ET574 Homework 4: Dice Game Simulator

Objective:
In this assignment, you will implement a simple dice game using object-oriented programming in Python. You will define classes, create a playable console program, and write unit tests using the unittest module.

Game Rules

Total	Outcome
7 or 11	Win
2, 3, or 12	Lose
4–6, 8–10	Roll Again

The game uses two 6-sided dice. Each round, the player rolls both dice and their sum is evaluated based on the rules above. The player can continue playing rounds until they choose to exit.

Program Requirements

1. Classes
You must implement the following classes:

- Die class:

- __init__(self, sides=6) – initializes with a default of 6 sides.

- roll(self) – returns a random integer from 1 to the number of sides.

- DiceGame class:

- __init__(self) – creates two Die objects.

- play_round(self) – rolls both dice and returns the two values, total, and result.

- evaluate_roll(self, total) – determines the outcome based on the total.

2. main() Function
Implements a simple console menu:
1. Play a round
2. Exit
Displays the two dice rolled, their total, and the game result.

3. Unit Testing
A starter file test_dicegame.py will be provided. It includes 3 test cases:
- Die roll bounds
- Outcome evaluation (Win/Lose/Roll Again)
- Result structure of play_round()

You must add at least 3 of your own unit tests to verify functionality.

4. Repository and Version Control
You must create your own GitHub repository for this assignment and use version control throughout development. The repository name must follow this naming format exactly:

<Course Name>-<Section Name>-<HW4>-<Your Last Name>

For example, ET574-K-HW4-SMITH (ET 574 Section K Homework 4 for Smith)

After creating the repository, add the instructor as a collaborator. Your commit history will be reviewed as proof of your work.

Files You Will Work With
- main.py – You will write the full game code here
- test_dicegame.py – Add your unit tests here

What to Submit
1. main.py – Complete, working program
2. test_dicegame.py – includes at least 3 additional test cases
3. File Header – A brief comment at the top of each file with your name and QCC ID
4. PDF Submission - The PDF must include the following:

- Codespaces Environment
Screenshot showing the repository open in GitHub Codespaces (browser view with file tree visible).

- Source Code
Screenshots of the final source code for the assignment in Codespaces.

- Commit History
Screenshot of the commit history tab in GitHub showing:
Multiple commits over time, with clear and descriptive commit messages that match your work.

- Program Output / Results
Screenshot(s) showing the program running successfully:
Terminal output with test results.