COHESION	**COUPLING**
Cohesion is the concept of intra-module.	Coupling is the concept of inter-module.
Cohesion represents the relationship within a module.	Coupling represents the relationships between modules.
Increasing cohesion is good for software.	Increasing coupling is avoided for software.
Cohesion represents the functional strength of modules.	Coupling represents the independence among modules.
Highly cohesive gives the best software.	Whereas loosely coupling gives the best software.
In cohesion, the module focuses on a single thing.	In coupling, modules are connected to the other modules.
Cohesion is created between the same module.	Coupling is created between two different modules.
When it comes to cohesion, the module concentrates only on a single thing.	When it comes to coupling, the modules are associated with the other modules.

There are Six types of Cohesion (F SC PT LC) 1. Functional Cohesion. 2. Sequential Cohesion. 3. Communication Cohesion 4. Procedural Cohesion. 5. Temporal Cohesion. 6. Co-incidentental Cohesion.	There are Six types of Coupling (DSC ECC) 1. Data Coupling. 2. Stamp Coupling 3. External Coupling 4. Control Coupling 5. Common Coupling. 6. Content Coupling.

Black Box Testing	**White Box Testing**
It is a way of software testing in which the **internal structure or the program or the code** is hidden and nothing is known about it.	It is a way of testing the software in which **the tester has knowledge about the internal structure or the code or the program of the software**.
Implementation of code is not needed for black box testing.	**Code implementation is necessary** for white box testing.
It is mostly done by **software testers**.	It is mostly done by **software developers.**

No knowledge of implementation is needed.	**Knowledge of implementation** is required.
It can be referred to as **outer or external software testing**.	It is **the inner or the internal software testing.**
It is **a functional test** of the software.	It is **a structural test** of the software.
This testing can be initiated based on the **requirement specifications document.**	This type of testing of software is started **after a detail design document**.
No knowledge of programming is required.	It is **mandatory** to have knowledge of programming.
It is the **behavior testing** of the software.	It is the **logic testing** of the software.
It is applicable to **the higher levels** of testing of software.	It is generally applicable to **the lower levels** of software testing.
It is also called **closed testing.**	It is also called as **clear box**

	testing.
It is **least time consuming**.	It is **most time consuming.**
It is **not suitable or preferred for algorithm t**esting.	It is **suitable** for algorithm testing.
Can be done by trial and error ways and methods.	Data domains along with inner or internal boundaries can be better tested.
Example: Search something on Google by using keywords	**Example:** By input to check and verify loops
Black-box test design techniques- • Decision table testing • All-pairs testing • Equivalence partitioning • Error guessing	**White-box test design techniques-** • Control flow testing • Data flow testing • Branch testing
Types of Black Box Testing: • Functional Testing • Non-functional testing • Regression Testing	**Types of White Box Testing:** • Path Testing • Loop Testing • Condition testing
It **is less exhaustive** as compared to	It is comparatively **more**

white box testing.	**exhaustive** than black box testing.

Verification	Validation
We check whether we are developing the right product or not.	We check whether the developed product is right.
Verification is also known as **static testing**.	Validation is also known as **dynamic testing**.
Verification is a static practice of verifying documents, design, code and program.	***Validation*** is a dynamic mechanism of validating and testing the actual product.
Verification includes different methods like Inspections, Reviews, and Walkthroughs.	Validation includes testing like functional testing, system testing, integration, and User acceptance testing.
It is a process of checking the work-products (not the final product) of a development cycle to decide whether the product meets the specified requirements.	It is a process of checking the software during or at the end of the development cycle to decide whether the software follow the specified business requirements.
Quality assurance comes under verification testing.	**Quality control** comes under validation testing.

In Verification testing - does not involve executing the code.	In validation testing - always involves executing the code.
In verification testing, we can find the bugs early in the development phase of the product.	In the validation testing, we can find those bugs, which are not caught in the verification process.
Verification testing is executed by the Quality assurance team to make sure that the product is developed according to customers' requirements.	Validation testing is executed by the testing team to test the application.
Verification is done before the validation testing.	After verification testing, validation testing takes place.
In this type of testing, we can verify that the inputs follow the outputs or not.	In this type of testing, we can validate that the user accepts the product or not.
It is human based checking of documents and files.	It is computer based execution of program
Verification uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.	***Validation*** uses methods like black box (*functional*) testing, gray box testing, and white box (*structural*) testing etc
Verification is to check whether the software conforms to specifications.	***Validation*** is to check whether software meets the customer expectations and requirements.
Verification can catch errors that validation cannot catch. It is low level exercise.	***Validation*** can catch errors that verification cannot catch. It is High Level Exercise.

