

Task 1:  
[https://github.com/IEEn-Lee/DSSS\\_WS24-25\\_exercise02.git](https://github.com/IEEn-Lee/DSSS_WS24-25_exercise02.git)

Task 3:

```
D:\FAU\FAU_course\ws24_25\DSSS\exercise\DSSS_WS24-25_exercise02>git merge code_cleanup
Updating d728f59..5eed944
```

Fast-forward

```
math_quiz.py | 85 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1 file changed, 85 insertions(+)
create mode 100644 math_quiz.py
```

```
def random_integer(min, max):
    """
    Generate a random integer between two given numbers.

    Args:
        min (int): The minimum value (inclusive).
        max (int): The maximum value (inclusive).

    Returns:
        int: A random integer between min and max.
    """
    return random.randint(min, max)

def random_operator():
    """
    Choice a random mathematical operator within +, -, *.

    Returns:
        str: A random mathematical operator within +, -, *.
    """
    return random.choice(['+', '-', '*'])

def math_operation(num1, num2, operator):
    """
    Give the problem and calculate the expected answer based on the operator.

    Args:
        num1 (int): The first given number.
        num2 (int): The second given number.
        operator (str): The given operator.

    Returns:
        tuple: the tuple containing the problem as the string and the answer as int.
    """
    problem = f"{num1} {operator} {num2}"
    if operator == '+':
        answer = num1 + num2
    elif operator == '-':
        answer = num1 - num2
    else:
        answer = num1 * num2
    return problem, answer
```

```
def math_quiz():
    """
    Generate random math problems, the user give the answer, and will receive the feedback.
    """
    score = 0
    total_question = 3

    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(total_question):
        num1 = random_integer(1, 10)
        num2 = random_integer(1, 5)
        operator = random_operator()

        PROBLEM, ANSWER = math_operation(num1, num2, operator)
        print(f"\nQuestion: {PROBLEM}")

        try:
            useranswer = input("Your answer: ")
            useranswer = int(useranswer)
        except ValueError:
            print("Invalid input. Please enter an integer.")
            continue

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

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

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

Task 4:

```
import unittest
from math_quiz import random_integer, random_operator, math_operation

class TestMathGame(unittest.TestCase):

    def test_random_integer(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 = random_integer(min_val, max_val)
            self.assertTrue(min_val <= rand_num <= max_val)

    def test_random_operator(self):
        # TODO
        # Test if the operator is among the expected options.
        operators = set(['+', '-', '*'])
        for _ in range(1000):
            operator = random_operator()
            self.assertIn(operator, operators)

    def test_math_operation(self):
        # Test if the math_operation function returns the correct problem and answer
        test_cases = [
            (5, 2, '+', '5 + 2', 7),
            (5, 2, '-', '5 - 2', 3),
            (5, 2, '*', '5 * 2', 10),
            (1, 10, '+', '1 + 10', 11),
            (1, 10, '-', '1 - 10', -9),
            (1, 10, '*', '1 * 10', 10),
        ]

        for num1, num2, operator, expected_problem, expected_answer in test_cases:
            # TODO
            problem, answer = math_operation(num1, num2, operator)
            self.assertEqual(problem, expected_problem)
            self.assertEqual(answer, expected_answer)

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

Task 5:

```
D:\FAU\FAU_course\ws24_25\DSSS\exercise\DSSS_WS24-25_exercise02>pip install git+https://github.com/IEEn-Lee/DSSS_WS24-25_exercise02.git
Collecting git+https://github.com/IEEn-Lee/DSSS_WS24-25_exercise02.git
  Cloning https://github.com/IEEn-Lee/DSSS_WS24-25_exercise02.git to c:\users\user\appdata\local\temp\pip-req-build-qd_nojnv
  Running command git clone --filter=blob:none --quiet https://github.com/IEEn-Lee/DSSS_WS24-25_exercise02.git 'C:\Users\USER\AppData\Local\Temp\pip-req-build-qd_nojnv'
  Resolved https://github.com/IEEn-Lee/DSSS_WS24-25_exercise02.git to commit 7e664alc449f7897315fa19e8e8c0ab0dcdcl868
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: math_quiz
  Building wheel for math_quiz (pyproject.toml) ... done
  Created wheel for math_quiz: filename=math_quiz-0.1-py3-none-any.whl size=2190 sha256=3ae45852bfcdcf95248658a9046854fb034a974698093fa9c466ab8b4678dd3a
  Stored in directory: C:\Users\USER\AppData\Local\Temp\pip-ephem-wheel-cache-xl9pmtgp\wheels\d4\02\d1\26f332cba8d98701df9f058ceb8357e2da458bfc349383d369
Successfully built math_quiz
Installing collected packages: math_quiz
Successfully installed math_quiz-0.1
```