**Program Development Process**

**Step 1: Requirements specification**

Read the problem to be solved.

State the problem clearly / Understand the problem.

Describe the problem to be solved in your own words here.

We need to write a program that challenges students’ math abilities, specifically multiplication and division. We should limit the program scope to meet the following requirements:

* **Give positive reinforcement when student is correct**
* **Give encouragement when student is incorrect**
* **The computer randomly generates two numbers between 0 and 9, then asks the student for the correct product or quotient**
* **Calculate the percentage of correctly answered questions based on how many questions the student requests**

**Step 2: Analyze**

Describe the data flow and to identify the inputs, outputs and constants.

Identify what the output is first, and then figure out what input data you need.

This list will eventually lead to the list of variables and constants to be defined.

|  |  |  |
| --- | --- | --- |
| Inputs | Outputs | Constants |
| Student answer | Intro describing \* and / | Passing score |
| Exit signal | Question |  |
|  | The integers to be tested |  |
|  | Positive response |  |
|  | Negative response |  |
|  | Overall Score |  |
|  |  |  |

**Step 3: Program design**

Describe the process for obtaining the output from the input.

Note the steps you are performing.

This will lead to the C++ statements you write.

Your algorithm design is described here.

Show your manual calculations (hand work) here.

1. Welcome user to program
2. Explain what \* and / mean
3. Indicate that -1 will exit program
4. Place reset checkpoint
5. Generate question for student:
   1. Determine \* or / randomly
      1. questionType = Rand() % 2 + 1  
         if questionType > 5 then \*  
         else if questionType < 5 && > 0 then /
   2. Deterimine integers for the expression (a \* b = c or a / b = c)
      1. A = rand() % 10
      2. B = rand() % 10
      3. If dividing
         1. If b == 0
            1. B = b+1
            2. We cannot divide by zero
   3. Determine answer to the expression
      1. C = a \* b or c = a / b
   4. Output question to student
      1. questionsAsked++
   5. Receive input
   6. Test input against c
      1. If input != c
         1. incorrectAnswer++
      2. while input != c
         1. Output encouragement (use switch statement per requirements!), ask for another response
      3. Output positive reinforcement (again, switch statement!) for correct response
   7. Generate another question for the student until exit signal is sent
6. Upon exit signal, calculate student’s score
   1. correctAnswers = questionsAsked – incorrectAnswer
      1. We will not be penalizing student for multiple incorrect answers on the same question.
   2. score = correctAnswers / questionsAsked
   3. if score >= .75
      1. output congrats
   4. else
      1. output recommendation to seek additional help
7. Ask if another student wishes to use the program
   1. If yes, return to reset checkpoint
   2. If no, thank user and exit

**Step 4: Implementation in C++**

Also known as coding.

Develop a C++ solution using your work from step 3.

Write the declarations first and then write the C++ statements.

Enter and debug program on the computer.

Show your source code here.

Source code will be sent in a separate file.

**Step 5: Testing**

Test your program with sample data set to make sure the output is correct.

Should test multiple data sets including the boundary cases.

Summary and analyze your result.

Show the output screen shots here.

Screenshot will be sent in a separate file.