Index	Black Box Testing	White Box Testing	Grey Box Testing
1	Knowledge of internal working structure (Code) is **not required** for this type of testing. **Only GUI** (Graphical User Interface) is **required** for test cases.	Knowledge of internal working structure (Coding of software) is **necessarily required** for this type of testing.	**Partially Knowledge** of the internal working structure is **required**.
2	Black Box Testing is also known as **functional testing, data-driven testing, and closed box testing**.	White Box Testing is also known as **structural testing, clear box testing, code-based testing, and transparent testing**.	Grey Box Testing is also known as **translucent testing** as the **tester has limited knowledge of coding.**
3	The approach towards testing includes **trial techniques and error guessing method** because **tester does not need knowledge of internal coding** of the software.	White Box Testing is preceded by **verifying the system boundaries and data domains inherent** in the software as there **is no lack of internal coding knowledge.**	If the tester has knowledge of coding, then it is preceded by **validating data domains and internal system boundaries** of the software.
4	The testing space of tables for inputs (inputs to be used for creating test cases) is **pretty huge and largest among all**	The testing space of tables for inputs (inputs to be used for creating test cases) is **less as compared** to Black Box testing.	The testing space of tables for inputs (inputs to be used for creating test cases) **is smaller than Black Box**

	testing spaces.		**and White Box testing.**
5	It is very **difficult to discover hidden errors** of the software because **errors can be due to internal working** which is **unknown for Black Box testing.**	It is **simple to discover hidden errors** because it can be due to **internal working which is deeply explored** in White Box testing.	**Difficult to discover the hidden error**. Might be found in **user level testing**.
6	It **is not** considered for **algorithm testing**.	It is **well suitable and recommended** for algorithm testing.	It is **not considered** for algorithm testing.
7	**Time consumption** in Black Box testing **depends upon the availability** of the functional specifications.	White Box testing takes **a long time to design test cases** due to lengthy code.	Test cases designing can be done in **a short time period.**
8	**Tester, developer and the end user** can be the part of testing.	**Only tester and developer** can be a part of testing; the end user can not involve.	**Tester, developer and the end user** can be the part of testing.
9	It is **the least time-consuming process** among all the testing processes.	The entire testing process is **the most time consuming** among all the testing processes.	**Less time consuming** than White Box testing.
10	Resilience and security against viral	Resilience and security against viral attacks	Resilience and security against

	attacks are **covered under Black Box testing**.	are **not covered under White Box testing**.	viral attacks are **not covered under Grey Box testing.**
11	The base of this testing is **external expectations internal behavior is unknown**.	The base of this testing is coding which is **responsible for internal working.**	Testing based on **high-level database diagrams and dataflow diagrams.**
12	It is **less exhaustive** than White Box and Grey Box testing methods.	It is **most exhaustive** between Black Box and Grey Box testing methods.	**Partly exhaustive**; depends upon the type of test cases are coding based or GUI based.

Difference between Static testing and Dynamic Testing

Static testing	Dynamic testing
In static testing, we will check the code or the application without executing the code.	In dynamic testing, we will check the code/application by executing the code.
Static testing includes activities like code Review, Walkthrough, etc.	Dynamic testing includes activities like functional and non-functional testing such as UT (usability testing), IT (integration testing), ST (System testing) & UAT (user acceptance testing).
Static testing is a **Verification**	Dynamic testing is a **Validation**

Process.	Process.
Static testing is used to prevent defects.	Dynamic testing is used to find and fix the defects.
Static testing is a more cost-effective process.	Dynamic testing is a less cost-effective process.
This type of testing can be performed before the compilation of code.	Dynamic testing can be done only after the executables are prepared.
Under static testing, we can perform the statement coverage testing and structural testing.	Equivalence Partitioning and Boundary Value Analysis technique are performed under dynamic testing.
It involves the checklist and process which has been followed by the test engineer.	This type of testing required the test case for the execution of the code.

Differences between the Alpha testing and Beta testing are:

Sl. No.	Alpha Testing	Beta Testing
1.	Alpha testing performed by a team of highly skilled testers who are usually the internal employee of the organization.	Beta testing performed by clients or end-users in a real-time environment, who is not an employee of the organization.
2.	Alpha testing performed at the developer's site; it always needs a testing environment or lab environment.	Beta testing doesn't need any lab environment or the testing environment; it is performed at a client's location or end-user of the

		product.
3.	Reliability or security testing not performed in-depth in alpha testing.	Reliability, security, and robustness checked during beta testing.
4.	Alpha testing involves both white box and black-box techniques.	Beta testing uses only black-box testing.
5.	Long execution cycles maybe require for alpha testing.	Only a few weeks are required for the execution of beta testing.
6.	Critical issues or fixes can be identified by developers immediately in alpha testing.	Most of the issues or feedback is collecting from the beta testing will be implemented for the future versions of the product.
7.	Alpha testing performed before the launch of the product into the market.	At the time of software product marketing.
8.	Alpha testing focuses on the product's quality before going to beta testing.	Beta testing concentrates on the quality of the product, but gathers users input on the product and ensures that the product is ready for real-time users.
9.	Alpha testing performed nearly the end of the software development.	Beta testing is a final test before shipping a product to the customers.
10.	Alpha testing is conducting in the presence of developers and the absence of end-users.	Beta testing reversed of alpha testing.