College of Computer, Science & Information Technology - Junagadh

AFFILIATED TO BHAKTA KAVI NARSINH MEHTA UNIVERSITY

CCSIT

✦ **Syllabus** (NEP–2020) ✦

B.Sc.(IT) (Honors)

[Bachelor of Science in Information Technology]

[Semester – V]

Academic Year : 2025 – 26

(Effective from June – 2025)

Credit Structure for Semester – 5

Sr. No.	Course Group (Major/Minor)	Course Paper Title	Credit	Ext. Marks (Theory)	Int. Marks (Theory)	Ext. Marks (Pract.)	Int. Marks (Pract.)	Total Marks
1	Major-11	**Advanced Java and J2EE (Theory)**	04	50	50	--	--	100
2	Major-12	**Digital Applications of IKS using J2EE**	04	--	--	50	50	100
3	Major-13	**Computer Networks and Communication Protocols**	04	50	50	--	--	100
4	Minor-4	**In House Project Development**	04			50	50	100
5	Minor-5	**Programming in Python**	04	50	50	--	--	100
6	SEC	**System Analysis and Design**	02	25	25	--	--	50
		Total Credits	22		**Total Marks**			550

General Instructions:

1. **For each 100 Marks Theory paper :**

   Internal Marks : 50 Marks

   University Theory Exam Marks : 50 Marks

   Duration for University Theory Exam : 2 Hrs.

   Five Questions in each paper.

2. **For each 50 Marks Theory paper :**

   Internal Marks : 25 Marks

   University Theory Exam Marks : 25 Marks

   Duration for University Theory Exam : 1 Hrs.

   Three Questions in each paper.

Paper Style: For Credit 4 and 50 Marks

Ques. No.	Particular	From Which Unit	Marks
1	Questions (Any Two Out Of Four)	1	10
2	Questions (Any Two Out Of Four)	2	10
3	Questions (Any Two Out Of Four)	3	10
4	Questions (Any Two Out Of Four)	4	10
5	Questions (Any Two Out Of Four)	From Each Unit	10
	Total Marks		**50**

Paper Style: For Credit 2 and 25 Marks

Ques. No.	Particular	From Which Unit	Marks
1	Questions (Any Two Out of Four)	1	10
2	Questions (Any Two Out of Four)	2	10
3	Questions (Any Two Out of Four)	From Each Unit	05
	Total Marks		**25**

B.Sc.(IT) SEMESTER – V
Major-11 : Advance JAVA & J2EE

Course Objectives:
- To understand different enterprise architecture styles like Two-tier, Three-tier, and N- tier architectures.
- To provide hands-on knowledge of Java database connectivity using JDBC.
- To build dynamic web applications using Servlets and JSP.
- To introduce key enterprise-level technologies like RMI, EJB, Hibernate, Spring, and Struts.
- To enable development of full-stack Java enterprise applications using MVC architecture.

Course Learning Outcomes: After completion of the course:
- Understand the fundamentals and architecture of the J2EE platform.
- Identify the components of J2EE and explain how they work together in enterprise applications.
- Explain the need for JDBC and its role in Java applications.
- Understand the architecture and life cycle of Servlet and JSP.
- Collaborate between servlets and JSP with JDBC for dynamic data processing.
- Understand and implement MVC (Model-View-Controller) architecture in web applications.
- Explain the need and benefits of using Hibernate as an ORM tool.
- Understand the architecture and core features of the Spring framework.
- Understand the flow of control in the Struts framework.

Course Contents

Unit	Contents	Hours
1	- Introduction to J2EE - Enterprise Architecture Styles: - Two-Tier Architecture - Three-Tier Architecture - N-Tier Architecture - J2EE Architecture - The J2EE Platform - Introduction to J2EE APIs (Servlet, JSP, EJB, JMS) - Introduction to Containers - Introduction of JDBC - JDBC Architecture - Data types in JDBC - Database Exception Handling - JDBC Introduction and Need for JDBC - Types of JDBC Drivers - JDBC API for Database Connectivity (java.sql package) - Statement, PreparedStatement, CallableStatement - ResultSetMetaData - DatabaseMetaData - Connecting with Databases (MySQL, Access, Oracle)	15
2	- Servlet Introduction - Architecture of a Servlet - Servlet API (Javax.servlet and Javax.servlet.http) - Servlet Life Cycle - Developing and Deploying Servlets - Handling Servlet Requests and Responses - Session Tracking Approaches (URL Rewriting, Hidden Form Fields, Cookies, Session API) - Servlet Collaboration - Servlet with JDBC - RMI overview - RMI architecture - Stub and Skeleton	15

