1. Difference between retesting and regression testing?

Regression Testing	Retesting
Regression testing is a type of software testing that intends to ensure that changes like defect fixes or enhancements to the module or application have not affecting unchanged part.	Retesting is done to make sure that the tests cases which failed in last execution are passing after the defects against those failures are fixed.
Regression testing is not carried out on specific defect fixes. It is planned as specific area or full regression testing.	Retesting is carried out based on the defect fixes.
In Regression testing, you can include the test cases which passed earlier. We can say that check the functionality which was working earlier.	In Retesting, you can include the test cases which failed earlier. We can say that check the functionality which was failed in earlier build.
Regression test cases we use are derived from the functional specification, the user manuals, user tutorials, and defect reports in relation to corrected problems.	Test cases for Retesting cannot be prepared before start testing. In Retesting only re-execute the test cases failed in the prior execution.

2. Which of the one are part of functional testing -
a. UAT, Integration, Regression
b. Maintenance, Volume, Performance

c. Sanity, Localization, unit
Answer: (c) Sanity,Localization,unit

3. System testing is done before integration testing – True/False
Ans : False

4. Confirmation testing is same as regression testing – True/False
Ans : False

5. Difference between static and dynamic testing.

Static Testing	Dynamic Testing
1. Static Testing is white box testing which is done at early stage if development life cycle. It is more cost effective than dynamic testing	1. Dynamic Testing on the other hand is done at the later stage of development lifecycle.
2. Static testing has more statement coverage than dynamic testing in shorter time	2. Dynamic Testing has less statement stage because it is covers limited area of code
3. It is done before code deployment	3. It is done after code deployment
4. It is performed in Verification Stage	4. It is done in Validation Stage

5. This type of testing is done without the execution of code.	5. This type of execution is done with the execution of code.

6. Difference between SDLC & STLC

SDLC (Software Development Life cycle)	**STLC (Software Test Life Cycle)**
SDLC is Software Development LifeCycle, it is a systematic approach to develop a software.	The process of testing a software in a well planned and systematic way is known as software testing life cycle(STLC).
Requirements gathering	Requirements Analysis is done is this phase, software requirements are reviewed by test team.
Design	Test Planning, Test analysis and Test design is done in this phase. Test team reviews design documents and prepares the test plan.
Coding or development	Test construction and verification is done in this phase, testers write test cases and finalizes test plan.
Testing	Test Execution and bug reporting, manual testing, automation testing is done, defects found are reported. Re-testing and regression testing is also done in this phase.
Deployment	Final testing and implementation is done is this phase andfinal test report is prepared.

Maintenance	Maintenance testing is done in this phase.

7. List 3 advantage/disadvantage of Waterfall model
Ans: Advantage-

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.

Disadvantage:

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.

8. What do you understand by the term Functional testing?

FUNCTIONAL TESTING is a type of software testing whereby the system is tested against the functional requirements/specifications.

Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how

processing occurs, but rather, with the results of processing. It simulates actual system usage but does not make any system structure assumptions.

9. Is it true that we can do system testing at any stage?
Ans: No

10. List down difference between validation and verification processes

Verification	***Validation***
1. **Verification** is a static practice of verifying documents, design, code and program.	1. **Validation** is a dynamic mechanism of validating and testing the actual product.
2. It does not involve executing the code.	2. It always involves executing the code.
3. It is human based checking of documents and files.	*3.* It is computer based execution of program.
4. **Verification** uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.	*4.* **Validation** uses methods like black box (functional) testing, gray box testing, and white box (structural) testing etc.
5. **Verification** is to check whether the software conforms to specifications.	5. **Validation** is to check whether software meets the customer expectations and requirements.

11. What are stubs and drivers

Stubs :Stubs are used to test modules and are created by the team of testers during the process of Top-Down Integration Testing. With the assistance of these test stubs testers are capable of stimulating the behaviour of the lower level modules that are not yet integrated with the software. Moreover, it helps stimulates the activity of the missing components.

Drivers: Drivers, like stubs, are used by software testers to fulfil the requirements of missing or incomplete components and modules. These are

usually complex than stubs and are developed during Bottom-Up approach of Integration Testing. Drivers can be utilized to test the lower levels of the code, when the upper level of codes or modules are not developed. Drivers act as pseudo codes that are mainly used when the stub modules are ready, but the primary modules are not ready.

12. Final product or the software cannot be released without passing through the STLC process - True/False
Ans: True

13. Choose the correct one
a. Testing should start after development
b. Testing should start as early as possible in software cycle
c. Exhaustive testing is proof of delivering correct product
d. Testing is context independent
Ans: (b) Testing should start as early as possible in software cycle

14. Maintenance testing deals with retesting to show that the rest of the system has not been affected by the maintenance work – True/False
Ans : True

15. Maintenance testing deals with regression testing to show that the rest of the system has not been affected by the maintenance work – True/False
Ans: True

16. Unit testing is performed by developers - True/False
Ans: True

17. In V model testing activities are carried out in parallel with development activities - True/False
Ans : True

18. Static testing include –
a. Inspection, regression, unit testing
b. Retesting, system, End user
c. Review, inspection, Walkthrough
d. Review, inspection, acceptance

Ans : C

19. Acceptance testing is most often focused on a validation type of testing - True/False

Ans :

20. Integration testing focuses on testing different modules all together - True/False
Ans: True