**Day 1**

1. Overview of DevOps
   1. Definition of DevOps
   2. Devops and agile
   3. Devops challenges
   4. Tools compatibility and networking
   5. Cost of Devops
   6. Devops Tools
   7. Software requirement, architecture and design tools
   8. Code development tools
   9. Software testing tools
   10. Release management tools
   11. Configuration and monitoring tools
   12. Cloud computing
   13. IaaS
   14. PaaS
   15. SaaS
   16. Exercise: Setting up your first cloud based virtual machine in GCP and AWS
   17. Setting up lab in AWS and MS Azure
   18. Install Maven on GCP
   19. Information Security Threats
   20. Defence in depth
   21. Minimize attack surfaces
   22. Fail secure
   23. Least privilege
   24. Security through obscurity
   25. Segregation of duties
   26. Token authentication
2. Virtualization
   1. Overview, advantages and history
   2. Virtualization software – type 1 hypervisor
   3. Virtualization software – type 2 hypervisor
   4. Linux virtualization
   5. Control groups
   6. Containers
   7. OS virtualization and UFS
3. Lightweight Containers
   1. Dockers
   2. Linux docker
   3. OS X docker
   4. Windows docker
   5. Docker architecture
   6. Images and containers
   7. Daemon container
   8. Pulling Containers
   9. Running containers
   10. Attaching containers
   11. Stopping & pausing containers
   12. Starting containers
   13. Removing containers
   14. Removing images
   15. Exercise: Install docker, pull linux container, run container, try all of the docker commands seen in previous slides
   16. Adding packages to containers, verifying and committing
   17. Build your first image – Dockerfile
   18. Dockerfile commands – shell, copy, add, run, env, cmd, build
   19. Docker registry
   20. Private registry
   21. Creating and using a private registry
   22. Dockerhub registry
   23. Using Dockerhub registry
   24. Exercise: create a docker image using Dockerfile, run container from this image and connect using SSH
   25. Exercise: Install and run docker registry, pull an image, start it in container and push this image in registry
   26. Docker machine
   27. Docker networking
   28. Docker container networks
   29. Default
   30. Bridge
   31. Host
   32. User defined
   33. Remote access
   34. Firewall rules

**Day 2**

1. Software Development Tools
   1. Source control systems introduction
   2. File system based source control system
   3. Client server based source control system
   4. Distributed source control system
   5. CVS
   6. Subversion
   7. Exercise: Install subversion, create a project directory, add a new file, commit changes
   8. Git
   9. Gitk
   10. Exercise: Install Git, create a project directory, add a new file, commit, push changes to master
   11. Build Tools
   12. Make
   13. Ant
   14. Maven
   15. Continuous integration
   16. Introduction to Sonarcube
   17. Feature, fitment in continuous integration
   18. Jenkins
   19. Continuous integration using Jenkins
   20. Lab: Install Jenkins, setup Jenkin pipeline to demonstrate CI – continuous builds, run TDD test cases (may be), connect pipeline with Git
2. Test Driven Development
   1. Introduction, history of testing
   2. Origin of TDD
   3. Use and benefits of TDD
   4. IDEs – features, benefits
   5. Eclipse
   6. IntelliJ
   7. Visual Studio
   8. XCode
   9. Centralized quality check of projects using Sonarcube
   10. IDEs and support for testing
   11. IDEs and deployment
   12. Unit testing frameworks
   13. TDD development and setting up tests
   14. Refactor
   15. Exercise: Write TDD test cases using Junit in Eclipse.
   16. BDD – behaviour driven development
3. Quality of Code
   1. Principles
   2. Technical debt
   3. Code smells
   4. Code smell – comments
   5. Code smell – long methods
   6. Code smell – duplicate code
   7. Code smell – hard coded values
   8. Code smell – large classes
   9. Code smell – too many parameters
   10. Code smell – dead code
   11. Code smell – empty blocks
   12. Code smell – missing paths
   13. Code smell – complex expressions
   14. Code smell – structural complexity
   15. Code smell – cyclomatic complexity
   16. Code smell – design level complexity
   17. Code smell – class complexity
   18. Code quality tools
   19. Running code quality tools
   20. Coding standards
   21. Code coverage
   22. Test coverage tools
   23. SQALE
   24. Sonarcube
   25. Exercise: run different code quality tools, fix issues in code and run again

**Day 3**

1. Software Integration
   1. Integration testing
   2. Strategy & dependencies
   3. Mocking
   4. Mocking frameworks
   5. Exercise: install mocking framework, implement mocking framework on java code
   6. Docker swarm
   7. Creating and managing swarm
   8. Joining swarm
   9. Docker swarm services
   10. Updating and removing docker swarm services
   11. Exercise: create a bridged network, list networks, add networking tools package, connect new bridged network to container, disconnect
   12. Docker compose
   13. Run compose
   14. Build images using docker compose
   15. Exercise: build a docker image with SSH, create data volume, create docker compose to run image with volume mapped, run container using docker compose
   16. Integration tools
   17. Apache Camel
   18. Apache ActiveMQ
   19. Puppet
   20. Teamcity – Intro
   21. Teamcity fitment in CI & CD
   22. Lab: Example using Teamcity
2. Software Support
   1. Configuration management
   2. Build management
   3. Support environment
   4. Defect tracking
   5. Exercise: pull configuration management image, start server, use client to communicate
   6. Kubernetes
3. Monitoring & Performance
   1. Monitoring overview
   2. Scheduling
   3. Cron scheduling
   4. Autosys
   5. Scheduling management
   6. Nagios
   7. Nagios agents
   8. Genios
   9. Performance parameters of software systems
   10. Optimization
   11. Compiler optimization
   12. Performance testing
   13. Java profiling
   14. Profiling tools
   15. Combining Jenkins, Docker and Puppet – landscape view