3	• Introduction to JSP and JSP Basics
	• JSP vs. Servlet
	• JSP Architecture
	• Life cycle of JSP
	• JSP Elements: Directive Elements, Scripting Elements, Action
	• Elements
	   o Directives Elements (page, include)
	   o Scripting Elements (Declaration, scriptlet, expression)
	   o Action Elements (JSP:param, JSP:include, JSP:Forward, JSP:plugin)
	• JSP Implicit Objects
	• Including and Forwarding from JSP Pages
	• include Action
	• forward Action
	• Working with Session & Cookie in JSP
	• Error Handling and Exception Handling with JSP
	• JDBC with JSP
	• Introduction to JavaBean
	• JavaBean Properties
	• JavaBean Methods
		15

4	• Introduction to MVC • Implementation of MVC Architecture • Introduction to EJB • Benefits of EJB • Types of EJB: Session Beans, Entity Beans, Message-driven beans • Introduction to Hibernate • Features of hibernate • Exploring Hibernate Architecture • Hibernate Configuration file • Hibernate Mapping file • Basic Example of Hibernate • Annotation • Hibernate Inheritance • Inheritance Annotations • Introduction of Spring Framework • Spring Architecture • Spring & MVC • Understanding Struts Framework • Struts Flow of Control	**15**

Suggested Reading:
References Book:
1. The Complete Reference Java 2 - Herbert Schildt and Patrick Naughton
2. Java Server Programming For Professionals, Ivan Bayross, Sharanam Shah – Shroff Publication
3. Advanced Java Programming [ISBN: 978 - 93 - 81786 - 91 – 8] by Bharat & Company
4. Developing Java Servlets – Techmedia
5. JSP Beginner's Guide – Tata McGraw Hill by Gary Bolling, Bharathi Nataragan
6. Spring and Hibernate, K. Santosh Kumar, - Tata McGraw-Hill

Web site References:
1. https://www.w3schools.com/java
2. https://www.javatpoint.com/java-tutorial
3. https://www.tutorialspoint.com/java
4. https://www.guru99.com/java-tutorial.html

Major-12 : Digital Applications of IKS using J2EE

Objectives:
- To introduce the fundamentals and relevance of Indian Knowledge Systems (IKS) in the digital era.
- To explore how enterprise web technologies (J2EE) can support the documentation, dissemination, and preservation of IKS.
- To equip students with practical knowledge of building J2EE-based applications that reflect or promote traditional Indian knowledge (e.g. Ayurveda, Sanskrit and Vedas).
- To foster value-based and culturally contextual software development practices.
- To encourage innovation in applying modern web frameworks for real-world IKS- based systems.

Outcomes:
- Explain the core components of Indian Knowledge Systems and their modern-day significance.
- Design and implement web-based applications using J2EE (Servlets, JSP, JDBC, JavaBeans) to present IKS content.
- Structure and manage databases for storing traditional Indian cultural data using JDBC and Hibernate.
- Build interactive, multilingual, and culturally sensitive user interfaces for IKS platforms.
- Integrate frameworks like spring and Struts in creating scalable IKS-based applications.
- Demonstrate ethical considerations and cultural authenticity in digital projects related to Indian heritage.

