**GENERAL TOPIC TO KEEP CHECKING**.

SDET – Test Automation Engineer (Software Quality Engineer)

Client Server Architecture

Types of clients and servers.

Computer Science (Hardware and Software systems), Software Engineering and Development (SDLC).

Review, Re-Study everything – Lifelong Learning

Understand WHAT the theoretical knowledge, HOW to Implement the practical hands-on exercises, Apply to a specific problem.

UNDERSTAND **Front-end – Middleware – Backend technologies** used for developing software application, **WHAT** software testing tools are built for testing these applications and **HOW** to use them to solve testing needs.

Familiarize and play around with **COMPUTER PROGRAMS**. E.g Specflow, TestNg, Nunit, Java, CSharp, RestAssured, Karate API, Selenium and many more.

**Computer Files and Data –** file types are defined by the file extensions and **WHAT** kind of data can be stored in file and **HOW** to construct or create the data syntax vocabulary rules in the file and how to access data in the file**.**

**Computer Programs allows a user to complete a specific task.**

* SDET – Test Developer, Test Automation Engineer
* Test Automation RoadMap
* Software Engineering, Computer Science and Software development.
* **Front-end, Middleware and Backend software.**
* Study order developers code, study how they logically solve a programming problem.
* CICD
* ROI – Return on investment.
* UML diagram and Entity relationship modelling.
* Model an object.
* All Objects communicate to each other using a common interface.
* **SOLID** principles
* **Logical Reasoning and Analysis.**
* **Computer hardware and Computer software or computer program.**
* **Computer Software (Program) System software, Operating Systems (Linux, windows, Android, IOS), Network Operating Systems.**
* **Application software (Command-Line, Software libraries – API, COM), GUI web-based, desktop, mobile, virtual software application.**
* **Mathematics and Physics.**
* **Understand singular and plural English.**
* Characters for data – data forms information – information forms knowledge – knowledge forms wisdom.
* Forms of learning – read text, videos, explain and apply ton solve a specific problem.
* Base64 mechanism.
* **Program** – input process output storage.
* **Object** – state and behaviour.
* **It can be called Program, Element, Object, Resource, Service, Entity, Record.**
* **It can be called property, field, attribute, input, key.**
* **Data – set of value (singular or plural)**
* **Related objects as attributes or properties that REPRESENT an OBJECT.**
* **Operations, Mechanism, Process, Function (statements), Method.**
* Defects and bug.
* Debugging.
* Download COMPLETE REFERNCE BOOKS.
* **Study with textbooks, Videos, explain and apply to a specific problem**.
* Nth tier layer (phase) architecture.
* Presentation – Business Logic layer – Implementation Layer.
* Design Patterns, Data Structures.
* **Problem Solving Skills – Algorithm – Implementation - maintenance.**
* **Analytical Skills and Soft skills.**
* Lambda.
* Follow instructions, Rule, Constraints.
* GIT and GITHUB, BITBUCKET.
* Java, JavaScript, C#, Python.
* Clues, ideas, guess the keywords (define it and look for object relationship)
* Object REPRESENTATION.
* Client Server architecture.
* Required relation properties that REPRESENT an object, e.g. Student.
* PUSH and always PULL mindset.
* Annotations and Attributes (A programmer gives a set of logical instructions to the **compiler to process a program**).
* Unit Testing Frameworks.
* Serialization and Deserialization.
* TestNG, Specflow, Cucumber, Nunit, Pytest.
* Features or Functionality on a program, API, Software application.
* IDE – **Compiler.**
* UNDERSTAND **Compile time, Build-time, Runtime Objects or Types.**
* Explicit and Implicit meaning.
* **WHAT and HOW.**
* **There are many approaches or ways to apply a solution to a problem.**

**Software Testing**

* Specifications, Requirements (needs)
* Defect
* Types and Kinds of testing.
* Testing books.

**Object Oriented Programming Languages**

* Java – access (getters) and mutators (setters).
* C#
* Python
* JavaScript