Automation Testing :

1. Manual Testing
   1. Different type of testing
   2. TestCase / Test Plan
   3. SDLC / STLC
   4. Defect LC
2. JAVA
   1. OOPS Concept
   2. Why Java is popular
   3. Installation
      1. Download
      2. Set the environments ( Windows and MAC )
      3. IDE (**Eclipse**, Netbeans, **intelllij**.. )
      4. Create a project
   4. Variables
      1. Local variables
      2. Global variables
      3. Constants
   5. Methods
      1. Methods without arguments
      2. Methods with arguments
      3. Methods without return type
      4. Methods with return type
   6. Access specifies
      1. Private
      2. Default
      3. Protected
      4. Public
   7. Access modifiers
      1. Static
      2. Final
      3. Abstract
      4. Synchronized
   8. Classes
      1. Abstract class
      2. Final class
      3. *Inner classes*
   9. Interfaces
   10. Conditional Statements
       1. If
       2. If ,else
       3. If ,elseif ,else
       4. Switch
   11. Looping Statements
       1. For
       2. While
       3. Do-while
       4. Foreach
   12. Operators (arithmetic, logical operators, increment and decrement Operators, conditional operators )
   13. String
       1. String functions
       2. Programs
       3. StringBuffer and StringBuilder
   14. Threads
       1. Why ? Advantages ????
       2. Thread class
       3. Runnable interface
   15. Exception Handling
       1. Try
       2. Catch
       3. Throws
       4. Throw
       5. Finally
       6. How to write UDE
   16. File Handling
       1. How to read and write files
          1. Txt
          2. Properties
          3. Excel
   17. Collections
       1. List
       2. Set
       3. Map
   18. Generics
3. Automation with Selenium:
   1. Why Automation is required
   2. What are the different tools available
   3. Why Selenium is Popular
   4. Different versions of selenium
      1. Selenium IDE -> version-1
      2. Selenium ~~RC -> Selenium 1.0~~
      3. Selenium Web Driver -> Selenium 2.0 / Selenium 3.0 / Selenium 4.0- alpha
      4. Selenium GRID
   5. Framework
      1. Function driven Automation Framework
      2. Keyword driven Automation Framework
      3. Data driven Automation Framework
      4. Hybrid driven Automation Framework
      5. **TestNG**
      6. **POM**
4. **Maven/ Gradle (Build Automation Tool )**
5. **GIT (Version Control Tool)**
6. **Jenkins ( CI and CD )**
7. **BDD (Cucumber)**

GIT

Version Control System



Advantages :

1. We can allow multiple developers to work simultaneously
2. Any deleted files you can get it back
3. Remove any newly added code from the project - can be done easily

Types:

1. LVC – Local Version controlling
2. Centralized version control tool - SVN, Perforce, TFS



1. Distributed Version Control -> GIT, Big bucket,..





Installation:

<https://git-scm.com/download/win>

Configuration:

git config --global user.name "Aravinda HB"

git config --global user.email [aru03.info@gmail.com](mailto:aru03.info@gmail.com)

Creating local repository and making changes

1. go to any folder which you want to make as a repository and execute
   1. git init
2. Add files in that and commit the changes to local repository
   1. git add <<filename>>
   2. git commit –m “<<any valid message>>”
3. Create a Remote Repository (gitlab, github,bigbucket……)

git remote add origin git@github.com:AravindaHB/ITAutomation\_June\_Batch.git

git push -u origin master

java

Installation

- JDK

- JRE

Setting Environment Variables –

- JAVA\_HOME

- PATH

Data types in JAVA :



Variables –

- Local Variables :

- any variable which we declare inside a method is called local variable.

- scope of the variable is only inside the method outside the method we cannot access local variables.

- We cannot specify any access specifies or access modifiers to local variables

-Global Variables :

1. Instance Variable

2. Static Variable

3. Constants

Methods :

/\*

\* Methods describes the ***Behavior*** of an Object

\* reusable entities - you write once and use it many times

\* /

Syntax :

[AS] [AM] return\_type name\_of\_method([arguments list])

{

}

return\_type - can be void OR any data\_type

if the return\_type is other than void - last statement in your method should be return statement

**TYPES:**

1. Methods without return type and with out arguments

2. Methods without return type and with arguments

3. Methods with return type and with arguments

**NOTE**

non-static means instance (Object)

**Access specifies :**

