Autonomous Program Structure of
Final Year B. Tech. (Information Technology)
Academic Year: 2019-2020 Onwards

Final Year B. Tech. (IT) Semester – I										
Course Code	Course Title	Teaching Scheme Hours /Week			Examination Scheme				Marks	Credit
		Lecture	Tutorial	Practical	In Semester	End Semester	Oral	Practical		
IT 4101	Software Architecture & Design Patterns	3	0	0	50	50	0	0	100	3
IT 4102	Cloud Computing	3	0	0	50	50	0	0	100	3
HS 4101	Green Computing*	3	0	0	50	50	0	0	100	3
OE 4101	Open Elective –I	3	0	0	50	50	0	0	100	3
IT 4103	Software Architecture & Design Patterns Laboratory	0	0	2	0	0	50	0	50	1
IT 4104	Project Phase-I	0	2	14	100	0	50	0	150	9
	Total	**12**	**2**	**16**	**300**	**200**	**100**	**0**	**600**	**22**
	Grand Total	**30**			**600**				**600**	**22**

*Advanced Entrepreneurship Development**

**Prerequisite: Basic Course ED

MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

DEAN ACADEMICS
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

Principal
MKSSS's Cummins College of Engg. For Women, Karvenagar, Pune-52.

APPROVED BY
Governing Body Members
MKSSS's Cummins College of Engineering for Women Karvenagar, Pune-411052

IT 4101 Software Architecture and Design Patterns

Teaching Scheme:
Lectures: **3** hrs/week
Tutorial: NIL

Examination Scheme:
In-Semester: **50** marks
End-Semester: **50** marks
Credits: **3**

Prerequisites: Object Oriented Paradigms, Software Engineering

Course Objectives:

Familiarize students with

1. Concepts of software architecture
2. Different types of software architectural styles
3. Concepts and applications of design patterns.
4. Different types of design patterns

Course Outcomes:

Students should be able to

1. Analyze and suggest architecture design for an application
2. Apply design patterns to software design
3. Evaluate and select appropriate design pattern for a situation
4. Compare the performance of the software on inclusion of various design patterns.

Unit – I: Software Architecture **(07)**

Overview of software Architecture, What drives software architecture, Quality attributes, Architecture design, Architecture documentation

Unit – II: Architectural Patterns **(07)**

Client server multitier architectural pattern, Even driven architectural pattern, Service Oriented Architectures, Component based architecture

Unit – III: Role of design patterns in architecture design **(07)**

Introduction to architecture design, introduction to design patterns, Types of design patterns Abstract factory, builder, factory method, singleton design patterns

Unit – IV: Creational Design Patterns **(07)**

Abstract factory, builder, factory method, singleton design patterns with case study

Unit – V: Structural Design Patterns (07)

Adapter, bridge, composite, facade, decorator, chain of responsibility with case study

Unit – VI: Behavioral Design Patterns (07)

State, Observer, Strategy, template method with case study,

Text Books:

1. Craig Larman, Applying UML and Patterns, Pearson Education, Second Edition, ISBN: 9780130925695.
2. Elizabeth Freeman, Kathy Seirra, Head first design patterns O'Reilly Media ISBN 0596007124

Reference Books:

1. Len Bass, Paul Clements, Rick Kazman Software Architecture in Practice, Pearson Education, ISBN: 978-81-7758-996-2
2. Eric Gamma and other authors Design Patterns Elements of reusable object oriented software Addison Wesley Professional Series ISBN 0-202-63361-2

IT 4102 Cloud Computing

Teaching Scheme:
Lectures: **3** hours/week
Tutorial: NIL

Examination Scheme:
In-Semester: **50** marks
End-Semester: **50** marks
Credits: **3**

Prerequisites: Operating Systems and Computer Networks

Course Objectives:
Familiarize students with
1. Distributed Systems and its ecosystem.
2. Basics of virtualization and its importance.
3. In-depth analysis of cloud computing capabilities.
4. Overview of cloud programming and services.

