# What is Unit Testing?

Essentially, a unit test is a method that instantiates a small portion of our application and verifies its behaviour independently from other parts. In procedural programming, a unit referred to as an individual program, while object-oriented programming languages include Base/Superclass, abstract class, Derived/Child class takes place. A typical unit test contains 3 phases: First, it initializes a small piece of an application it wants to test (also known as the system under test, or SUT), then it applies some stimulus to the system under test (usually by calling a method on it), and finally, it observes the resulting behaviour. If the observed behaviour is consistent with the expectations, the unit test passes, otherwise, it fails, indicating that there is a problem somewhere in the system under test. These three-unit test phases are also known as Arrange, Act and Assert, or simply AAA.

[**TestMethod**]

**public** **void** **IsPalindrome\_ForPalindromeString\_ReturnsTrue**()

{

*// In the Arrange phase, we create and set up a system under test.*

*// A system under test could be a method, a single object, or a graph of connected objects.*

*// It is OK to have an empty Arrange phase, for example if we are testing a static method -*

*// in this case SUT already exists in a static form and we don't have to initialize anything explicitly.*

PalindromeDetector detector = **new** PalindromeDetector();

*// The Act phase is where we poke the system under test, usually by invoking a method.*

*// If this method returns something back to us, we want to collect the result to ensure it was correct.*

*// Or, if method doesn't return anything, we want to check whether it produced the expected side effects.*

**bool** isPalindrome = detector.IsPalindrome("kayak");

*// The Assert phase makes our unit test pass or fail.*

*// Here we check that the method's behavior is consistent with expectations.*

Assert.IsTrue(isPalindrome);

}

Unit testing relies on mock objects being created to test sections of code that are not yet part of a complete application. Mock objects fill in for the missing parts of the program.

For example, you might have a function that needs variables or objects that are not created yet. In unit testing, those will be accounted for in the form of mock objects created solely for the purpose of the unit testing done on that section of code.

# What is Integration testing

**It’s** a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated Integration Testing focuses on checking data communication amongst these modules. Integration[Test Case](https://www.guru99.com/test-case.html)differs from other test cases in the sense it**focuses mainly on the interfaces & flow of data/information between the modules.** Here priority is to be given for the **integrating links** rather than the unit functions which are already tested.

# Unit Test vs. Integration Test

Another important thing to consider is the difference between unit testing and integration testing.

The purpose of a unit test in software engineering is to verify the behaviour of a relatively small piece of software, independently from other parts. Unit tests are narrow in scope, and allow us to cover all cases, ensuring that every single part works correctly.

On the other hand, integration tests demonstrate that different parts of a system **work together in the real-life environment.** They validate complex scenarios (we can think of integration tests as a user performing some high-level operation within our system), and usually require external resources, like databases or web servers, to be present.

# References

[Unit Testing and Coding Best Practices for Unit Tests: A Test-Driven Perspective | Toptal](https://www.toptal.com/qa/how-to-write-testable-code-and-why-it-matters)

[What is Integration Testing (Tutorial with Integration Testing Example)](https://www.softwaretestinghelp.com/what-is-integration-testing/)

[What is Integration Testing? | How to perform integration testing? | Edureka](https://www.edureka.co/blog/what-is-integration-testing-a-simple-guide-on-how-to-perform-integration-testing/)

Unit and Integration testing in Flutter

<https://flutter.dev/docs/testing>

<https://flutter.dev/docs/cookbook/testing/unit/introduction>

<https://medium.com/nonstopio/unit-testing-in-flutter-80554f68316>

General web-app testing

<https://www.softwaretestinghelp.com/web-application-testing/>