BridgeLabz

Employability Delivered

Quantity Measurement TDD Problem

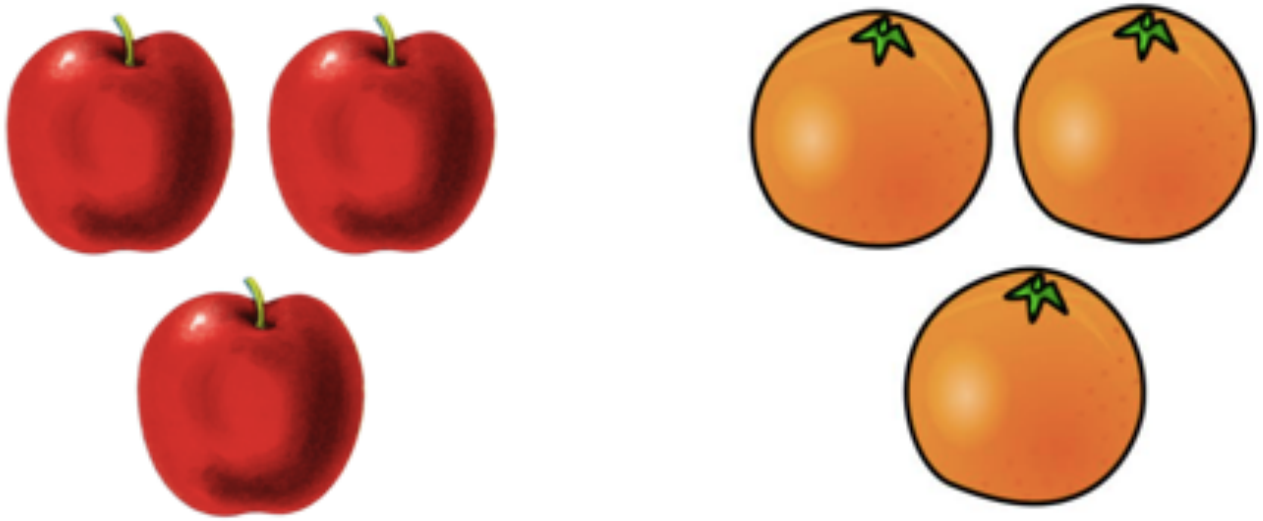
Emphasis on

Rules

- Start with Welcome message in the Main Branch
- Every Use case should have its own commit
- Every Test Case should have its own Commit
- Every Refactor also should have its own commit
- Follow Follow Programming Hygiene with proper and consistent naming and indentation convention
- Follow DRY principle and Refactor Code