Course Outcomes:
Students should be able to
1. Recognize need of cloud based solutions.
2. Justify the importance of distributed systems.
3. Determine effective techniques to program cloud systems.
4. Evaluate current challenges and trade-offs in cloud computing.

Unit – I Introduction to Distributed Systems (07)
Scalable Computing over the Internet, Technologies for Network-Based Systems, System Models for Distributed and Cloud Computing, Software Environments for Distributed Systems and Clouds, Performance, Security, and Energy Efficiency

Unit – II Computer Clusters for Scalable Parallel Computing (07)
Clustering for Massive Parallelism, Computer Clusters and MPP Architectures, Design Principles of Computer Clusters, Cluster Job and Resource Management, Case Study: Top Supercomputer Systems

Unit – III Virtual Machines and Virtualization of Clusters and Data Centers (07)
Implementation Levels of Virtualization, Virtualization Structures/Tools and Mechanisms, Virtualization of CPU, Memory, and I/O Devices, Virtual Clusters and Resource Management, Virtualization for Data-Center Automation

Unit – IV Cloud Platform Architecture over Virtualized Data Centers (07)
Cloud Computing and Service Models, Data-Center Design and Interconnection Networks, Architectural Design of Compute and Storage Clouds, Public Cloud Platforms: GAE, AWS, and Azure, Inter-cloud Resource Management, Cloud Security and Trust Management

Unit – V Cloud Programming and Software Environments (07)

Features of Cloud and Grid Platforms, Parallel and Distributed Programming Paradigms, Programming Support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, Emerging Cloud Software Environments,

Unit – VI Grids, P2P, and the Future Internet (07)

Grid Architecture and Service Modeling, Grid Projects and Grid Systems Built, Peer-to-Peer Computing Systems, Cloud Trends in Supporting Ubiquitous Computing, Enabling Technologies for the Internet of Things

Text Books

1. Jack J. Dongarra, Kai Hwang, Geoffrey C. Fox, Distributed and Cloud Computing: From Parallel Processing to the Internet of Things, Elsevier, First Edition

Reference Books

1. Thomas Erl, Zaigham Mahmood and Ricardo Puttini, Cloud Computing: Concepts, Technology & Architecture, Pearson, First Edition
2. Rajkumar Buyya, Christian Vecchiola, S.Thamarai Selvi, Mastering Cloud Computing: Foundations and Applications Programming, McGraw Hill, First Edition
3. A. Srinivasan, J. Suresh, Cloud Computing: A practical approach for learning and implementation, Pearson, First Edition
4. Anthony T. Velte, Cloud Computing: Practical Approach, McGraw Hill, and First Edition
5. Ronald L. Krutz and Russell D. Vines, Cloud Security: A Comprehensive guide to Secure Cloud Computing, Wiley, First Edition

HS 4101 Green Computing

Teaching Scheme:
Lectures: **3** hrs/week
Tutorial: NIL

Examination Scheme:
In-Semester: **50** marks
End-Semester: **50** marks
Credits: **3**

Prerequisites: Basic Sciences

Course Objectives:

Familiarize students with

1. Knowledge of green computing practices to minimize negative impacts on the environment.
2. Principles of green computing.
3. Green Computing and how it can help improve environmental sustainability.
4. Green Computing in enterprises and its impact.

Course Outcomes:

Students should be able to

1. Relate to the socio cultural aspects of green computing.
2. Create awareness about green computing and promote green agenda in their working environments leading to green movement.
3. Apply green computing skills such as energy efficiency, IT assets disposal, carbon footprint estimation, reporting and development of green products.
4. Justify green initiatives while developing applications and services in enterprises.

