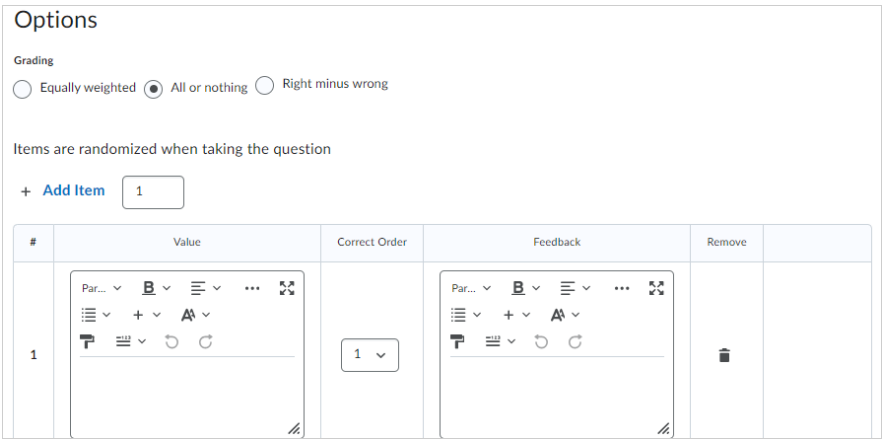

Figure: Set up your ordering question in the **Options** area.

6. To verify your question, click **Preview**.
7. Click **Save** to create your ordering question.

Create arithmetic questions

Arithmetic questions enable you to assess users' knowledge and comprehension of mathematics and number theory. You can ensure each respondent receives a unique question by including variables enclosed with curly braces that randomly generate numbers within the problem. For example, if you set variables x, y, and z with a Min 1 to Max 5 number range in 1-step increments, the question “You have {x} green marbles, {y} red marbles, and {z} blue marbles. How many marbles do you have in total?” will randomly generate a rational number (1, 2, 3, 4, 5) for {x}, {y}, and {z}.

Note: In arithmetic questions, D2L recommends using answer precision to limit the number of acceptable decimal places allowed in a response. You can require that correct answers contain a specific number of decimal places.

To create arithmetic questions

1. Navigate to Quizzes, Surveys, or Self Assessments.
2. Click the Question Library tab.
3. Click the New drop-down menu, select Arithmetic Question (2+2).
4. In the Title field, enter an optional title for your question.
5. In the Points* field, enter the number of points a student will receive for answering the question correctly.
6. In the Question Text field, enter your arithmetic question text.

   Note: The character limit for the Question Text field is 512 characters.

7. To add an image to your arithmetic question, click Insert an Image and select the image you want to add and enter a description for the image.
8. To enable users to add attachments to support their answers, select the Allow attachments to support answers.
9. Enter your Formula using the following supported operations, functions, and constants:

A list of Enumerations and Descriptions to be used in the **Formula**field.

Enumerations	Description
+,-,*,/,\,^	Basic mathematical operators
%	Modulo (remainder) operator
{x}^{y}	x to the power of y
abs({n})	Absolute value of n
cos({n})	The cosine of n (in radians)
sin({n})	The sine of n (in radians)
sqr({n})	The square root of n
tan({n})	The tangent of n (in radians)
log({n})	The log base 10 of n
ln({n})	The log base e of n
atan({n})	The inverse tangent of n

A list of Enumerations and Descriptions to be used in the **Formula**field.

Enumerations	Description
sec({n})	The secant of n
cosec({n})	The cosecant of n
cotan({n})	The cotangent of n
asin({n})	The inverse sin of n (in radians)
acos({n})	The inverse cos of n (in radians)
log2({n})	The base 2 logarithm of n
sinh({n})	The hyperbolic sin of n
cosh({n})	The hyperbolic cos of n
Factorial	Factorial of n, or (n!)
exp	The power of natural log (e)
pi	pi 3.14159 (accurate up to 50 decimal places)
e	e 2.71828 (accurate up to 50 decimal places)

10. Note: To verify your formula before sharing it with students, click Test.

10. From the Answer Precision drop-down menu, select the number of acceptable decimal places allowed in a response and select the enforce precision to require correct answers to contain this specific number of decimal places.
11. Use Tolerance levels to accept near-accurate, estimated, or rounded answers.
12. To assess if answers include correct units of measurement, enter a unit type (mm, cm, grams, inches, etc.), and set one of the following Evaluation Options:
    - Case Insensitive - Auto-grading searches for a matching character pattern in the answer text with or without letter case correctness.
    - Case Sensitive - Auto-grading searches for a matching character pattern in the answer text that must have letter case correctness.
    - Regular Expression - Auto-grading uses meta-characters to search for one or more matching strings in the answer text's character pattern. What you set as meta-character parameters helps determine letter case sensitivity.
13. To assign a weighted points value to the measurement unit, from the Worth: drop-down menu, select a percentage.
14. In the Variables area, do the following:

