Lesson Plan

14MT22501: SERVICE ORIENTED ARCHITECTURE

Name of the Faculty Member : Mr. P. SRINIVASA REDDI

Class& Semester : M.Tech (SE) - II Semester

S. No.	Topic	No. of periods	Book(s) followed	Topics for self study
UNIT – I: SOA and Web Services Fundamentals				
1.	**Introducing SOA**: Fundamental SOA	1	T1	Case Studies : RailCo Ltd and Transit Line Systems Inc Ref:T1
2.	Common Characteristics of Contemporary SOA	3	T1	
3.	Common tangible benefits of SOA, Common pitfalls of adopting SOA	1	T1	
4.	**Web Services and primitive SOA**-The Web Services frame work	1	T1	
5.	Services	2	T1	
6.	Service descriptions	2	T1	
7.	Messaging	1	T1	
Total periods required:		**11**		
UNIT – II: SOA and WS-* Extensions				
8.	**Web Services and Contemporary SOA (Part I-Activity management and Composition**) -Message Exchange Patterns	2	T1	Case studies : Message Exchange Pattern, Service activity, Coordination and Reliable Messaging Ref:T1
9.	Service Activity , Coordination	2	T1	
10.	Atomic transactions, Business Activities	2	T1	
11.	Orchestration, Choreography	2	T1	
12.	**Web Services and Contemporary SOA (Part-II-Advanced Messaging, Metadata and Security) -**Addressing, Reliable messaging	2	T1	
13.	Correlation, Policies	1	T1	
14.	Metadata exchange	1	T1	
15.	Security, Notification and eventing.	1	T1	
Total periods required:		**13**		
UNIT – III: Principles, Service Layers and Planning				
16.	**Principles of Service-Orientation -** Anatomy of SOA	1	T1	Case studies: Common Principles of Service – Orientation, Application Service Layer, Business Service Layer, the Top-down strategy and the bottom-up strategy. Ref:T1
17.	Common Principles of Service–Orientation	3	T1	
18.	Service Orientation and Object Orientation	1	T1	
19.	**Service Layers -** Service-Orientation and Contemporary SOA,	1	T1	
20.	Service Layer abstraction, Application Service Layer	2	T1	
21.	Business Service Layer	1	T1	
22.	Orchestration Service Layer	1	T1	
23.	Agnostic Services, Service Layer Configuration Scenarios	2	T1	
24.	**SOA Delivery Strategies -** SOA delivery lifecycle phases	1	T1	
25.	The Top-down strategy	1	T1	

26.	The bottom-up strategy	1	T1	
27.	The agile strategy	1	T1	
Total periods required:		**12**		
UNIT – IV: Building SOA (Analysis)				
28.	**Service Oriented Analysis (Part I-Introduction) -**Introduction to Service Oriented Analysis	1	T1	Case studies: Deriving Business Services, Service Modeling and Contrasting Service modeling approaches. Ref:T1
29.	Benefits of a Business Centric SOA	1	T1	
30.	Deriving Business Services	2	T1	
31.	**Service Oriented Analysis (Part-II-Service Modeling) -** Service Modeling	1	T1	
32.	Service Modeling guidelines	2	T1	
33.	Classifying Service model logic	1	T2	
34.	Contrasting Service modeling approaches	1	T2	
Total periods required:		**9**		
UNIT – V: Building SOA (Design)				
35.	**Service Oriented Design (Part I-Introduction)-**Introduction to Service-Oriented design	1	T1	Case studies: Service Design, Entity-centric business Service Design, Application Service Design. Ref:T1
36.	WSDL related XML Schema language basics	1	T1	
37.	WSDL language basics, SOAP language basics	2	T1	
38.	Service interface design tools	1	T1	
39.	**Service Oriented Design (Part III-Service Design)** - Service Design overview	1	T1	
40.	Entity-centric business Service Design	1	T1	
41.	Application Service Design	2	T1	
42.	Task-centric business Service Design, Service Design guidelines	2	T1	
43.	**Service Oriented Design (Part IV-Business Process Design):** WS-BPEL language basics	2	T1	
44.	WS- Coordination overview	1	T1	
45.	Service Oriented Business process Design	1	T1	
Total periods required:		**15**		
Grand total periods required:		**60**		

TEXT BOOK:

T1: Thomas Erl, "*Service-Oriented Architecture - Concepts, Technology, and Design",* Pearson, 2008.