Unit – I: Introduction to Green Computing (07)

Environmental Impacts of IT, Need of green computing, Green IT Standards, Enterprise Green IT Strategy, Hardware: Reuse, Recycle and Dispose, present scenario in industry, health issues relevance, Software: Energy-Saving Software Techniques

Unit – II: Software Development and Green Data Centers (07)

Sustainable Software, Software Sustainability Attributes, Software Sustainability Metrics, Sustainable Software Methodology, Data Centres and Associated Energy Challenges, Data Centre IT Infrastructure, Data Centre Facility Infrastructure: Implications for Energy Efficiency, IT Infrastructure Management, Green Data Centre Metrics

Unit – III: Green Data Storage and Networks (07)

Storage Media Power Characteristics, Energy Management Techniques for Hard Disks, System-Level Energy Management, Objectives of Green Network Protocols, Green Network Protocols and Standards

Unit – IV: Enterprise Green IT Strategy (07)

Approaching Green IT Strategies, Business Drivers of Green IT Strategy, Business Dimensions for Green IT Transformation, Multilevel Sustainable Information,

Sustainability Hierarchy Models, Product Level Information, Individual Level Information, Functional Level Information, Organizational Level Information, Regional/City Level Information

Unit – V: Green Computing Services and Roles (07)

Factors Driving the Development of Sustainable IT, Sustainable IT Services (SITS), Sustainable IT Roadmap, Organizational and Enterprise Greening, Information Systems in Greening Enterprises, Greening the Enterprise

Unit – VI: Regulating Green Computing (07)

The Regulatory Environment and IT Manufacturers, Nonregulatory Government Initiatives, Industry Associations and Standards Bodies, Green Building Standards, Green Data Centres, Social Movements

Text Books:

1. San Murugesan, G. R. Gangadharan: Harnessing Green IT, WILEY, 1st Edition-2013.

Reference Books:

1. Woody Leonhard, Katherrine Murray, "Green Home computing for dummies", August 2009, WILEY
2. Bhuvan Unhelkar, "Green IT Strategies and Applications-Using Environmental Intelligence", CRC Press, June 2011
3. Alin Gales, Michael Schaefer, Mike Ebbers, "Green Data Center: steps for the Journey", Shroff/IBM redbook, 2011.
4. Jason Harris, "Green Computing and Green IT-Best Practices on regulations & industry", Lulu.com, 2008
5. Carl Speshocky, "Empowering Green Initiatives with IT", John Wiley & Sons, 2010.
6. Wu Chun Feng (Editor), "Green computing: Large Scale energy efficiency", CRC Press, 2012.

OE 4101 Software Testing and Quality Assurance

Teaching Scheme:
Lectures: **3** hrs/week
Tutorial: NIL

Examination Scheme:
In-Semester: **50** marks
End-Semester: **50** marks
Credits: **3**

Prerequisites: Any programming language

Course Objectives:

Familiarize students with

1. Application of testing strategies in projects.
2. Test management strategies and tools for testing
3. Various quality assurance models

Course Outcomes:

Students should be able to

1. Analyze the project scenario and to select proper testing technique
2. Apply testing techniques to deliver a project
3. Choose quality assurance models for the project
4. Choose one of the latest testing tools suitable for the project

Unit – I: Software testing fundamentals **(07)**

Testing as a Process, Software testing principles, The tester's role in a software development organization, Origins of defects, Defect classes, Testing fundamentals, the defect repository and test design, Defect examples, Developer /Tester support for developing a defect repository. Process model to represent Different phases, Lifecycle models

Unit – II: Levels of testing **(07)**

Need for levels of testing, Unit testing, Integration testing, System Testing - Usability and Accessibility Testing, Configuration Testing, Compatibility Testing, Stress testing, Regression testing, Alpha, Beta and Acceptance testing.

Unit – III: Testing techniques **(07)**

Using White Box Approach to Test design - Static Testing, Structural Testing, Unit Functional Testing, Challenges in White box testing, Using Black Box Approaches to Test Case Design, Random Testing, Requirements based testing, Decision tables, State-based testing, Cause-effect graphing, Error guessing, Compatibility testing.

Unit – IV: Fundamentals of software quality assurance **(07)**

SQA basics, Components of the Software Quality Assurance System, software quality in

business context, planning for software quality assurance, product quality and process quality, software process models, 7 quality control Tools and Modern Tools.