- Create any variables you want to use with your question by giving the variable a **Name**, a minimum value in **Min**, and a maximum value in **Max**.
- Set the number of decimals to provide in the **Decimal Places** drop-down list.
- To set the system's incrementing steps as it generates numbers from the range set by the **Min** and **Max** fields, in the **Step** field, enter a number.

15. Enter a **Question Hint** and **Question Feedback**.
16. To verify your question, click **Preview**.
17. Click **Save** to create your arithmetic question.

Create significant figures questions

Significant figures questions require respondents to answer in scientific notation and provide solutions that contain a specified number of significant figures. Math and science courses commonly use this question type. You can ensure each respondent receives a unique question by including variables enclosed with curly braces that randomly generate scientific notations within the problem.

To create significant figures question

1. Navigate to **Quizzes**, **Surveys**, or **Self Assessments**.
2. Click the **Question Library** tab.
3. Click the **New** drop-down menu, select **Significant Figures (x10)**.
4. In the **Title** field, enter an optional title for your question.
5. In the **Points** field, enter the number of points a student will receive for answering the question correctly.
6. Set the **Difficulty** level for the question.
7. In the **Question Text** field, enter your arithmetic question text.
8. To add an image to your arithmetic question, click **Insert an Image** and select the image you want to add and enter a description for the image.
9. To enable users to add attachments to support their answers, select **Allow attachments to support answers**.
10. Enter your arithmetic question **Formula** using the following supported operations, functions, and constants:

A list of Enumerations and Descriptions to be used in the **Formula**field.

Enumerations	Description
+,-,*,/,\,^	Basic mathematical operators
%	Modulo (remainder) operator

A list of Enumerations and Descriptions to be used in the **Formula**field.

Enumerations	Description
{x}^{y}	x to the power of y
abs({n})	Absolute value of n
cos({n})	The cosine of n (in radians)
sin({n})	The sine of n (in radians)
sqr({n})	The square root of n
tan({n})	The tangent of n (in radians)
log({n})	The log base 10 of n
ln({n})	The log base e of n
atan({n})	The inverse tangent of n
sec({n})	The secant of n
cosec({n})	The cosecant of n
cotan({n})	The cotangent of n
asin({n})	The inverse sin of n (in radians)
acos({n})	The inverse cos of n (in radians)
log2({n})	The base 2 logarithm of n
sinh({n})	The hyperbolic sin of n
cosh({n})	The hyperbolic cos of n
Factorial	Factorial of n, or (n!)
exp	The power of natural log (e)
pi	pi 3.14159 (accurate up to 50 decimal places)
e	e 2.71828 (accurate up to 50 decimal places)

11. Note: To verify your formula before sharing it with users, click Test.
11. From the Significant Figures drop-down menu, select the number of significant figures the response should contain.
12. From the Deduct drop-down menu, select the percentage of the answer's score to deduct for including incorrect significant figures in a response.
13. Use Tolerance levels to accept near-accurate, estimated, or rounded answers.

14. To assess if answers include correct units of measurement, enter a unit type (mm, cm, grams, inches, etc.), and set one of the following **Evaluation Options**:
    - **Case Insensitive**: Auto-grading searches for a matching character pattern in the answer text with or without letter case correctness.
    - **Case Sensitive**: Auto-grading searches for a matching character pattern in the answer text that must have letter case correctness.
    - **Regular Expression**: Auto-grading uses meta-characters to search for one or more matching strings in the answer text's character pattern. What you set as meta-character parameters helps determine letter case sensitivity.
15. In the **Variables** area, do the following:
    - Create any variables you want to use with your question by giving the variable a **Name**, a minimum value in the **Min** field, and a maximum value in the **Max** field.
    - To set the system's incrementing steps as it generates numbers from the range set by the **Min** and **Max** fields, enter a number in the **Step** field.
16. To verify your question, click **Preview**.
17. Click **Save** to create your significant figures question.

Understanding arithmetic and significant figures question components

In arithmetic questions, use answer precision to limit the number of acceptable decimal places allowed in a response. You can require that correct answers contain a specific number of decimal places.

