What is Unit Testing?

Unit testing is a software testing method in which individual components, or "units," of a program are tested in isolation to ensure they function as expected. A "unit" is typically the smallest testable part of an application, such as a function, method, or class. In this case, the units are the methods of the Yatzy class (e.g., Ones(), TwoPairs(), Yatzy()).

- **Key Characteristics:**
  - **Isolation:** Each unit is tested independently of other parts of the program. For example, testing TwoPairs() doesn't rely on the game's full logic, just the dice values.
  - **Automation:** Unit tests are usually automated, meaning they can be run repeatedly without manual intervention (e.g., via GitHub Actions in your assignment).
  - **Granularity:** Focuses on specific functionality, not the entire system.
- **Purpose:**
  - Verify that each method behaves correctly for given inputs.
  - Catch bugs early in development.

Why is Unit Testing Important?

1. **Reliability:** Ensures your Yatzy class methods return correct scores (e.g., Yatzy() returns 50 only when all dice are identical).
2. **Maintainability:** Makes it easier to update code (e.g., fixing TwoPairs() in Worksheet 3) without introducing new errors, as tests confirm the fix works.
3. **Automation in CI/CD:** In Worksheet 2, GitHub Actions runs your tests automatically on every push, ensuring consistent quality.
4. **Documentation:** Tests act as examples of how methods should work (e.g., test_small() shows Small() expects a 1-2-3-4-5 sequence).
5. **Collaboration:** In Worksheet 3, unit tests help your classmate verify your fix resolves their reported issue.

How Unit Testing Applies

For Worksheet 2, you're tasked with:

- Creating a Yatzy class with specific methods.
- Writing tests for all methods (e.g., Ones(), TwoPairs(), etc.).
- Automating these tests with GitHub Actions.