Unit – V: Quality assurance models (07)

Models for Quality Assurance, ISO-9000 series, CMM, CMMI, Test Maturity Models, SPICE, Malcolm Baldrige Model- P-CMM, Clean-room software engineering ,Defect Injection and prevention, Inspections & Walkthroughs, Case Tools and their effect on Software Quality.

Unit – VI: Software test automation and current industry trends (07)

Software Test Automation, Skills needed for Automation, Scope of Automation, Design and Architecture for Automation, Requirements for a Test Tool, Challenges in Automation Tracking the Bug. Combining Manual and Automated Testing, Adoption of DevOps, Big Data Testing, IoT Testing, Introduction to testing tools.

Text Books:

1. Srinivasan Desikan, Gopalaswamy Ramesh,Software Testing: Principles and Practices,Pearson
2. Ilene Burnstein, Practical Software Testing, Springer International edition

Reference Books:

1. Paul C. Jorgensen, Software Testing: A Craftsman's Approach, Auerbach Publications
2. William Perry, Effective Methods of Software Testing, Wiley Publishing, Third Edition
3. Stephen Kan, Metrics and Models in Software Quality, Addison – Wesley, Second Edition
4. Watts S Humphrey, Managing the Software Process, Pearson Education Inc.

IT 4103 Software Architecture and Design Pattern Laboratory

Teaching Scheme:
Practical: **2** hours/week
Tutorial: NIL

Examination Scheme:
Oral: **50** marks
Credits: **1**

Prerequisites: Web Engineering Technology, Programming skill development laboratory

Course Objectives:

Familiarize students with

1. One client side programming Technology
2. One server side programming Technology
3. Developing a multiuser application

Course Outcomes:

Students should be able to

1. Apply appropriate technology to design the client side of the application
2. Apply appropriate technology to design the server side part of the application
3. Design the persistent layer classes their connection to database
4. Deploy and run the complete application

List of Assignments

1. Identify a system having three or four user expectations. Prepare its use case model
2. For the same system, prepare its analysis class model.
   Implement it using java language
3. For the same system, refine the analysis model and prepare the design class model.
   Implement it using java. Include appropriate applicable design patterns while designing the system.
4. Add view classes to your model and run the code handling appropriate events.
5. Design persistent layer classes and connect the business logic to database.
6. Deploy the application on server and ensure that it runs for various clients.
   Comment on the Quality attributes addressed in the system.

Text Books

1. Robert Sebasta, Programming the world wide web, Pearson Education, Edition 7, 2013
2. Deitel, Deitel and Nieto, Internet and World wide web how to program
   Pearson Education, Edition 5, 2013

Reference Books

1. Kogent Learning Solutions Inc Web Technologies Black Book 2009

IT 4104-PROJECT PHASE – I

Teaching Scheme:
Tutorial: **2** hrs/week
Practical:**14** hrs/week

Examination Scheme:
In semester: **100** marks
Oral: **50** marks
Credits: **9**

Course Objectives :
Familiarize students with:

1. The practical implementation of theoretical knowledge gained till date.
2. implementation of their ideas/real time industrial problem/ current application of Computer Science or Information Technology.

Course Outcomes :
At the end of this course the student should be able to :

1. Formulate a statement for the problem in Computer Science or Information Technology domain.
2. Prepare prototype for the identified problem.
3. Prepare System Specifications.
4. Work in team using ethical practices.

Following activities are expected to be completed in Project Phase-I:

1. Identification of Problem
2. Feasibility study
3. Formulation of Problem Statement
4. Abstract writing
5. Literature Survey
6. Project planning and maintaining log
7. High level System Design
8. Preparation of UML diagram using Tools.
9. Study of technology/platform
10. Technical Report writing
11. Start project based online course.

All this should be done with frequent meetings with internal and external guide.
The log has to be maintained.
Every project group has to give 2 Reviews in Semester-I
In Review-I, Point 1 to 4 should be completed. Demonstration and discussion with reviewers will be done.
In Review-II, Point 5 to 11 should be completed. Demonstration and discussion with reviewers will be done.