Quantity Measurement Problem

3 != 3



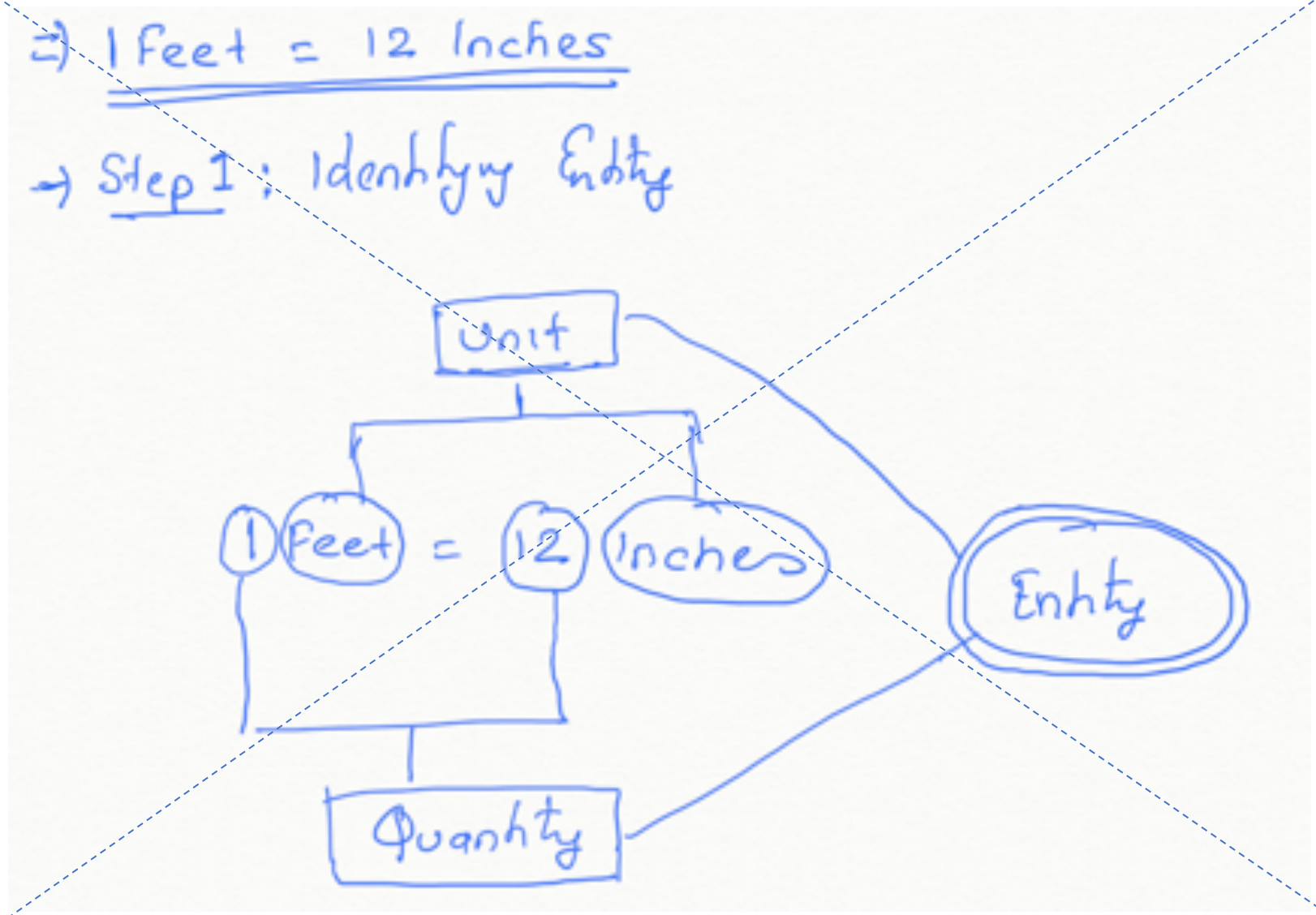
UC 1

As a math student, I wish to compare lengths

1ft = 12in

Quantity Measurement – Equal Lengths

BridgeLabz
Employability Delivered



DDT
Vs
TDD

Design Development & Test way of thinking starts with identifying Entities is opposite of TDD.

Lets Restart

What is TDD???

- Three Laws of TDD
  - You aren’t allowed to write any production code until you have written a failing unit test
  - You aren’t allowed to write more of a unit test than it is sufficient to fail. And not compiling is failing.
  - You aren’t allowed to write more production code than is sufficient to pass the currently failing unit test.

