

MODULE 1: INTRODUCTION TO PROGRAMMING

Test Driven Development





Yesterday

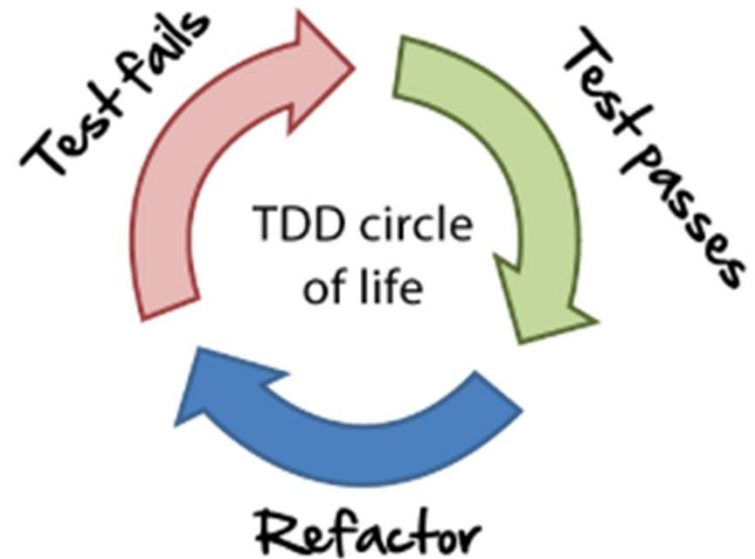
- What is SDLC?
- What are the two main types of SDLC?
- What is one pro of manual testing?
- What is one con of manual testing?
- What is one pro of automated testing.
- What is one con of automated testing?
- Who does the testing?

Testing First



Test Driven Development

1. Write failing test
2. Make test pass with minimum code
3. Refactor



Refactoring

- **Refactoring** introduces modifications to the code intended to improve the structure or design without changing functionality.
- Ways to refactor:
 - Eliminate duplicate code
 - Extract a method by breaking down long difficult methods
 - Extract complex operations to variables
 - Introduce constants for magic numbers
 - Simplify conditional expressions

TDD Benefits

- Program for specific conditions
- Baby steps towards solving a problem, incrementally adding code
- Refactoring with new patterns
- Previous tests hold value with each modification

General Strategy

1. Create a list of tests needed.
2. Start by writing just enough test code
3. Always run the test to see it fail in the way you expect
4. Write enough code to make the test build
5. Write enough code to make that test pass (possibly by faking it).
6. When the implementation is obvious then that can be typed in but go back to #4 if not
7. Generalizing the code by eliminating code duplication or reducing dependencies

LET'S CODE!



ELEVATE  YOURSELF

WHAT QUESTIONS DO
YOU HAVE?



Reading for tonight:

Error Handling

