**Level 1: Codeathon TOC (e.g.) Python, Web testing……**

Level 2: Individual Syllabus for each Codeathon TOC

Level 3: Course Content (i.e.) Actual PPT/ Examples…

-------------------------------------------------------------------------------------------------

**Testoper Codeathon (Become the Expert)**

The Testoper Codeathon is a deep dive coding program delivered by the collaborative engagement of community experts for the community members who are dedicated and inspired to play the role of an expert in future innovation. The intended program content is available at <https://github.com/testoper/Codeathon>

This program is free; however, you register and pay upfront to cover the program space, snacks, and dinner expenses for the whole year. To enjoy the benefits, reserve your spot today, limited seats available.

Start Date:**Mar 5, 2020, from 6 to 8 PM**

Recurrence:**Bi-Weekly every Thursday**

No of Sessions: **22 in 2020**

Capacity:**30 Testopers**

**Location: TBD**

Space and Food Cost:**300 per person/per year**

**Who should attend?**

* Architects, designers, and developers to learn test automation and operations skills in the world of agile to become the expert
* Testers and operations experts for developing skills to advance career in the automation and intelligent automation space
* The individuals who are hungry to stand out of the crowd and passionate about innovation in everything they do
* The individuals who are determined to make a difference at work as an Intrapreneur delivering continuous improvement, and innovation
* And, for individuals to connect, develop, launch, and advance their Entrepreneur dreams into reality

**Prerequisites:**

Prerequisite is good to know before attending Testoper Codeathon. Even if you do not know, No WORRIES! You will learn it together with the community.

1. **Programming Language** – Python, Java, JavaScript
2. **Scripting Language** – Node JS
3. **Modelling Language** - YAML
4. **Database** – MongoDB, CouchDB, MongoDB, PostgreSQL, MySQL
5. **DevOps (version control) –** GitHub

**General Track:**

This track will introduce to the basics of automation and relevant disruptive technologies. And, it will set the excellent foundation for acquiring the breadth of knowledge, which is vital to spark the innovation within, and in what we do in subsequent Codeathon sessions.

1. **Introduction to Automation**
2. **Disruptive Technology Basics** - Cloud, NFV/SDN, MEC, 5G, IoT, Industry 4.0, Blockchain, AI, AR and VR
3. **Introduction to Automation for Testing and Operations (DevOPS)**
4. **Introduction to Intelligent Automation -** Data Science, AI and ML

**Automation for Testing and Operations Track:**

The goal of this track is to develop automation skills that are mandatory for innovation to become an intrapreneur and entrepreneur in the world of testing, operating, and delivering high-quality application products and solutions effectively, efficiently, and faster.

1. **Web Application** – Scripting and Framework with Python and Selenium, Visual Testing with Applitools, Selenide with Java, BDD Framework with Python, Parallel execution and scaling.
2. **Mobile Application** – Appium with Python, XCTest, Cloud-based devices and configurations, Calabash with Java, Tools and Framework – Kobiton, TestProject.
3. **Automation Framework** – Page Object Model with PageFactory.
4. **Application Programmable Interface (API)** – Postman, Rest Assured with Java, Karate with Java, Tavern with Python, DB integration with Rest Testing, Mocha and Chai with NodeJS.
5. **Integration testing** – Charles.
6. **Application Security Testing** – Different types and tools like SonarCube.
7. **Unit and Functional Testing**- JUnit, Jasmine.
8. **Web Socket Testing** – JMeter, Test Socket
9. **Desktop Application** – Winium, and Sikuli
10. **Database Testing and Application Performance** – Redshift/Snowflake testing with Python BDD Framework, JMeter with Java, Locust with Python, TOAD.
11. **Logging, Analytics, and Management** – Logstash, Log4j / Slf4j, Kibana, and Elastic Search, Jira, TestLink, and Ride GUI
12. **DevOps**
    1. **Source Code Management** - Git
    2. **Build Management -** Gradle
    3. **Code Quality and Test management** – Cucumber
    4. **Infrastructure Management** – AWS
    5. **Deployment & Configuration Management** – Ansible
    6. **CICD** – Jenkins
13. **Analytics, and AI / ML for testing and operations**

**Intelligent Automation Track:**

The goal of this track is to develop hands-on experience to become a Data Analyst, Data Engineer, Data scientist, and AI/ML intelligent automation expert. It will build the foundation for future innovation in this subject within your company as an Intrapreneur and an Entrepreneur.

1. **Introduction to Data science** – Moneyball, Voter Turnout, and Engineering Solutions
2. **Statistics, AI, Machine Learning, Deep Learning, Software engineering for Data Science** – NumPy, PyTorch, Jupyter, and Apache Spark
3. **Structure and output of a Data Science Project / Experiment**
4. **Defining Success -** Four Secrets of a successful Data Science Experiment
5. **Data Science Toolbox –** Big data, Datasets, Stat Models, Refining data, Training models, Dataflow diagrams level 0-4, validation and deployment
6. **Data Science for Startup, SME, and Large Organizations**
7. **Common Challenges and difficulties for Data Science –** Interaction and internal
8. **Managing Data Analysis** – Forecasting, replenishment, and optimization
9. **The Data Analysis Iteration** – HLD, data structures, data flow and predictive analytics
10. **Epicycle of Analysis** – Process and data flow diagrams
11. **Exploratory Data Analysis and Modeling**
    1. What Are the Goals of Formal Modeling?
    2. Associational Analyses
    3. Prediction Analyses
12. **What You’ve Gotten Yourself Into**
    1. Data double duty
    2. Multiplicity
13. **Randomization versus observational studies**
    1. The Data Pull is Clean
14. **The Experiment is Carefully Designed: Principles**
    1. Causality
    2. Confounding
15. **The Experiment is Carefully Designed: Things to Do**
    1. A/B testing
    2. Sampling
    3. Blocking and Adjustment
16. **Results of the Analysis Are Clear**
    1. Multiple comparisons
    2. Effect sizes, significance, modeling
    3. Comparison with benchmark effects
    4. Negative controls
17. **The Decision is Obvious**
    1. The decision is (not) obvious
    2. Estimation target is relevant

Points taken off from Web Application:

Parallel Tests and scaling in selenium grid with docker, WebDriverIO with Javascript. Data driven Frameworks and Reporting, Cross Browser Test Implementation.