# JUNIT

→ What is the latest version of Junit ?

Junit 5

→ What is Junit ?

Junit is a open source framework to write and run repeatable tests.

→ Junit 3 and Junit 4

Junit 4 and above provides annotation to define test methods & life cycle methods of Unit testcase.

→ What is Unit Testing ? What is Unit TestCase ?

Testing done by developer is called Unit Testing and written test code is called Unit Testcase.

→ What is regression Testing ? Give a Scenario ?

When we test newly developed method along with old methods of a class then it is called Regression Testing.

Scenario :-

When we test newly created method registerUser() along with we test old method add().

→ Junit Annotations :-

@Test , @Ignore , @BeforeClass , @AfterClass

@Before , @After

→ Fails Vs AssertMethods

Assert methods are used to record failed tests whereas fail() method is used to fail the test on a custom condition.

→ Which assert methods are available to test the different test Conditions?

* assertTrue() :- Check True Condition
* assertFalse() :- Check false condition
* assertEqual() :- check equality of two Objects
* assertNotEqual() :- Check both objects are not Equal
* assertNull() :- Check if Object is Null
* assertNotNull() :- Check if object is NotNull

→ Who develop unit test cases?
Developer

→ who run the testcase?
Developer

→ What is Test Suite? Who runs test suite & when?

* Test Suite is the set of Unit test cases. we can run multiple testcases with the help of test suite.
Test Suite can be defined by two annotation @Run with and @Suite annotations.
* Deployment lead run test suit after deploying the build.

→ Which Annotation is used to test the method?
@Test

→ TestCase LifeCycle Methods :-

* @BeforeClass :- Executed only once when testcase class execution is started.
* @AfterClass :- Executed only once when testcase class execution is finished.
* @Before :- executed before each test method
* @After :- executed after each Test method.

Unit Testcase :- Testcase is a test program to test a UseCase functionality.

Test Runner :- A test runner is Software that runs tests and reports result.

# Log4J

Dependency
log4j - 1.2.17

→ What is the latest version of Log4J?
Log4J 2 Log4j 2.13 (Java 8 or higher)

→ What is Log4J?
Log4J stands for logging messages for java. Loggng Log4J is a open source framework that facilitates a developer to log messages during application development

→ Why do you use logging in your application?
To track the health & flow of the application.

→ What is Appender?
Appender is an object that sends log messages to their final destination.

→ How many types of Appenders are there?
* File appender
* Console Appender
* Rolling File Appender
* SMTP Appender
* Socket Appender
* Jdbc Appender
* Telnet Appender
* Null Appender

→ Different levels of Messages:-
Debug < Info < Warn < Error < fatal

* Debug has lowest priority.
* Fatal has Highest priority.
* Message levels in different Environment:-
  - > Development Environment — DEBUG
  - > QA Environment — Info
  - > Production Environment — WARN

Which layout to have used → Pattern Layout (conversion pattern)

→ Logging Destination :
Console, File, Network and Database.

→ Can we send messages to multiple distination at same time
Yes

→ What is Rolling file Appender ?
Rolling file Appender can be configured to create a new log file every day or create a new file when given file size is finished.

→ Configuration file :-
log4J.properties & log4j.xml

→ where do you keep log4j.properties or log4j.xml file in your application class path
In Root package of our application

→ How many components in Log4J ?
There are 3 components in Log4J :-

1. Logger :- It is responsible for capturing logging information
2. Appender :- It is responsible for publishing logging information to various destinations
3. Layout :- It is responsible for formatting logging information in different style.

→ How to disable log4j Message ?
Log4j.rootlogger = OFF

→ Why we use log4J rootlogger property ?
Log4J.rootlogger property is used to set the target appenders and message level
Log4J.rootlogger = DEBUG, File, Stdout

→ How we make object of Logger class ?
Logger log = logger.getlogger (Current Class)

Rolling file Appender [illegible] log files [illegible] size [illegible] date both depending on [illegible]

→ Which appenders will we used in our application?
Rolling File Appender and Console Appender

→ How many types of layout are Available?
- DateLayout
- SimpleLayout
- HTML layout
- XML layout
- patternlayout

→ How you can Configure multiple Appenders in properties?
- First we set target Appenders in Rootlogger then we set Org.apache.log4j.RollingFileAppender & Org.apache.log4j.ConsoleAppender.

→ Which design pattern is followed by Log4J?
factory design pattern is followed by
logger log = logger.getLogger(this.getClass());