Task 1:

Link to GitHub repository:
https://github.com/Wolff11912/dsss_homework_2.git

Task 3:

```python
import random


def randomInteger(min, max):
    """
    Calculates a random integer between a given minimum and maximum (minimum and maximum are included).

    Args:
        min (int): a given minimum.
        max (int): a given maximum.

    Returns:
        int: a random number between a given minimum and maximum (minimum and maximum are included).
    """
    try:
        # Check if both arguments are int values
        if not isinstance(min, int) or not isinstance(max, int):
            raise ValueError("Both min and max must be integers.")

        #Check if min is smaller than max
        if min >= max:
            raise ValueError("min must be less than max.")

        return random.randint(min, max)

    except ValueError as e:
        print(f"Error: {e}")
        return None
```

```python
def randomOperator():
    """
    Picks an arithmetic operator randomly.

    Returns:
        char: a randomly chosen arithmetic operator between +, -, *
    """
    return random.choice(['+', '-', '*'])


def calculator(num1, num2, operator):
    """
    Calculates a solution based on the given numbers and the operator.

    Args:
        num1 (int): First given integer.
        num2 (int): Second given integer.
        operator (char): Operator for calculation.

    Returns:
        prob (str): The calculation that is done.
        sol (int): The calculated solution.

    """
    prob = f"{num1} {operator} {num2}" #convertes the calculation into a string
    if operator == '+':
        sol = num1 + num2  #if operator is '+' the sum of the two numbers is gererated
    elif operator == '-':
        sol = num1 - num2  #if operator is '-' the difference of the two numbers is gererated
    else:
        sol = num1 * num2  #if operator is '*' the product of the two numbers is gererated
    return prob, sol
```

```python
def math_quiz():
    score = 0   #score of the user
    rounds = 3  #the number of rounds that are played

    print("Welcome to the Math Quiz Game!")
    print("You will be presented with math problems, and you need to provide the correct answers.")

    for _ in range(rounds):
        number1 = randomInteger(-6, -1);
        number2 = randomInteger(2, 5);
        operator = randomOperator()

        PROBLEM, ANSWER = calculator(number1, number2, operator)
        print(f"\nQuestion: {PROBLEM}")

        try:
            useranswer = int(input("Your answer: "))
        except ValueError:
            print("Invalid input! Please enter an integer.")
            continue  # Start next round if user input is invalid

        if useranswer == ANSWER:
            print("Correct! You earned a point.")
            score += 1  #if the answer of the user is correct the score is increased by 1
        else:
            print(f"Wrong answer. The correct answer is {ANSWER}.")

    print(f"\nGame over! Your score is: {score}/{rounds}")

if __name__ == "__main__":
    math_quiz()
```

```
PS C:\Users\Norbert\DataScienceSurvival\Exercise2\local\dsss_homework_2> git merge code_cleanup
Updating 1257f64..ad03250
Fast-forward
 math_quiz/math_quiz.py       | 79 +++++++++++++++++++++++++++++++++++----------
 math_quiz/tests_math_quiz.py | 26 +++++++++------
 2 files changed, 78 insertions(+), 27 deletions(-)
PS C:\Users\Norbert\DataScienceSurvival\Exercise2\local\dsss_homework_2>
```

Task 4:

```python
import unittest
from math_quiz import randomInteger, randomOperator, calculator


class TestMathGame(unittest.TestCase):

    def test_randomInteger(self):
        # Test if random numbers generated are within the specified range
        min_val = 1
        max_val = 10
        for _ in range(1000):  # Test a large number of random values
            rand_num = randomInteger(min_val, max_val)
            self.assertTrue(min_val <= rand_num <= max_val)

    def test_randomOperator(self):
        # Test if one of the listed operators is picked
        for _ in range(1000):
            picked_operator = randomOperator()
            self.assertTrue(picked_operator in ['+', '-', '*'])

    def test_calculator(self):
            test_cases = [
                (5, 2, '+', '5 + 2', 7),
                (9, 4, '-', '9 - 4', 5),
                (8, 3, '*', '8 * 3', 24),
                (4, -6, '+', '4 + -6', -2),
                (-3, -7, '-', '-3 - -7', 4),
                (-2, 10, '*', '-2 * 10', -20)
            ]

            for num1, num2, operator, expected_problem, expected_answer in test_cases:
                problem, answer = calculator(num1, num2, operator)
                self.assertTrue((expected_problem == problem) and (answer == expected_answer))

if __name__ == "__main__":
    unittest.main()
```