Course Contents

Unit	Contents	Hours
1	**JDBC and Database Connectivity for IKS** • **Setup JDBC with MySQL for IKS Content** Create a database for Indian heritage content such as Ayurvedic herbs, Sanskrit terms, or Panchang data using JDBC and MySQL. • **Demonstration of JDBC Driver Types** Explore and test different JDBC driver types to understand which are best suited for connecting to heritage data storage platforms. • **Querying IKS Data with JDBC API** Use Statement, PreparedStatement, and CallableStatement to fetch and update data related to Vedic rituals, traditional texts, and Yoga poses. • **Metadata Handling for Cultural Databases** Use ResultSetMetaData and DatabaseMetaData to extract schema information and document the structure of IKS-based datasets. • **Exception Handling in Heritage Applications** Implement exception handling techniques to ensure robust and ethical handling of user input and retrieval of IKS content from databases.	**30**
2	**Servlets for Dynamic IKS Applications** • **Create a Panchang Viewer Using Servlet** Develop a web application that fetches and displays daily Panchang details (tithi, nakshatra, sunrise time) using Java Servlets. • **Servlet Collaboration for Multi-Module IKS Applications** Design a system where multiple servlets share data, such as linking anchatantra stories with related Sanskrit shlokas. • **Session Tracking for Personalized IKS Apps** Use session tracking techniques (cookies, session API, hidden fields) to personalize IKS apps such as favorite Ayurveda herbs or saved Yoga routines. • **Servlet with JDBC for Ayurvedic Remedies** Build a form-based system using Servlet and JDBC that provides remedies for common ailments based on traditional Ayurvedic knowledge. • **Remote Access to IKS Data using RMI** Implement a basic RMI-based application that allows remote access to an IKS dataset, such as a herb knowledge base or Sanskrit glossary.	**30**

3	**JSP and JavaBeans for IKS Web Interfaces**
	• **Sanskrit Shloka Display using JSP**
	   Create a web page that displays a Sanskrit Shloka or philosophical quote of the day using JSP and JDBC.
	• **Session and Cookie Management in IKS Sites**
	   Store and retrieve user preferences in an IKS learning portal (e.g., preferred Vedic topic or language setting) using cookies and sessions in JSP.
	• **Use of JSP Action Elements in Cultural Apps**
	   Apply jsp:include, jsp:forward, and jsp:param to modularize components in applications like Panchatantra Story Readers or Gurukul admission forms.
	• **Exception Handling in JSP for Jyotish Application**
	   Develop a Zodiac generator using JSP that includes proper error and exception handling when user input (date/time) is invalid.
	• **Integration of JavaBeans for Yoga Posture Viewer**
	   Create reusable JavaBeans to manage data related to Yoga asanas, including name, category, Sanskrit translation, and benefits.
			30

4	**MVC and Advanced Frameworks for IKS Projects**
	• **Mini Project using MVC Pattern**
	   Implement a full-stack mini project such as a Gurukul Student Registration System or Sanskrit Learning Portal using the MVC architectural pattern.
	• **Hibernate-Based Application for IKS Content**
	   Use Hibernate ORM to map and retrieve data like Yoga asanas, Vedic sutras, or Ayurveda formulations from the database.
	• **Struts Framework for Panchang Calendar Application**
	   Develop a Panchang calendar app using Struts MVC framework to demonstrate enterprise-level project structure.
	• **Spring MVC for Sanskrit Content Delivery**
	   Build an app to upload, store, and search Sanskrit verses or texts using the Spring MVC framework.
	• **Hibernate Annotations for Modeling Cultural Hierarchies**
	   Apply annotations in Hibernate to represent structured Indian knowledge hierarchies, such as Vedas → Samhitas → Mandalas.
			30

Suggested Reading:
References Book:
For J2EE and Web Technologies

1. The Complete Reference Java 2 - Herbert Schildt and Patrick Naughton
2. Java Server Programming for Professionals, Ivan Bayross, Sharanam Shah – Shroff Publication
3. Advanced Java Programming by Bharat & Company
4. Developing Java Servlets – Techmedia
5. JSP Beginner's Guide – Tata McGraw Hill by Gary Bolling, Bharathi Nataragan
6. Spring and Hibernate, K. Santosh Kumar, - Tata McGraw-Hill
7. "Head First Servlets and JSP" – Bryan Basham, Kathy Sierra, Bert Bates (O'Reilly)
8. "Beginning JSP, JSF and Tomcat: Java Web Development" – Giulio Zambon
9. "Java EE 7 Essentials" – Arun Gupta
10. "Spring in Action" – Craig Walls
11. "Mastering Hibernate" – Ramin Rad