REFERENCE BOOKS:

R1: Shankar Kambhampaty, *"Service Oriented Architecture for Enterprise and Cloud Applications,"* Wiley- India, 2012

R2: Eric Newcomer and Greg Lomow, *"Understanding SOA with Web Services,"* Pearson Education, 2007.

R3: M. Rosen and others, *"Applied SOA,"* Wiley India Pvt. Ltd, 2009.

Signature of the Faculty Member **Signature of the HOD**

Lesson Plan

14MT22502: SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

Name of the Faculty Member : Mr. K. KHAJA BASEER

Class& Semester : M.Tech (SE) - II Semester

S. No.	Topic	No. of periods	Book(s) followed	Topics for self study
UNIT – I: Envisioning Architecture				
1.	What Is Software Architecture, What Software Architecture Is and What It Isn't	1	T1	Case Study 1: Key Word in Context apply on Shared Memory, Events, ADT, and Dataflow styles Case 2: A Fresh View of Compilers apply on heterogeneous architectures Case 3: A Layered Design with Different Styles for the Layers Case 4: An Interpreter Using Different Idioms for the Components Case 5: A Blackboard Globally Recast as Interpreter **[Reference:1] [Read Martin-' Patterns' and Mary Shaw-'An Introduction to Software Architecture' papers]** **Assessment Method:** Project Team Formed and Presentation by teams
2.	Architectural Structures and Views	1	T1	
3.	What Makes a Good Architecture, Why Is SA Important, Where Do Architectures Come From	3	T1	
4.	Software Processes and the ABC	1	T1	
5.	Architectural Patterns, Reference Model, and Reference Architecture. Pattern System – What is Pattern System, Pattern Classification, Pattern Selection	1	R1	
6.	Introduction to Architectural Patterns, Pipes & Filter	3	R1	
7.	Model-View-Controller	2	R1	
	Total periods required:	**12**		
UNIT – II: Creating an Architecture				
8.	Understanding the Requirements – Functionality & Architecture, Architecture & Quality Attributes (QA), System QAs	2	T1	(i) J2EE/EJB framework: A Case Study of an Industry-Standards Computing Infrastructure. (ii) ASP.NET MVC 3 /4 frameworks [**Reference:** ASP.NET MVC - http://www.asp.net/mvc http://pluralsight.com/training/Player?author=scott-allen&name=mvc4-building-m6-ajax&mode=live&clip=0&course=mvc4-building] **Assessment Method**: illustrate and Submit a report of 4 pages
9.	QA Scenarios in Practice	2	T1	
10.	Business & Architecture Qualities	1	T1	
11.	Achieving Qualities	2	T1	
12.	Designing the Architecture – Architecture in the Life Cycle	1	T1	
13.	Attribute Driven Design (ADD)	2	T1	
14.	Documenting Software Architectures – Uses of Architectural Documentation, Views	2	T1	
15.	Reconstructing Software Architectures – Information Extraction, Database Construction, View Fusion, and Reconstruction.	2	T1	
	Total periods required:	**14**		
UNIT -III: Evaluating & Reconstructing the Architecture and Moving from one system to many				
16.	Evaluating the Architecture – The ATAM	2	T1	Comparison of ATAM,CBAM

