|  |
| --- |
| **Q. TEST CASE TO BE AUTOMATED: Tests are used repeatedly 🡺Tests involves a lot of data entry 🡺 The test execution rate is high 🡺 Tests with several combinations 🡺 Higher manual execution time 🡺Tests clearly pass or fail 🡺Tests deliver an exact result 🡺Tests use consistent UI and regular controls 🡺Tests are only to do what they're told — not check anything else.**  **Q. NOT AUTOMATED: Subjective testcases like test cases that are not testing a clear function 🡺New functionalities 🡺 Complex Functionalities 🡺 Installation & set up Testing.**  **Q. WHY AUTOMATION: 🡺Faster Execution 🡺 Create one time and execute multiple times 🡺 Easy for Compatibility Testing. It enables parallel execution in the combination of different OS and browser environments. 🡺 It allows us to integrate with CROSS BROWSER TESTING TOOLS, JENKINS, GITHUBS 🡺 Test scripts can be run unattended 🡺 Maximum Coverage, helps us to increase the test coverage.**  **Q. WHAT TYPE TESTS HAVE YOU AUTOMATED: Main focus is to automate test cases to REG, SMOKE, SANITY test. Sometimes based on the project and the test time estimation, we do focus on End-to-End testing.** |
| **JDK is a collection of programming tools, JRE, JVM. For create, compile & run your java program need to installed JDK. JRE is a part of JDK. JRE contains JVM, set of libraries plus other files. When you have JRE installed on your system, you can run a java program however you won’t be able to compile it. Writing of the program is of course done by java programmer like you and me it is called source code (.java file) and javac compiler convert source code to bytecode(.class file). JVM executes the bytecode generated by compiler.** |
| **🡺Throwable class inherits by exception & error class.**  **🡺Try Block: Java try block is used to enclose the code that might throw an exception.**  **🡺Catch Block: Java catch block is used to handle the exception by declaring the type of exception within the parameter.**  **Finally Block: You should place those statements in this block that must be executed always. Basically, file close, database connection etc. can be closed in finally block. Finally block will not be executed if program exits (either by calling System.exit).**  **🡺Throw: Java can't create each and every exception. That time I can create my own exception by throw keyword. It created by "add throws declaration" or "surround with try-catch block".**  **🡺Throws: Used to handle the exception but we don't report anything, there is no option for report anything. The "throws" is used to throw an exception for a method.** |
| **🡺CONSTRUCTOR cannot be STATIC, FINAL, INHERITED, ABSTRACT. Return Instance of a Class.**  **🡺METHODOVERLOADING can be declare as FINAL, SYNCHRONIZED, cannot overload by STATIC**  **🡺Static, Final, Private, non-static as Static cannot be OVERRIDEN, Synchronized in sub-class.**  **🡺Data hiding, code reuse, sub-class INHERIT STATIC members, tightly bound, achieved by Method Overloading.**  **🡺hiding data implementation, has Constructor, Static, Final method. Static method can’t be ABSTRACT.**  Abstraction solves the problem at design level while encapsulation solves the problem at implementation level.  🡺 INTERFACE is a collection of public, static, final (variables) & abstract method. Can’t be FINAL. **break;}**  **🡺 SWITCH statement executes one statement from multiple conditions. char ch = 'O’; switch(ch) {case 'a': sout("Vowel");** |
| Collection Interface extends Iterable Interface. Three interface LIST, QUEUE, SET extends Collection Interface. Array List, Linked List, Vector class implement LIST interface. Stack extends Vector. DEQUE extends QUEUE and PRIORITYQUEUE implements QUEUE. HASH SET, LINKED HASH SET implements SET. SORTED SET extends SET. TREESET interface implements SORTED SET. |

|  |
| --- |
|  |
| Q. What is Jenkins:  Jenkins is a CI/CD tools. In our project we use Jenkins for continuous development, continuous integration and continuous deployment. So continuous development in the sense like whatever code developers are writing and storing those code in github. In Jenkins we have configured it. So, Jenkins will download the code from github. It will check for compilation issue, logic is correct or not, build is brokage or not. In automation testing we are using it for framework build testing where continuous integration process is happen, all the test engineer who work in the frame work, they push their code in repo, Jenkins will download the framework build, when the developers are giving the build, so it deploying into the testing environment and it will get the framework build from the GitHub, then it will execute the framework build in the testing environment and after that if we are getting the build successful message and no issues is shown then it will deployed into the UAT environment. In Jenkins we have different type of execution like on scheduling and also support parameter. |