In significant figure questions, you can select a percentage of the answer's score to deduct for including incorrect significant figures in a response.

Use tolerance levels to accept near-accurate, estimated, or rounded answers.

When rounding, Brightspace automatically applies the Round to Half Even rounding rule when assessing answers that contain decimal places that end with "5". Currently, there are no options to change rounding rules. Applying the Round to Half Even rule, answers with decimal places that end with "5" will round down instead of round up.

Example One: 3.41 * 25 = 85.25

If you create an arithmetic question and set the Answer Precision to 1, the correct answer using Round to Half Even is 85.2.

Example Two: -3.41* 25 = -85.25

If you create an arithmetic question and set the Answer Precision to 1, the correct answer using Round to Half Even is -85.2.

You can enter a unit type (mm, cm, grams, inches, etc.) to assess if answers include correct units of measurement. For significant figures questions, you can select a percentage to assign a weighted points value to the measurement unit. If you use units in your question, you can set the following Evaluation options:

- Case Insensitive - Auto-grading searches for a matching character pattern in the answer text with or without letter case correctness.
- Case Sensitive - Auto-grading searches for a matching character pattern in the answer text that must have letter case correctness.
- Regular Expression - Auto-grading uses meta-characters to search for one or more matching strings in the answer text's character pattern. What you set as meta-character parameters helps determine letter case sensitivity.

Create Likert questions

Create Likert (LIK) questions to measure subjective information such as personal opinions, knowledge, abilities, and attitudes. Likert questions enable you to create surveys that evaluate the intensity of respondents' feelings towards statements presented to them.

Note:Similar to self assessments, all question types you import into Surveys automatically omit point value and difficulty level indicators.

To create Likert questions

1. Navigate to Quizzes, Surveys, or Self Assessments.
2. Click the Question Library tab.
3. Click the New drop-down menu, select Likert Question (LIK).
4. In the General area, enter your Likert question details.
5. In the Questions area, do the following:
    - Select the Scale you want the question to use:
        - One to Five (1 - 5)
        - One to Eight (1 - 8)
        - One to Ten (1 - 10)
        - Agreement Scale (Disagree–Agree)
        - Satisfaction Scale (Dissatisfied–Satisfied)
        - Frequency Scale (Never–Always)
        - Importance Scale (Unimportant–Important)
        - Opposition Scale (Oppose–Support)
    - To give users the option to select not applicable, select Include an N/A option.

- In each **Value** field, enter a statement. To include additional statements, click **Add Option**.

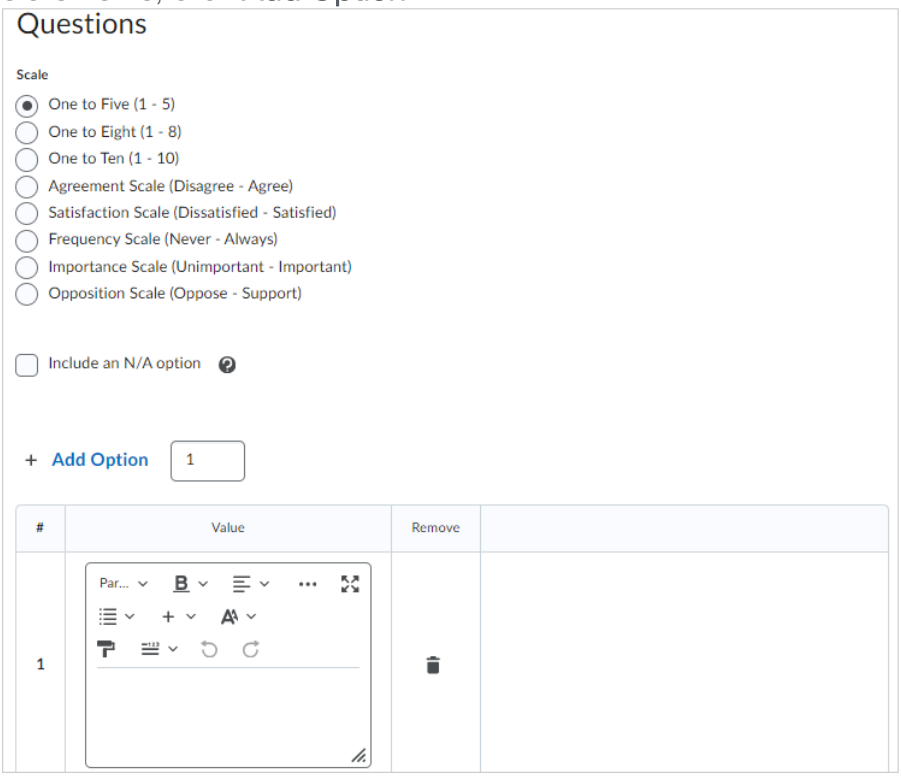

Figure: Set up your Likert question in the **Questions** area.