17.	The CBAM	1	T1	and SAAM **[Reference:2]** **Assessment Method**: Chart / Mind Map submission using Microsoft Visio 2010
18.	Moving from one system to many – Software Product Lines	2	T1	
19.	Building Systems from off the shelf components	2	T1	
20.	Software Architecture in the future	1	T1	
Total periods required:			**08**	
UNIT – IV: Introduction to Design Pattern and Creational Patterns				
21.	**Introduction to Design Patterns** What is Design Pattern(DP)	1	T2	Write a Java Program on Singleton A Case Study: Design a Document Editor using Lexi **[Text Book 2 and Reference 1]** **Assessment Method**: Assignment
22.	Design Patterns in Smalltalk MVC	1	T2	
23.	Describing DPs	1	T2	
24.	The Catalog of DPs & Organizing the Catalog	1	T2	
25.	How DP Solve Design Problem & How to Select & Use of a DP	1	T2	
26.	**Creational Patterns:** Abstract Factory & Builder	2	T2	
27.	Factory Method & Prototype	2	T2	
Total periods required:			**09**	
UNIT – V: Structural and Behavioral Patterns				
28.	**Structural Patterns:** Adapter	1	T2	Write the 14 elements for the following Patterns: Bridge, Façade, Proxy, Chain of Responsibility, Interpreter, Memento, Strategy, Visitor **[Reference 1 and 4]** **Assessment Method**: Assignment
29.	Composite & Decorator	2	T2	
30.	Flyweight	1	T2	
31.	**Behavioral Patterns:** Command	1	T2	
32.	Iterator & Mediator	2	T2	
33.	Observer & State	2	T2	
Total periods required:			**11**	
Grand total periods required:			**54**	

TEXT BOOKS:

T1: Len Bass, Paul Clements & Rick Kazman, *"Software Architecture in Practice,"* 2nd Edition, Addison-Wesley, Pearson Education, 2003

T2: Erich Gamma, Richard Helm, Ralph Johnson, & John Vlissides *"Design Patterns: Elements of Reusable Object-Oriented Software,"* Pearson Education, 1995

REFERENCE BOOKS:

R1: Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad & Michael Stal, *"Pattern-Oriented Software Architecture: A System of Pattern,"* Volume 1, John Wiley & Sons; 2001.

R2: Mary Shaw & David Garlan, *"Software Architecture: Perspectives on an Emerging Discipline,"* Prentice Hall, 1996

R3: Eric Freeman & Elisabeth Freeman, *"Head First Design patterns,"* O'Reilly, 2004.

R4: Richard N. Taylor, N. Medvidovic & Eric M. Dashofy, *"Software Architecture: Foundations, Theory, and Practice,"* Wiley, 2006

Signature of the Faculty Member **Signature of the HOD**

Lesson Plan

14MT22503: SOFTWARE MEASUREMENT AND METRICS

Name of the Faculty Member : Dr. K. Ramani

Class& Semester : M.Tech (SE) - II Semester

S. No.	Topic	No. of periods	Book(s) followed	Topics for self study
UNIT – I: Fundamentals of Measurement				
1.	Introduction, Measurement in Everyday Life	1	T1	Explain Statistical Operations on Measures Ref: T1: Page no:57-59
2.	Measurement in Software Engineering	2	T1	
3.	Scope of Software Metrics	1	T1	
4.	Representational Theory of Measurement	1	T1	
5.	Measurement and Models	2	T1	
6.	Measurement Scales and Scale Types	3	T1	
	Total periods required:	**10**		
UNIT – II: Analysis of Measurement				
7.	Classifying Software Measures	1	T1	Identify Questions and metrics for Goals: Improve Software Estimation Minimize Development Cost Ref- http://spc.ca/resources/metrics/step3.pdf
8.	Determining What to measure	2	T1	
9.	Applying Frame Work	2	T1	
10.	Software Measurement Validation	2	T1	
11.	Software Metrics Data Collection: Good Data, Definition of Data	2	T1	
12.	Collecting the data, Storing and Extracting Data	2	T1	
	Total periods required:	**11**		
UNIT -III: Product Attributes				
13.	Measuring Internal Product Attributes: Size-Aspects of software size	2	T1	Identify Difficulties with General Complexity Measures, Data structure. Ref-T1: Page no:322-324
14.	Length, Reuse	2	T1	
15.	Functionality, Complexity	3	T1	
16.	Structure- Types of structural measures, Control-flow structure	2	T1	
17.	Modularity and information flow attributes	3	T1	
	Total periods required:	**12**		
UNIT – IV: Measurement and Management				
18.	Measuring External Product Attributes: Modeling Software Quality	2	T1	Explain Object-Oriented Metrics Ref-T1: Page no:317-319
19.	Measuring Aspects of Quality	3	T1	
20.	Object-Oriented Concepts and Constructs	1	T2	
21.	Design and complexity metrics	2	T2	
22.	Productivity Metrics , Quality and Quality Management Metrics	2	T2	
23.	Lessons Learned from OO Projects	1	T2	
	Total periods required:	**11**		
UNIT – V: Quality Metrics				
24.	Product Quality Metrics	2	T2	Identify the Metrics Programs at Motorola, H-P and IBM Rochester Ref-T2: Page no:110-116
25.	In-Process Quality Metrics	1	T2	
26.	Metrics for software Maintenance	2	T2	
27.	Collecting Software Engineering Data	2	T2	

