**CS1180**

**Project 2**

**Purpose:** This program will provide experience using decision statements and loops.

You have just been hired as a teacher’s aide in the WSU School of Math, grade 3. Your students need extra practice with their arithmetic skills, and you have been given the task of writing a computer program that will allow them to practice on their own.

To help you understand how the program should work, **a sample program run is shown on the next several pages**. The basic algorithm is described below.

1. Display the menu shown on the sample program run and get the user’s choice. If the user does not enter a valid choice, output an appropriate message and prompt for another choice.
2. Display 5 problems of the user-requested type. For instance, if the user chooses option 1, your program should display an addition problem, using randomly-generated operands between 1 and 10, then it should get the user’s answer, then it should check the answer and output an appropriate message based on its correctness (if the answer was incorrect, the message should include the correct answer). The display problem/get answer/check answer steps should repeat 5 times.
3. Repeat steps 1 and 2 until the user chooses to exit.
4. Keep track of the number of problems attempted and the number of problems correct. When the user has completed all their practice problems, output the number of problems correct, the number of problems tried, and the percentage of problems correct to 2 decimal places.

Other requirements:

* If the user chooses subtraction, be sure that the second operand is never greater than the first operand, so that negative answers are not required.
* You must use a while loop to repeat the menu display/get user choice action.
* You must use a for loop to repeat the display problem/get answer/check answer steps
* Avoid using methods! While they are usually desirable, the focus in this project is decision statements and loops; the use of methods will begin with project 3.

**Generating random numbers:**

Use the following formula to generate random numbers.

(int)(lowerBound + Math.random() \* (upperBound – lowerBound + 1)

You may use literals in place of the upper and lower bounds. For instance, to display values between 1 and 10 (inclusive on both ends), the formula would look like this:

(int)(1 + Math.random() \* 10)

**The grading rubric follows on the last page of this document.**

**Sample program run:**

run:

Welcome to the Wright State School of Math!

Please choose one of the following options for your math quiz:

1. Addition with numbers 1-10

2. Addition with numbers 1-100

3. Subtraction with numbers 1-10

4. Subtraction with numbers 1-100

5. Multiplication with numbers 1-10

6. Exit the program

2

Enter the answer to the following problem:

12 + 37

49

That is the correct answer!

Enter the answer to the following problem:

47 + 90

137

That is the correct answer!

Enter the answer to the following problem:

77 + 67

34

Sorry, that is incorrect. The correct answer is 144

Enter the answer to the following problem:

42 + 14

55

Sorry, that is incorrect. The correct answer is 56

Enter the answer to the following problem:

40 + 46

86

That is the correct answer!

Please choose one of the following options for your math quiz:

1. Addition with numbers 1-10

2. Addition with numbers 1-100

3. Subtraction with numbers 1-10

4. Subtraction with numbers 1-100

5. Multiplication with numbers 1-10

6. Exit the program

8

Valid choices are 1-6; please re-enter

7

Valid choices are 1-6; please re-enter

5

Enter the answer to the following problem:

6 \* 3

18

That is the correct answer!

Enter the answer to the following problem:

4 \* 2

8

That is the correct answer!

Enter the answer to the following problem:

8 \* 1

8

That is the correct answer!

Enter the answer to the following problem:

6 \* 6

12

Sorry, that is incorrect. The correct answer is 36

Enter the answer to the following problem:

3 \* 6

18

That is the correct answer!

Please choose one of the following options for your math quiz:

1. Addition with numbers 1-10

2. Addition with numbers 1-100

3. Subtraction with numbers 1-10

4. Subtraction with numbers 1-100

5. Multiplication with numbers 1-10

6. Exit the program

3

Enter the answer to the following problem:

8 - 3

5

That is the correct answer!

Enter the answer to the following problem:

10 - 1

9

That is the correct answer!

Enter the answer to the following problem:

3 - 2

1

That is the correct answer!

Enter the answer to the following problem:

9 - 5

4

That is the correct answer!

Enter the answer to the following problem:

7 - 2

5

That is the correct answer!

Please choose one of the following options for your math quiz:

1. Addition with numbers 1-10

2. Addition with numbers 1-100

3. Subtraction with numbers 1-10

4. Subtraction with numbers 1-100

5. Multiplication with numbers 1-10

6. Exit the program

6

You got 12 problems correct out of 15 problems attempted. That is 80.00 percent correct.

Goodbye!

BUILD SUCCESSFUL (total time: 1 minute 47 seconds)

**Grading (100 pts total):**

(4) Neatly and correctly displays menu

(5) Gets a **valid** user choice

(5) Uses a **switch statement** to process the user’s choice

(50) (10 points per choice) For each choice the user makes, the program displays 5 problems. For each problem, the program asks the user for his/her answer, then displays a message indicating the correctness of that answer. If incorrect, the program displays the correct answer as part of the message.

Note: **-5 points for each choice if a for loop is not used** to display the problems, even if this part of the program works correctly

(10) After each set of problems is processed, the menu is again displayed and a valid choice is input.

(8) The program exits correctly when the user chooses option 6 (displays the number correct, the number tried, and the percentage correct, and an appropriate “goodbye” message)

(5) Uses proper Javadoc comments above and below the package name and above main

(4) Uses single-line comments above each major step

(4) Variables are meaningfully-named

(5) Code is properly formatted (i.e. indenting, alignment of braces)