For Indian Knowledge Systems (IKS):

1. "Indian Knowledge Systems: Concepts and Applications" – Kapil Kapoor
2. "The Argumentative Indian" – Amartya Sen
3. "The Cultural Heritage of India" – The Ramakrishna Mission Institute of Culture (multi-volume)
4. "Vedic Science and Technology" – Dr. A.V. Srinivasan
5. National Education Policy 2020 (NEP 2020) – Government of India
6. NITI Aayog Reports on Indian Knowledge Systems

Web site References:
For J2EE and Related Technologies

1. https://www.w3schools.com/servlet
2. https://www.javatpoint.com/jdbc-tutorial
3. https://www.javatpoint.com/jsp-tutorial
4. https://www.tutorialspoint.com/jsp
5. https://spring.io/
6. https://docs.oracle.com/javaee/ – Official Java EE Documentation
7. https://www.baeldung.com/ – Advanced articles on Spring, Hibernate, JSP, etc.
8. https://www.geeksforgeeks.org/ – Practice and examples for Java web technologies

For IKS:
1. https://iksindia.org – Official website of Indian Knowledge Systems Division, MoE, Govt. of India
2. https://epgp.inflibnet.ac.in – e-PG Pathshala (Modules on Sanskrit, Yoga, Ayurveda, etc.)
3. https://www.indianculture.gov.in – National Digital Repository of Indian Culture
4. https://vedicheritage.gov.in – Vedic Heritage Portal
5. https://ccrtindia.gov.in – Centre for Cultural Resources and Training

Major-13 : Computer Networks and Communication Protocols

Objectives:
- Understand fundamental concepts of networking including devices, protocols, and communication models.
- Identify and apply various physical components like cables, connectors, and switches in a networking setup.
- Configure and troubleshoot IP addressing and network connectivity issues.
- Set up and secure home and small office wireless networks.
- Use open-source tools for network monitoring and diagnostics.
- Develop practical skills through hands-on labs and mini-projects simulating real- world scenarios.

Outcomes:
- Explain networking terms, protocols, and standards relevant to small-scale networks.
- Demonstrate the ability to crimp and test RJ-45 cables and set up wired LAN connections.
- Configure IPv4 addresses (static and DHCP), identify common errors, and use tools like ping and ipconfig.
- Setup and secure a Wi-Fi network using routers, encryption, MAC filtering, and other configurations.
- Apply troubleshooting techniques to identify and resolve basic networking issues.
- Use open-source tools for network scanning and diagnostics.
- Implement subnetting and design small office/home networks

Course Contents

Unit	Contents	Hours
1	• Basics of Networking and Connections • What is networking? • Need and real-life examples • Network topologies: Star, Bus, Ring, Mesh • Wired and Wireless Connections: Overview and Demo • Practical: Setting up a basic wired connection between two PCs • Practical: Connecting a computer and mobile device to a Wi-Fi network • Difference between LAN and Internet • Introduction to network devices (router, switch, NIC) • Common Networking Protocols: TCP, UDP, HTTP, HTTPS, FTP, DHCP, DNS (basic overview) • OSI and TCP/IP Model (layer-wise overview with examples) • Bandwidth, latency, and throughput concepts • Network Standards and Bodies (IEEE, IETF) • Internet structure: ISPs, IXPs, DNS hierarchy	
2	• **Cables, Crimping and Physical Setup** • Types of cables: UTP, STP, Cat5e, Cat6 • Safe cable handling and color coding • RJ-45 pin configuration (T568A/T568B standards) • Practical: Crimping RJ-45 for Straight-through & Crossover cables • Practical: Testing cables using LAN cable tester • Patch panel and switch connections • Cable distance limitations and signal degradation • Structured cabling standards (TIA/EIA-568) • Network virtualization (VLANs / basics of cloud networking)	