6. To verify your question, click **Preview**.
7. Click **Save** to create your Likert question.

Understanding regular expressions

Regular expressions give users grading certain question types the ability to evaluate responses against a set of acceptable values. A regular expression uses alpha-numeric and meta-characters to create a pattern that describes one or more strings that must be identically matched within a body of text.

You can use regular expressions in short answer, multi-short answer, arithmetic, significant figures, and fill in the blanks questions.

For example, the fill in the blank question "What word describes red, blue, green, yellow, pink, etc." can use regular expressions for the answer "colou?r*"

Characters used in regular expressions

A list of Characters, Descriptions, and Examples of Regular Expressions.

Character	Description	Example
\	Marks the next character as a special character, a literal, a back-reference, or an octal escape.	The sequence \\ matches \ and **\(** matches **(** **n** matches the character **n** **\n** matches a new-line character
^	Matches the position at the beginning of the input string. If the RegExp object's Multi-line property is set, ^ also matches the position following '\n' or '\r'.	**^cat** matches strings that begin with **cat**
$	Matches the position at the end of the input string. If the RegExp object's Multi-line property is set, $ also matches the position preceding '\n' or '\r'.	**cat$** matches any string that ends with **cat**
*	Matches the preceding character or sub-expression zero or more times. * equals {0,}	**be*** matches **b** or **be** or **beeeeeeeee** **zo*** matches **z** and **zoo**
+	Matches the preceding character or sub-expression one or more times. + equals {1,}.	**be+** matches **be** or **bee** but not **b**
?	Matches the preceding character or sub-expression zero or one time. ? equals {0,1}	**abc?** matches **ab** or **abc** **colou?r** matches **color** or **colour** but not **colouur** **do(es)?** matches the **do** in **do** or **does**
	When this character immediately follows any of the other quantifiers (*, +, ?, {n}, {n,}, {n,m}), the matching pattern is non-greedy. A non-greedy pattern matches as little of the searched string as possible, whereas the default greedy pattern matches as much of	In the string **oooo**, **o+?** matches a single **o**, while **o+** matches all **o**s

A list of Characters, Descriptions, and Examples of Regular Expressions.

Character	Description	Example
	the searched string as possible.	
()	Parentheses create a sub-string or item that you can apply meta-characters to.	**a(bee)?t** matches **at** or **abeet** but not **abet**
{n,}	n is a non-negative integer. Matches at least n times.	**[0-9]{3,}** matches any three digits **o{2,}** does not match the **o** in **Bob**, but matches the two **o**s in **food** **b{4,}** matches **bbbb**
{n}	n is a non-negative integer. Matches exactly n times.	**[0-9]{3}** matches any three or more digits **o{2}** does not match the **o** in **Bob** and matches all the **o**s in **foooood** **o{1}** is equivalent to **o+** **o{0}** is equivalent to **o***
{n,m}	m and n are non-negative integers, where n <= m. Matches at least n and at most m times. **Note:** You cannot put a space between the comma and the numbers.	**[0-9]{3,5}** matches any three, four, or five digits **o{1,3}** matches the first three **o**s in **fooooood** **o{0,1}** is equivalent to **o?** **c{2, 4}** matches **cc**, **ccc**, **cccc**
.	Matches any single character except "\n". To match any character including the '\n', use a pattern such as '[\s\S]'.	**cat.** matches **catT** and **cat2**
(pattern)	Matches pattern and captures the match. The captured match can be retrieved from the resulting Matches collection, using the SubMatches collection in VBScript or the $0$9 properties in JScript. To match parentheses characters ( ), use '\(' or '\)'.	**(jam){2}** matches **jamjam** First group matches **jam**

A list of Characters, Descriptions, and Examples of Regular Expressions.