28.	Applying the Seven Basic Quality Tools in Software Development	4	T2	
Total periods required:		**11**		
Grand total periods required:		**55**		

TEXT BOOKS:

T1: Fenton, Pfleeger, *"Software Metrics",* 2nd Edition, Thomson, 2005.

T2: Stephen H. Kan: *"Metrics & Models in Software Quality Engineering",* 2nd Edition, Addison Wesely, 2011.

REFERENCE BOOKS:

R1: Linda M. Laird and Carol Brennan: *"Software Measurement and Estimation - A Practical Approach,"* IEEE Computer Science Press and Wiley Inter science, 2006.

R2: C. Ravindranath Pandian: *"Software Metrics: A guide to Planning Analysis, and Implementation,"* Auerbach Publications, 2005.

Signature of the Faculty Member **Signature of the HOD**

Lesson Plan

14MT22504: SOFTWARE SECURITY ENGINEERING

Name of the Faculty Member : Ms. V. Jyothsna

Class& Semester : M.Tech (SE) - II Semester

S. No.	Topic	No. of periods required	Book(s) followed	Topics for self study
UNIT-I: IMPORTANCE OF SECURITY IN SOFTWARE				
1.	**Security a software Issue:** Introduction, The problem, Software Assurance and Software Security	2	T1	Type of security strategy to be followed by analyzing enterprise perceptive. Incorporating security assurance cases into the SDLC. Ref:T1
2.	Threats to software security, Sources of software insecurity	2	T1	
3.	Benefits of Detecting Software Security.	1	T1	
4.	**What Makes Software Secure:** Introduction, Properties of Secure Software	2	T1	
5.	Influencing the security properties of software	3	T1	
6.	Asserting and specifying the desired security properties	2	T1	
	Total no of periods required:	**12**		
UNIT-II: REQUIREMENTS ENGINEERING				
7.	**Requirements Engineering for secure software:** Introduction, Misuse and abuse cases	1	T1	An Abuse Case Example. Requirements Prioritization Technique Comparison Ref:T1
8.	The SQUARE process Model	2	T1	
9.	SQUARE sample outputs	2	T1	
10.	Requirements elicitation	2	T1	
11.	Requirements prioritization	3	T1	
	Total no of periods required:	**10**		
UNIT-III: SECURITY PRINCIPLES IN SDLC				
12.	**Secure Software Architecture and Design:** Introduction	1	T1	Sources of Additional Information on Software Security Testing Ref:T1
13.	Software Security practices for Architecture and Design: architectural risk analysis	3	T1	
14.	Software security knowledge for Architecture and Design: Security principles, Security guidelines and Attack patterns	2	T1	
15.	**Secure coding and Testing:** Introduction, Code analysis	2	T1	
16.	Software Security testing	2	T1	
17.	Security testing considerations throughput the SDLC.	2	T1	
	Total no of periods required:	**12**		
UNIT-IV: SECURITY AND COMPLEXITY				
18.	**Security and Complexity:** System Assembly Challenges: Introduction	1	T1	Attacker behavior that leads to Security failure. Ref:T1
19.	Security failures	1	T1	
20.	Functional and attacker perspectives for security analysis	4	T1	
21.	System complexity drivers and security.	4	T1	
	Total no of periods required:	**10**		