3	• **IP Addressing, Configuration, and Troubleshooting**
	  • IP Address basics (IPv4) and private/public IPs
	  • Introduction to IPv6 and comparison with IPv4
	  • Subnetting basics (IPv4 only)
	  • DHCP vs Static IP
	  • Basics of MAC address and DNS functionality
	  • Practical: Assigning static and dynamic IP addresses
	  • Practical: Demonstrating DHCP and DNS resolution using command-line tools (ipconfig, nslookup)
	  • Introduction to ICMP protocol using ping and tracert
	  • Basic command-line networking on Linux (e.g., ifconfig, ping, netstat)
	  • Using open-source tools like Wireshark for packet capture (basic demo) & Angry IP Scanner for network scanning
	  • Introduction to Firewalls and Port Forwarding	

4	• **Router Configuration, Wireless Networking, and Open-Source Tools**
	  • Wi-Fi basics: Frequency bands, channels, SSID, encryption
	  • (WPA2/WPA3)
	  • Practical: Configuring a home Wi-Fi router (SSID, password, MAC filter)
	  • Guest network, parental control, and firmware update
	  • Open source: Using WiFi Analyzer (Android) for signal monitoring
	  • Connecting mobile and IoT devices to Wi-Fi
	  • Security threats: Wi-Fi sniffing, rogue APs, weak encryption
	  • Network security practices (firewall, MAC filter, strong passwords)
	  • Introduction to VPNs (Virtual Private Networks)
	  • QoS (Quality of Service) basics	

Suggested Practical:
- Setup a basic wired LAN connection between two PCs
- Assign Static and Dynamic IP addresses to computers
- Configure and test RJ-45 cables (Straight-through and Crossover)
- Test cables using a LAN cable tester
- Access router settings and configure SSID, password, MAC filter
- Use ping command to identify connectivity
- Simulate and resolve common network issues (IP conflict, DHCP failure)
- Perform network scanning using Angry IP Scanner or nmap
- Demonstrate port forwarding setup on a home router
- Set up a mobile hotspot and share internet with PC
- Connect and monitor an IoT device (e.g., smart bulb) to Wi-Fi
- Simulate network issues (e.g., unplug switch, change IP, disable DHCP) and troubleshoot

Capstone Practical Project
- **Design and implement a complete small home/office network**
  - including: Wired and wireless setup
  - Proper IP addressing
  - Device connectivity (PCs, mobile, IoT)
  - Security configurations (MAC filtering, passwords, firewall)
  - Network testing and troubleshooting of the setup

Mini Project:
- Set up a secure LAN + Wi-Fi network with 3–5 clients and test connectivity
- Perform subnetting using IP calculator tools and assign IPs accordingly
- Simulate a small home/office network with router, switch, and multiple clients

Recommended Books
1. "Computer Networking: A Top-Down Approach" – James F. Kurose and Keith W. Ross
2. "Data Communications and Networking" – Behrouz A. Forouzan
3. "Networking All-in-One For Dummies" – Doug Lowe
4. "CCNA 200-301 Official Cert Guide" – Wendell Odom (For practical network configuration knowledge)

Useful Websites & Online Resources
1. https://www.geeksforgeeks.org/computer-network-tutorials/ – Beginner to advanced topics
2. https://www.packetlife.net/ – Networking cheat sheets

3. https://wiki.wireshark.org/ – Official Wireshark usage documentation
4. https://nmap.org/ – Network scanning tool documentation
5. https://www.cloudflare.com/learning/ – Simple guides on DNS, firewalls, VPNs
6. https://howtogeek.com/ – Practical, real-life router and networking tutorials
7. https://ipcalc.co/ – Online IP subnet calculator

Minor-4 : In House Project Development

Objectives:
- Apply knowledge of programming, databases, and software development to real- world problems.
- Identify, design, and implement a project based on societal, commercial, or research-based needs.
- Use appropriate tools, platforms, and frameworks to build, test, and document a functional software solution.
- Demonstrate the ability to work independently or in teams and communicate effectively through reports and presentations.
- Develop problem-solving, critical thinking, and analytical abilities through end-to- end project development.

Outcomes:
- Identify a suitable project topic and define clear objectives.
- Plan, design, and implement a working prototype/application.
- Use software engineering practices for coding and documentation.
- Demonstrate effective teamwork, communication, and time management.
- Present and defend the developed project with technical clarity.