Simplest Test Case

BridgeLabz
Employability Delivered

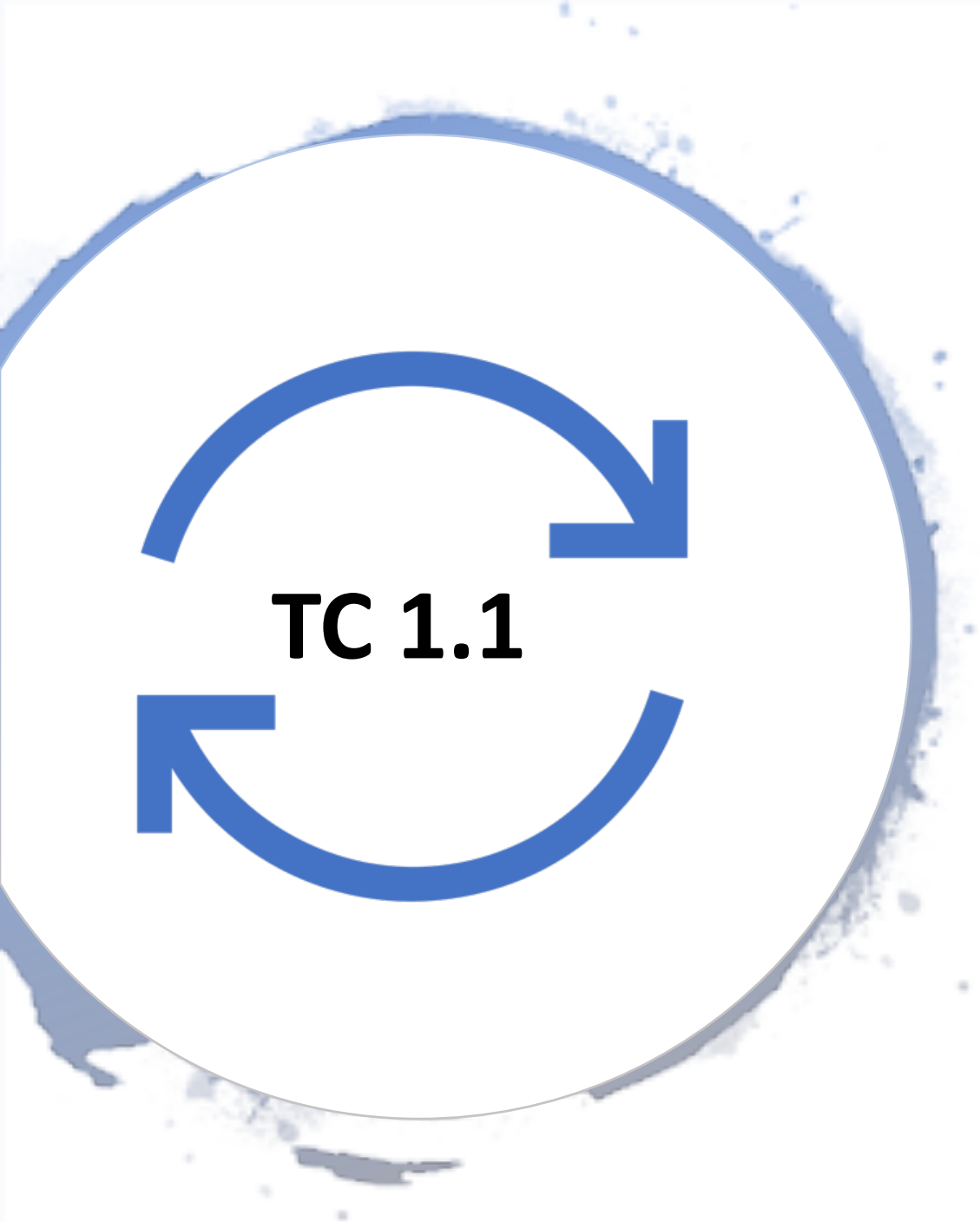
⇒ 1 ft = 12 in

Step1 : Do the Simplest thing that Works. i.e Attempt first with only One Variable at a time.

1 ft = 12 in

Start with Foot

⇒ Perform test cases for Foot Equality.



Given 0 Feet and 0 Feet Should Return equal

Compare vs Equality Check

NOTE : Compare Vs Equal

Compare → Less, Equal, more

Equal → True, False

⇒ Goal is Equality check hence call use or override equals method.

