### ****1. White Box Testing****

**Definition**: Testing the internal workings or logic of the application. The tester needs knowledge of the code, algorithms, and architecture.

#### ****Key Features****:

* Focuses on code structure, logic, and flow.
* Tests internal paths, conditions, and loops.
* Requires programming skills.

#### ****Examples****:

1. **Unit Testing**:  
   Testing a function that calculates the sum of two numbers to ensure it handles edge cases like negative numbers or zero.

python

def add(a, b):

return a + b

* + Test Case: add(2, -3) should return -1.

1. **Code Coverage Testing**:  
   Checking if all conditional branches in a piece of code are executed. For example:

python

if age >= 18:

print("Eligible to vote")

else:

print("Not eligible")

* + Ensuring both the "if" and "else" paths are tested.

### ****2. Black Box Testing****

**Definition**: Testing the application without any knowledge of its internal code or architecture. Focuses on inputs and outputs.

#### ****Key Features****:

* Tests functionality, usability, and performance.
* No need for programming knowledge.
* Treats the system as a "black box."

#### ****Examples****:

1. **Login Feature Testing**:  
   Testing if the login page accepts correct credentials and denies incorrect ones without looking at the code.
   * Input: Username: user123, Password: password@123
   * Expected Output: Login successful.
2. **Boundary Value Analysis**:  
   Testing a field that accepts age between 18 and 60.
   * Test Inputs: 17, 18, 60, 61.
   * Expected Outputs: Invalid for 17 and 61, Valid for 18 and 60.

### ****3. Grey Box Testing****

**Definition**: Combines elements of both White Box and Black Box testing. The tester has partial knowledge of the internal structure but tests from a user perspective.

#### ****Key Features****:

* Balances internal knowledge with user-oriented testing.
* Can test integration points or database interactions.
* Tester uses knowledge of code for better test design.

#### ****Examples****:

1. **Database Interaction Testing**:  
   Testing a user registration form while knowing how data is stored in the database.
   * Input: Fill out a registration form.
   * Verify: Check if the user details are stored correctly in the database table.
2. **Session Management Testing**:  
   Testing a logout feature:
   * Knowledge: Understand that session IDs are invalidated on logout.
   * Test: Log out, then attempt to reuse the session ID to access the application.

### ****Comparison****

| **Aspect** | **White Box Testing** | **Black Box Testing** | **Grey Box Testing** |
| --- | --- | --- | --- |
| **Knowledge Required** | Full knowledge of code | No knowledge of code | Partial knowledge of code |
| **Focus** | Code structure, logic | Functionality and outputs | Internal and external behavior |
| **Tools** | Code analyzers, debuggers | Test management tools | Both types of tools |
| **Example** | Code coverage, unit testing | UI testing, functional testing | Database and API testing |