Character	Description	Example
(?:pattern)	Matches pattern but does not capture the match, that is, it is a non-capturing match that is not stored for possible later use. This is useful for combining parts of a pattern with the "or" character (|).	**industr(?:y|ies)** is a more economical expression than **industry|industries**
(?=pattern)	Positive lookahead matches the search string at any point where a string matching pattern begins. This is a non-capturing match, that is, the match is not captured for possible later use. Lookaheads do not consume characters: after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the lookahead.	**Windows (?=95|98|NT|2000)** matches **Windows** in **Windows 2000** but not **Windows** in **Windows 3.1**
(?!pattern)	Negative lookahead matches the search string at any point where a string not matching pattern begins. This is a non-capturing match, that is, the match is not captured for possible later use. Lookaheads do not consume characters, that is, after a match occurs, the search for the next match begins immediately following the last match, not after the characters that comprised the lookahead.	**Windows (?!95|98|NT|2000)** matches **Windows** in **Windows 3.1** but does not match **Windows** in **Windows 2000**
x|y	Matches x or y.	**July (first|1st|1)** matches **July 1st** but does not match **July 2** **z|food** matches **z** or **food**

A list of Characters, Descriptions, and Examples of Regular Expressions.

Character	Description	Example
		(z|f)ood matches **zood** or **food**
[xyz]	A character set. Matches any one of the enclosed characters.	**gr[ae]y** matches **gray** or **grey** **[abc]** matches the **a** in **plain**
[^xyz]	A negative character set. Matches any character not enclosed.	**1[^02]** matches **13** or **11** but not **10** or **12** **[^abc]** matches every letter except a in **plain**
[a-z]	A range of characters. Matches any character in the specified range.	**[1-9]** matches any single digit except **0** **[a-z]** matches any lowercase alphabetic character in the range **a** through **z**
[^a-z]	A negative range of characters. Matches any character not in the specified range.	**[^a-z]** matches any character not in the range **a** through **z**
\b	Matches a word boundary: the position between a word and a space.	**er\b** matches the **er** in **never** but not the **er** in **verb**
\B	Matches a non-word boundary.	**er\B** matches the **er** in **verb** but not the **er** in **never**
\cx	Matches the control character indicated by x. The value of x must be in the range of A-Z or a-z. If not, c is assumed to be a literal 'c' character.	**\cM** matches a **Control-M** or **carriage return character**
\d	Matches a digit character. Equivalent to [0-9]	
\D	Matches a non-digit character Equivalent to [^0-9]	
\f	Matches a form-feed character. Equivalent to \x0c and \cL	
\n	Matches a new-line character. Equivalent to \x0a and \cJ	

A list of Characters, Descriptions, and Examples of Regular Expressions.

Character	Description	Example
\r	Matches a carriage return character. Equivalent to \x0d and \cM	
\s	Matches any white space character including space, tab, form-feed, etc. Equivalent to [\f\n\r\t\v]	Can be combined in the same way as **[\d\s]**, which matches a character that is a **digit** or **whitespace**
\S	Matches any non-white space character. Equivalent to [^\f\n\r\t\v]	
\t	Matches a tab character. Equivalent to \x09 and \cI	
\v	Matches a vertical tab character. Equivalent to \x0b and \cK	
\w	Matches any word character including underscore. Equivalent to '[A-Za-z0-9_]'	
\W	Matches any non-word character. Equivalent to '[^A-Za-z0-9_]' You should only use \D, \W and \S outside character classes.	
\Z	Matches the end of the string the regular expression is applied to. Matches a position, but never matches before line breaks.	**.\Z** matches **k** in **jol\hok**
\xn	Matches n, where n is a hexadecimal escape value. Hexadecimal escape values must be exactly two digits long. Allows ASCII codes to be used in regular expressions.	**\x41** matches **A** **\x041** is equivalent to **\x04** and **1**

A list of Characters, Descriptions, and Examples of Regular Expressions.

Character	Description	Example
\num	Matches num, where num is a positive integer. A reference back to captured matches.	**(.)\1** matches two consecutive identical characters
\n	Identifies either an octal escape value or a back-reference. If \n is preceded by at least n captured sub-expressions, n is a back-reference. Otherwise, n is an octal escape value if n is an octal digit (0-7).	**\11** and **\011** both match a tab character **\0011** is the equivalent of **1**
\nm	Identifies either an octal escape value or a back-reference. If \nm is preceded by at least nm captured sub-expressions, nm is a back-reference. If \nm is preceded by at least n captures, n is a back-reference followed by literal m. If neither of the preceding conditions exists, \nm matches octal escape value nm when n and m are octal digits (0-7).	
\nml	Matches octal escape value nml when n is an octal digit (0-3) and m and l are octal digits (0-7).	
\un	Matches n, where n is a Unicode character expressed as four hexadecimal digits.	For example, **\u00A9** matches the copyright symbol (©)