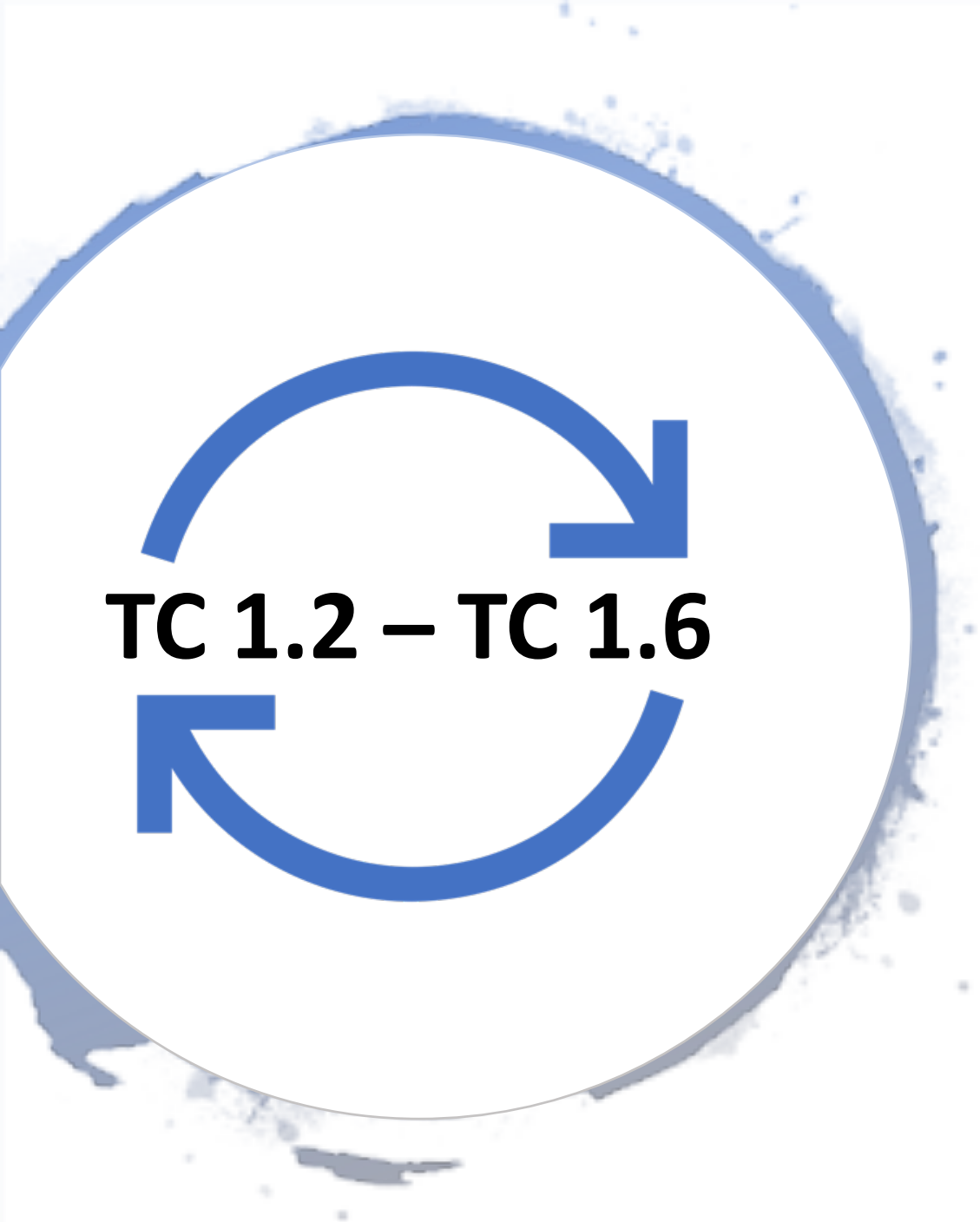
⇒ Rules for equals method

```
public boolean equals (object obj){
    // Null check
    if (obj == null) .....

    // Reference check
    if (obj == this) .....

    // Type check
    if (obj.getClass() == this.getClass()) ...

    // Equality check.
}
```