Course Contents

Module	Content	Hours
1	• Project Development (In House) With any of the Minor or Major Subject Basics.	120

Suggestive assessment scheme:

Component	Marks	Type
Internal Practical	25	Proposal, Flowcharts and Diagrams, attendance
	25	Code Implementation and Testing
External Practical(SEE)	50	Final report, viva, project presentation

Guidelines:
- Project must be developed in the computer laboratory of concern institute under the supervision of faculties of concern institute on any subject of previous semester or current semester.
- At the time of Project-Viva examination student must show all the Workouts, SDLC, Documentation, Program codes and project in running mode.
- The guide shall monitor progress of the student continuously. A student is required to present the progress of the Minor Project work during the semester as per the schedule provided by the Guide.
- Student has to submit a hard copy of Minor Project Report, workbook and a copy of Minor Project as per guide suggestion.
- Project submission should follow the guide's schedule, with a mandatory live demonstration during the viva, supported by a multimedia presentation.

Minor-5 : Programming in Python

Objectives:
- Understand the history and features that make Python popular.
- Learn how to install and set up Python in different environments.
- Understand Python's syntax, indentation rules, and naming conventions.
- Explore how variables, literals, and operators work in Python.
- Write basic input/output operations using input() and print().
- Use conditional statements and loops for flow control.
- Execute Python programs using different tools (Command Line, IDLE, etc.).
- Define and use custom functions with various parameter types.
- Work with Python modules for code reuse and organization.

Outcomes:
- Install Python and set up a development environment.
- Write and execute basic Python scripts.
- Understand and apply proper syntax, naming, and indentation.
- Use variables, literals, and operators correctly.
- Write programs that interact with users through input and output.
- Control program flow using if, else, for, while, break, and continue.
- Develop and organize reusable code using functions and modules.
- Execute Python scripts in multiple ways (CLI, IDLE, editor).

Course Content

Unit	Contents	Hours
1	- Python History, Features & Installation - Introduction to Python IDLE - Basic Syntax, Indentation, Reserved Words, Naming Conventions - Python Variables - Literals, Operators and Comments - Input and Output, print() function - Python Conditional and looping Statements - Python Functions - Python Modules	15
2	- Python Files I/O - Python Data Types - Python Strings - Python Lists - Python Tuples - Python Sets - Python Dictionary - Special Symbols and Characters - Regexes and Python - Text Processing: - Comma Separated Values (CSV files) - JavaScript Object Notation (JSON)	10
3	- Handling Exceptions - Exceptions as a control flow mechanism - Assertions, Abstract Data Types and Classes - Inheritance, Encapsulation and Information Hiding - Search Algorithms (Linear, Binary) - Sorting Algorithms (Selection, Bubble, Insertion, Shell, Quick)	10
4	- Python Tkinter - Tkinter Introduction - Tkinter Button, Canvas, Checkbutton, Entry, Frame, Label - Tkinter Listbox, Menubutton, Menu, Message, Radiobutton - Tkinter Scale, Scrollbar, Text, Toplevel,Spinbox - Tkinter PanedWindow, LabelFrame, MessageBox - Basic Python Library **NumPy** : array(), mean(), max(), reshape(), arange() **Pandas** : read_csv(), head(), describe(), dropna(), sort_values() **OpenCV – Image Functions** : imread(), imshow(), imwrite(), resize(), cvtColor() **OpenCV – Video/Webcam Functions** : VideoCapture(), read(), waitKey(), destroyAllWindows(), release() **OpenCV – Drawing Functions :** line() , rectangle() , circle() , putText() , polylines() **Matplotlib (plt) :** plot() , bar() , scatter() , xlabel() , show() **Seaborn (sns) :** histplot() , boxplot() , countplot() , heatmap() , scatterplot() **Plotly Express (px) :** bar() , line() , scatter() , pie() , show()	10
	PRACTICAL	30

