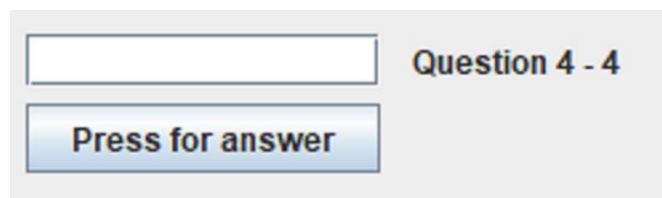


### Mini Project

#### Task 1 [20 Marks]

Your task is to write a basic maths teacher program with a graphical user interface (GUI). This will be called **MathTeacher**. The application will be used by young children to practice simple arithmetic. When the application is executed, it will display an arithmetic question such as “5 + 3”. The child will answer the question (“8” in this case), and be told if the answer is correct or not.

A screen shot of part of the application is shown in Figure 1, where the question is “4 - 4”. The answer is to be entered into the white rectangular text box, and pressing the button labelled “Press for answer” will display the correct answer on a label.



*Figure 1 Screenshot of Example Implementation*

The arithmetic operations allowed are {+, -, x, /}. The arguments are generated automatically and must be integers (whole numbers) in the range 1 to 10 (i.e. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10). The application will then generate arithmetic questions in the general form:

*(argument one) (arithmetic operation) (argument two)*

Correct examples include:

- “5 + 7”
- “9 - 8”
- “1 x 7”
- “6 / 3”

Examples of arithmetic questions that are not allowed include:

- “4.5 + 9.3” (i.e. we are not allowing decimals)
- “-8 + 8” (i.e. we are not allowing negative numbers for the arguments).
- “2 ^ 3” (i.e. we are not allowing operators other than {+, -, \*, /})

For the mini-project, you must implement the **MathTeacher** application, including its Graphical User Interface. We have purposefully left the specification relatively open-ended to allow you to be creative and devise your own approaches. However, the following is a list contains a minimum set of requirements that you must fulfil when writing your mini-project:

- a) The title of the **JFrame** must be ‘Math Teacher’.
- b) The initial text for the question should be a random arithmetic question e.g. “3 \* 4”.
- c) The input text field should be wide enough to display at least THREE characters (i.e. the maximum answer is 100).
- d) The button ‘Press for answer’ must not resize when the GUI is resized. See *Hint 2*.
- e) When first launched, a random arithmetic question will be displayed. Whenever a correct answer is given, the user should be informed. Then a new arithmetic question should be displayed.

- f) Nothing should happen if the user clicks the 'Press for answer' button without entering anything in the text field, i.e. no errors should be thrown in this case.
- g) The `MathTeacher` class must have a `main()` method that launches the application.

*Note: you can assume that only a numeric value will be entered into the text field.*

## Task 2 [5 Marks]

You may notice that entering a non-numeric value and clicking the 'Check!' button will cause a run-time error on the console. Therefore, your second task is to improve the application developed in Task 1 to ensure the user is only allowed to supply valid input values, i.e. a number between 1 and 100 (inclusive). The application must still function as specified in Task 1.

Hint: Use another appropriate component in place of the text field.

Note: All the necessary files (including any reused ones from Task 1) should be placed in a directory called Task2.

## Documentation [5 Marks]

You must include:

- a. Generated Javadocs
- b. Internal comments in your code.
- c. A User Manual. This should be no more than 2 pages and include how to run the program (both how to start and how to use it).

Note: All documentation files should be placed in a directory called Documentation.

## Extension 1 [5 Marks]

Integers are whole numbers such as 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. When doing division, it is possible that the answers are not integers. For example,  $2/4$  is 0.5. How can you restrict the questions asked such that the answer is *always* an integer? For example, we want questions such as  $8/2$  (which equals an integer) and NOT  $2/4$  (which equals a decimal).

Hint: Instead of thinking about  $x/y=z$  (where  $x$  and  $y$  are integers, but  $z$  may not be), think about  $(x*y)/y$  (where  $x$  and  $y$  are integers). This will guarantee the resulting number is an integer

## Extension 2 [5 Marks]

A record of the score should be maintained. Each time the child correctly answers the question, the integer score variable is increased by one. The total number of correct answers should be displayed in a text field. For example, if the child has correctly answered 5 out of 10 questions, the application will display "5 correct out of 10".

Hint: Consider using containers within other containers and using layouts intelligently.

## Extension 3 [5 Marks]

In addition to entering the answer in the textbox directly using the keyboard, the GUI should also have a "calculator key pad" with the digits 0-9. The user now has a choice of entering the answer via the keyboard or via the GUI.

Hint: Consider using Grid layout container.