UNIT-V: GOVERNANCE AND MANAGING				
22.	**Governance and Managing for More Secure Software:** Introduction, Governance and security	2	T1	Characteristics of Effective Security Governance and Management. Resources required for Security and Project Management Ref:T1
23.	Adopting an enterprise software security framework	3	T1	
24.	How much security is enough	1	T1	
25.	Security & Project Management	3	T1	
26.	Maturity of Practice	3	T1	
	Total no of periods required:	**12**		
	Grand total periods required:	**56**		

TEXT BOOKS:

T1: Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, and Nancy R. Mead, *"Security Engineering: A Guide for Project Managers,"* Pearson Education, 2009

REFERENCE BOOKS:

R1: Gary McGraw, *"Software Security: Building Security In,"* Addison-Wesley, 2006.

R2: Mark Dowd, John McDonald, Justin Schuh, *"The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities,"* 1st Edition, Addison-Wesley, 2006.

R3: G. Hoglund, G. McGraw, *"Exploiting Software: How to Break Code,"* Addison-Wesley, 2004.

Signature of the Faculty Member **Signature of the HOD**

Lesson Plan

14MT22505: SOFTWARE TESTING

Name of the Faculty Member **:** Mr. S. SREENIVASA CHAKRAVARTHI

Class& Semester **:** M.Tech (SE) - II Semester

S. No.	Topic	No. of periods	Book(s) followed	Topics for self study
UNIT – I: SOFTWARE TESTING CONCEPTS – I				
1.	A Perspective on Testing: Definitions, Test cases Identifying Test cases, Error & Fault Taxonomies; Levels of Testing.	1	T1	Retrospect Functional Testing in terms of Testing Effort, Test Efficiency, and Testing Effectiveness Retrospect Structural Testing in terms of Gaps and Redundancies, Metrics for Method Evaluations using a case study
2.	Functional Testing: - Boundary Value Testing: Boundary Value Analysis, Test case Examples, Random Testing.	2	T1	
3.	Functional Testing: - Equivalence Class Testing: Equivalence Classes, Test case Examples,	2	T1	
4.	Functional Testing: - Decision Table based Testing: Decision Tables, Test case Examples,.	1	T1	
5.	Guidelines and Observations in Functional Testing	1	T1	
6.	Structural Testing: - Path Testing: DD-Paths, Test Coverage Metrics, Basis Path Testing, Guidelines and Observations.;	2	T1	
7.	Dataflow Testing: Define/Use Testing & Examples, Slice based Testing & Examples	2	T1	
8.	, Guidelines and Observations in Structural Testing	1	T1	
Total periods required:		**12**		
UNIT – II: SOFTWARE TESTING CONCEPTS – II				
9.	Integration Testing: Scenario Testing, Defect bash.	2	T2, R1	Develop Functional and System test scenarios and test cases for Calculator application.
10.	System and acceptance testing: functional, non-functional testing.	2	T2	
11.	Performance testing,	1	T2	
12.	Regression Testing: Definition, Types, When & How to do regression testing.	2	T2	
13.	Internationalization Testing: Introduction, Test Phases of Internationalization testing, and Enabling testing.	3	T2	
14.	Locale Testing, Language testing, Localization testing.	2	T2	
Total periods required:		**12**		
UNIT -III: SOFTWARE TESTING CONCEPTS – III				
15.	Ad-hoc testing	1	T2	Arrive on Non-Functional Testing strategy and Test cases for Usability testing and Accessibility Testing
16.	Buddy testing, Pair Testing, Exploratory Testing,	1	T2	
17.	Iterative testing Agile and Extreme Testing	1	T2	
18.	Testing of Object-oriented systems: Introduction,	2	T2	
19.	Primer on object oriented software, Differences in OO testing.		T2	
20.	Usability Testing Introduction & Approach	1	T2	

