OBJECTIVE

* Explain the meaning of Unit testing and its difference on comparison with Functional testing
  + Smallest unit to test mocking dependencies

### ****Unit Testing****

**Definition:**  
Unit testing involves testing the **smallest possible piece of code**, typically a **function or method**, in **isolation** from the rest of the application.

### ****Functional Testing****

**Definition:**  
Functional testing verifies that a system or component behaves **according to requirements**, focusing on **what** the system does rather than **how** it does it.

### ****Difference Between Unit Testing and Functional Testing****

| **Feature** | **Unit Testing** | **Functional Testing** |
| --- | --- | --- |
| Scope | Smallest code unit (e.g., function) | Complete functionality/feature |
| Dependencies | Mocked or stubbed | Real (or integrated) components |
| Responsibility | Developer | QA/Testers |
| Type | White-box | Black-box |
| Speed | Very fast | Slower |
| Purpose | Check internal logic correctness | Validate user/business requirements |

* List various types of testing
  + Unit testing, Functional testing, Automated testing, Performance testing

### ****Types of Testing****

**Unit Testing** – Test individual units of code.

**Functional Testing** – Test features as per requirements.

**Automated Testing** – Use tools/scripts to perform tests automatically.

**Performance Testing** – Test system performance under load (e.g., speed, responsiveness).

* Understand the benefit of automated testing

· **Fast Feedback Loop** – Quickly catch bugs after every code change.

· **Reusability** – Test scripts can be reused across builds.

· **Reliability** – Reduces human error.

· **Cost-effective** – Saves time and money in the long run.

· **Continuous Integration Support** – Integrates well with CI/CD pipelines.

* Explain what is loosly coupled & testable design

**Loosely Coupled Design:**  
A system where **components are independent** of each other and interact through **well-defined interfaces**. This means changes in one component don’t heavily impact others.

**Testable Design:**  
A design is testable when:

Components can be tested in isolation (unit testing).

There are **clear inputs and outputs**.

Dependencies can be easily **mocked or injected** (using Dependency Injection).

* Understand the need of [SetUp], [TearDown] & [Ignore] attributes.

**[SetUp]**

**Why it’s needed:**

To initialize objects, variables, or database connections.

To **avoid repeating code** in each test.

[TearDown]

**Why it’s needed:**

To **release resources**, close connections, or clean temporary data.Helps prevent side effects from impacting other tests.

**[Ignore]**

**Why it’s needed:**

Test is not ready or broken and shouldn’t block the test suite.

Useful during development or when tracking known issues.

* Explain the benefit of writing parameterised test cases.

**Reduces Code Duplication**: You don’t need to write multiple test methods for different input values.

**Improves Test Coverage**: Allows you to test a wide range of input combinations using the same logic.

**Easier Maintenance**: If the test logic changes, you only update it in one place instead of multiple test methods.