Perform test for Equality

1: Null Check
2: Ref Check
3: Type Check
4: Value Check for equality

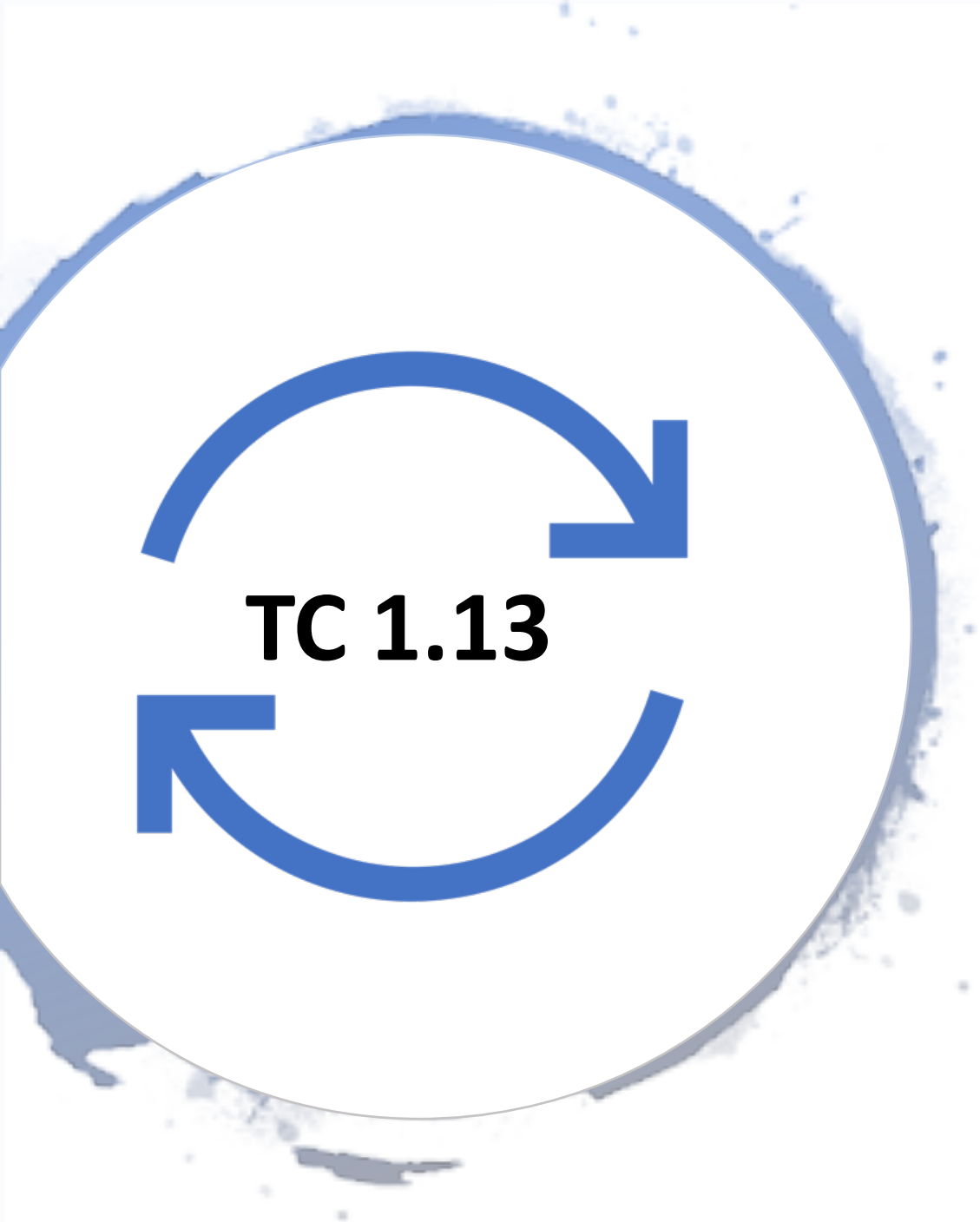
Perform similar test for Inch

1: Null Check
2: Ref Check
3: Type Check
4: Value Check for equality

Refactor 1

Refactor the code as DRY was violated for Feet and Inch with Length and Unit to differentiate

- **Note:**
- Equality check will have Unit Check
- Refactor Test Cases



Comparison Check
Given 0 Feet and 0 Inch
Should Return equal

TC 1.14 – 1.16

Perform test cases for comparing length

1: 1 ft != 1 in
2: 1 in != 1 ft
3: 1 ft = 12 in
4: 12 in = 1 ft

TDD Notes

- Start the first test typically keeping in mind the single variable and then build on it.
- Every Test adds New Constraints
- As Test becomes specific code becomes generic.
- Refactor code to avoid if else logic or duplication
- Refactor Code when ever there is a violation to the Design Principle – DRY or SOLID Principles.

UC 2

As a math student, I wish to compare lengths

3ft = 1yd

TC 1.14 – 1.16

Perform test cases for comparing length

1: 3ft = 1yd
2: 1 ft != 1 yd
3: 1 in != 1 yd
4: 1 yd = 36 in
5: 36 in = 1 yd
6: 1 yd = 3 ft

BridgeLabz

Employability Delivered

Thank You