21.	When to do usability testing,	1	T2	
22.	How to achieve usability and Quality factors for usability	1	T2	
23.	Importance of Accessibility testing, accessibility testing approaches,	1	T1	
24.	Tools for Usability and Accessibility testing.	1	T2	
	Total periods required:	**10**		
UNIT – IV: SOFTWARE TEST MANAGEMENT AND METRICS				
25.	Test planning,	1	T2	Develop a Test plan & Test metrics for any Web application.
26.	Test Management,	1	T2	
27.	Test Process and Reporting	1	T2	
28.	Software Test matrices and Measurement: Type of Metrics	2	T2	
29.	Project Metrics	1	T2	
30.	Productivity Metrics	2	T2	
31.	Progress Metrics.	1	T2	
32.	Release Metrics.	1	T2	
	Total periods required:	**10**		
UNIT – V: SOFTWARE TEST AUTOMATION				
33.	Test Automation: Scope of Automation	1	R2	Descriptive programming in QTP, Simplified scripting in RFT and Case studies on Cloud testing.
34.	Design and Architecture of automation	1	T2	
35.	Process Model for Automation.	1	T2	
36.	Load Runner	2	R3	
37.	Selenium	2	R6	
38.	QTP	2	R3	
39.	RFT & RQM	2	R5	
40.	Bugzilla	2	R4	
	Total periods required:	**13**		
	Grand total periods required:	**57**		

TEXT BOOKS:

T1: Paul C. Jorgensen, “*Software Testing: A Craftman’s Approach,*” 3rd Edition, Auerbach Publications, 2008.

T2: Srinivasan Desikan and Gopalswami Ramesh “*Software Testing: Principle and Practices,*” 1st Edition, Pearson Education, 2008.

REFERENCE BOOKS:

R1: M. G. Limaye, *“Software Testing: Principles and Techniques and Tools,”* 1st Edition, Tata Mc Graw – Hill Education, 2012.

R2: Ilene Burnstein, *“Practical Software Testing,”* Springers-Verilog International Edition, 2003.

R3: Dr. K. V. K. K.Prasad, *“Software Testing Tools,”* 1st Edition, Dreamtech, 2004.

R4: The Bugzilla Guide - 4.4.2+ Release (http://www.bugzilla.org/docs/4.4/en/pdf/Bugzilla-Guide.pdf)

R5: Introduction to IBM Rational Functional Tester 6.1 (http://www.ibm.com/developerworks/rational/library/04/r-3228/3228.html)

R6: Selenium-IDE — Selenium Documentation (http://docs.seleniumhq.org/docs/)

