✅ 1. Test and Lifecycle Annotations

Annotation	Description
`@Test`	Marks a method as a test case.
`@RepeatedTest`	Runs the test method multiple times.
`@ParameterizedTest`	Runs a test with multiple parameters.
`@BeforeEach`	Executes before each `@Test`.
`@AfterEach`	Executes after each `@Test`.
`@BeforeAll`	Executes once before all tests. Must be static (unless using `@TestInstance`).
`@AfterAll`	Executes once after all tests. Must be static (unless using `@TestInstance`).
`@Disabled`	Disables a test class or method. Skips it during execution.

✅ 2. Assertions and Assumptions (Used via methods, not annotations)

Not annotations but often confused with them:

- `Assertions.assertEquals()`, `assertTrue()`, etc.
- `Assumptions.assumeTrue()`, `assumeFalse()`, etc.

✅ 3. Parameterized Test Source Annotations

Used with `@ParameterizedTest`:

Annotation	Description
`@ValueSource`	Supplies primitive values or Strings.
`@EnumSource`	Supplies enum constants.

`@CsvSource`	Supplies CSV-formatted string values.
`@CsvFileSource`	Supplies CSV data from a file.
`@MethodSource`	Calls a static method that returns a stream or iterable of arguments.
`@ArgumentsSource`	Supplies arguments using a custom provider class.
`@NullSource`	Supplies a `null` value.
`@EmptySource`	Supplies empty values (e.g. "", empty list).
`@NullAndEmptySource`	Supplies both null and empty.

✅ 4. Tagging and Filtering

Annotation	Description
`@Tag`	Used to tag tests for filtering (e.g., `@Tag("integration")`).
`@DisplayName`	Sets a custom name for test methods/classes. Appears in test reports.

✅ 5. Test Organization

Annotation	Description
`@Nested`	Allows nested test classes with their own lifecycle.
`@TestInstance`	Controls lifecycle of test class (per-method or per-class).
`@TestMethodOrder`	Specifies the order of test method execution.
`@Order`	Used with `@TestMethodOrder` to define order of individual tests.

✅ 6. Extensions and Advanced Features

Annotation	Description
`@ExtendWith`	Registers a JUnit 5 extension (e.g., Mockito, Spring).
`@Timeout`	Fails a test if it exceeds given time limit.
`@TempDir`	Injects a temporary directory for test use.

`@RegisterExtension`	Declares a programmatic extension field (non-static).

🔁 Summary by Category

Category	Examples
Test Lifecycle	`@BeforeEach`, `@AfterEach`, `@BeforeAll`, `@AfterAll`, `@Test`, `@Disabled`
Parameterized	`@ParameterizedTest`, `@ValueSource`, `@CsvSource`, etc.
Organization	`@Nested`, `@DisplayName`, `@Tag`, `@Order`
Extensions	`@ExtendWith`, `@RegisterExtension`, `@TempDir`
Timing/Timeouts	`@Timeout`

Would you like a downloadable cheat sheet or a working code example using all of these annotations?