References Book:
1. Python: The Complete Reference
2. Python Programming Fundamentals- A Beginner's Handbook
3. Core Python Programming
4. Python GUI Programming with Tkinter by Alan D. Moore
5. Python for Data Analysis by Wes McKinney (also covers Pandas)
6. Matplotlib for Python Developers by Sandro Tosi
7. Python Concurrency with asyncio by Matthew Fowler

Web site References:
1. https://www.tutorialspoint.com/python
2. https://www.w3schools.com/python
3. https://docs.python.org/3/tutorial
4. https://www.guru99.com/python-tutorials.html
5. https://www.programiz.com/python-programming/tutorial
6. https://www.learnpython.org
7. https://realpython.com/tutorials/python

SEC : System Analysis and Design

Course Objectives:
- Understand the phases and significance of the Software Development Life Cycle (SDLC).
- Grasp foundational concepts of Software Engineering and its importance in structured software development.
- Differentiate between Quality Assurance (QA) and Quality Control (QC) in the software development process.
- Understand the principles and processes of software testing, including various testing levels and types.
- Identify and apply different software testing techniques, including black box, white box, and grey box testing.
- Explain and perform non-functional testing, including performance, load, stress, usability, and security testing.
- Compare and evaluate various SDLC models such as Waterfall, V-Model, Spiral, Prototyping, etc.
- Interpret and create software engineering diagrams like DFDs, E-R diagrams, UML diagrams (Class, Use-case).
- Utilize tools like Microsoft Visio or Draw.io for software designing and document

Course Outcomes:
- Describe the key phases of SDLC (Requirement, Design, Developmen testing, deployment, maintenance).
- Compare different SDLC models and choose the appropriate model based on project requirements.
- Understand the fundamental concepts of system analysis and design
- Analyze and model system requirements using DFDs and ERDs
- Design and document a basic information system
- Use standard tools for system modeling and design (e.g., Draw.io, Lucid chart, visio)
- Demonstrate project understanding through documentation and presentation

Course Content

Unit	Contents	Hours
1	- Introduction to System - Types of Systems - Role and Responsibilities of System Analyst - SDLC - Phases: Planning, Analysis, Design, Implementation, Maintenance - Fact – finding techniques/ Requirement Analysis - Interview, Questionnaire, Record review and observation - Introduction to Quality assurance (QA) and Quality Control (QC) - Software Faults and Failures - Introduction to software Testing - Techniques of software Testing Testing Tools	15
2	- Software Development Life Cycle Models - Open-source Software, Freeware, Shareware - Timeline chart - Data Flow Analysis Tool - Data Dictionary - Types of Data Dictionary - DFD (Data Flow Diagram), E-R Diagram, Use-Case - Diagram (Using Visio, draw.io,etc)	15

Suggested Reading:
References Book:
1. Analysis & Design of Information System - James A. Sen.
2. Pankaj Jalote, "Software Engineering – A Precise Approach", Wiley India
3. UML Distilled by Martin Fowler, Pearson Edition, 3rd Edition
4. Fundamentals of Software Engineering – RajibMall (PHP)
5. Software Engineering – A Practitioner's Approach – Pressman
6. UML – A Beginner's Guide –Jasson Roff – TMH
7. Roger Pressman, "Software Engineering"

Web site References:
1. http://en.wikipedia.org/wiki/Software_testing
2. http://www.onestoptesting.com/
3. http://www.opensourcetesting.org/functional.php
4. https://www.geeksforgeeks.org/

Courses Offered

➢ B.Sc. – Bachelor of Science

➢ B.Sc.(IT) – Bachelor of Science in Information Tech.

➢ B.C.A. – Bachelor of Computer Application

➢ D.M.L.T. – Diploma in Medical Laboratory Technology

➢ M.Sc.(IT) – Master of Science in Information Technology

➢ M.Sc.(Micro.) – Master of Science in Microbiology

➢ M.Sc.(Chem.) – Master of Science in Chemistry

<div style="border:1px solid">

◄ ADDRESS : C.C.S.I.T. - JUNAGADH ►

Green City, Bypass Road, Nr. Chobari Railway Crossing, Junagadh.
Website : http://ccsit.edu.in, Email : ccsitjunagadh@gmail.com
Phone : 92280 06940, 79906 61530

</div>