Signature of the Faculty Member **Signature of the HOD**

Lesson Plan

14MT22506: BIG DATA TECHNOLOGIES

Name of the Faculty Member : Mr. A. SRINIVASULU

Class& Semester : M.Tech (SE) - II Semester

S. No.	Topic	No. of periods required	Book(s) followed	Topics for self study
UNIT-I: INTRODUCTION TO BIG DATA				
1.	Introduction to Big data: The Evolution of Big Data, What Is Big Data, Why Big Data Matters.	2	T2	Hadoop Installation http://hadoop.apache.org/docs/r2.3.0/hadoop-project-dist/hadoop-common/SingleNodeSetup.html
2.	Big Data Sources, The Big data Revolution, Security, Compliance, Auditing and Protection.	2	T2	
3.	Meet Hadoop: Data Storage and Analysis, Comparison with Other Systems, A Brief History of Hadoop, Apache Hadoop and the Hadoop Ecosystem.	2	T2	
4.	MapReduce: A Weather Dataset, Analyzing the Data with Unix Tools, Analyzing the Data with Hadoop, Scaling Out, Hadoop Streaming, Hadoop Pipes.	2	T2	
5.	The Hadoop Distributed File system: The Design of HDFS, HDFS Concepts, The Command-Line Interface	2	T2	
6.	Hadoop File systems, The Java Interface, Data Flow, Parallel Copying with distcp, Hadoop Archives.	2	T2	
	Total no of periods required:	**12**		
UNIT-II: HADOOP I/O				
7.	Hadoop I/O: Data Integrity, Compression, Serialization, File-Based Data Structures.	1	T1	Hadoop Features http://hadoop.apache.org/releases.html
8.	Developing a MapReduce Application: The Configuration API, Configuring the Development Environment.	2	T1	
9.	Writing a Unit Test, Running Locally on Test Data, Running on a Cluster, Tuning a Job, MapReduce Workflows.	2	T1	
10.	How MapReduce Works: Anatomy of a MapReduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task Execution.	3	T1	
11.	MapReduce Types and Formats: MapReduce Types, Input Formats, Output Formats.	3	T1	
	Total no of periods required:	**12**		
UNIT-III: MAPREDUCE FEATURES				
12.	MapReduce Features: Counters, Sorting, Joins, Side Data Distribution, MapReduce Library Classes.	2	T1	Moore's law and open source http://www.mooreslaw.org/
13.	Setting Up a Hadoop Cluster: Cluster Specification, Cluster Setup and Installation, SSH Configuration	3	T1	

14.	Hadoop Configuration, YARN Configuration, Security, Benchmarking a Hadoop Cluster, Hadoop in the Cloud.	2	T1	
15.	Administering Hadoop: HDFS, Monitoring, Maintenance.	2	T1	
16.	Pig: Installing and Running Pig, Comparison with Databases, Pig Latin, User-Defined Functions, Data Processing Operators, Pig in Practice.	3	T1	
	Total no of periods required:	**12**		
UNIT-IV: HIVE				
17.	Hive: Installing Hive, Running Hive, Comparison with Traditional Databases,	2	T1	Continuing problems with incident detection and response http://en.wikipedia.org/wiki/Computer_security_incident_management
18.	HiveQL, Tables, Querying Data, User - Defined Functions.	2	T1	
19.	HBase: HBasics, Concepts, Installation, Clients, HBase versus RDBMS, Praxis.	2	T1	
20.	ZooKeeper: Installing and Running ZooKeeper, the ZooKeeper Service.	3	T1	
21.	Building Applications with ZooKeeper, ZooKeeper in Production.	2		
	Total no of periods required:	**11**		
UNIT-V: SQOOP				
22.	Sqoop: Getting Sqoop, Generated Code, Database Imports: A Deeper Look, Working with Imported Data.	2	T1	Mahout : Machine Learning Algorithms https://mahout.apache.org/
23.	Importing Large Objects, Performing an Export, Exports: A Deeper Look.	3	T1	
24.	Case Studies: Best Practices for Big Data Analytics, Hadoop Usage at Last.fm.	2	T1	
25.	Hadoop and Hive at Facebook, Nutch Search Engine, Log Processing at Rackspace, Cascading.	3	T1	
26.	TeraByte Sort on Apache Hadoop, Using Pig and Wukong to Explore Billion-edge Network Graphs.	3	T1	
	Total no of periods required:	**11**		
	Grand total periods required:	**58**		

TEXT BOOKS:

T1: Tom White, *"Hadoop: The Definitive Guide,"* 3rd Edition, Oreilly and Yahoo press, 2012.

REFERENCE BOOKS:

R1: Frank J. Ohlhorst, *"Big Data Analytics: Turning Big Data into Big Money,"* Wiley Publication, 2012.

R2: Kevin Roebuck, *"Big Data: High-Impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors,"* Tebbo publisher, 2011.

R3: Alex Holmes, *"Hadoop in Practice,"* Manning Publications Publisher, 2012.

Signature of the Faculty Member **Signature of the HOD**