Task 5:

```
PS C:\Users\Norbert\DataScienceSurvival\Exercise2\local\dsss_homework_2> pip install git+https://github.com/Wolff11912/dsss_homework_2.git
Collecting git+https://github.com/Wolff11912/dsss_homework_2.git
  Cloning https://github.com/Wolff11912/dsss_homework_2.git to c:\users\norbert\appdata\local\temp\pip-req-build-3nkrfe_i
  Running command git clone --filter=blob:none --quiet https://github.com/Wolff11912/dsss_homework_2.git 'C:\Users\Norbert\AppData\Local\Temp\pip-req-build-3nkrfe_i'
  Resolved https://github.com/Wolff11912/dsss_homework_2.git to commit b3b52be578bc45e3ea47e5d262c529bb001ed3d6
  Preparing metadata (setup.py) ... done
Collecting requests (from dsss_homework_2==0.1)
  Downloading requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Requirement already satisfied: numpy in c:\users\norbert\miniconda3\envs\dsss\lib\site-packages (from dsss_homework_2==0.1) (1.26.4)
Collecting charset-normalizer<4,>=2 (from requests->dsss_homework_2==0.1)
  Downloading charset_normalizer-3.4.0-cp312-cp312-win_amd64.whl.metadata (34 kB)
Collecting idna<4,>=2.5 (from requests->dsss_homework_2==0.1)
  Downloading idna-3.10-py3-none-any.whl.metadata (10 kB)
Collecting urllib3<3,>=1.21.1 (from requests->dsss_homework_2==0.1)
  Downloading urllib3-2.2.3-py3-none-any.whl.metadata (6.5 kB)
Collecting certifi>=2017.4.17 (from requests->dsss_homework_2==0.1)
  Downloading certifi-2024.8.30-py3-none-any.whl.metadata (2.2 kB)
Downloading requests-2.32.3-py3-none-any.whl (64 kB)
Downloading certifi-2024.8.30-py3-none-any.whl (167 kB)
Downloading charset_normalizer-3.4.0-cp312-cp312-win_amd64.whl (102 kB)
Downloading idna-3.10-py3-none-any.whl (70 kB)
Downloading urllib3-2.2.3-py3-none-any.whl (126 kB)
Building wheels for collected packages: dsss_homework_2
  Building wheel for dsss_homework_2 (setup.py) ... done
  Created wheel for dsss_homework_2: filename=dsss_homework_2-0.1-py3-none-any.whl size=7648 sha256=57b30f41c626b13f1d2d10b6a80feaf2700ea720696b805143c6420c1836f14d
  Stored in directory: C:\Users\Norbert\AppData\Local\Temp\pip-ephem-wheel-cache-w7xc7j4f\wheels\42\bb\0b\6f18120153b6c3ad1fad34300f286fc8fd8e0d9d1a8620ead1
Successfully built dsss_homework_2
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests, dsss_homework_2
Successfully installed certifi-2024.8.30 charset-normalizer-3.4.0 dsss_homework_2-0.1 idna-3.10 requests-2.32.3 urllib3-2.2.3
PS C:\Users\Norbert\DataScienceSurvival\Exercise2\local\dsss_homework_2>
```

Name: Constantin Wolff
Mat. Num.: 22442020
IdM: lu11synu