

# College of Computer, Science & Information Technology - Junagadh

AFFILIATED TO BHAKTA KAVI NARSINH MEHTA UNIVERSITY



♦ Syllabus (NEP-2020) ♦

## **B.C.A. (Honors)**

[ Bachelor of Computer Application ]

[ Semester – V ]

Academic Year : 2025 – 26

( Effective from June – 2025 )



◀ **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](mailto:ccsitjunagadh@gmail.com)

Phone : 92280 06940, 79906 61530

### 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	Web Development Using ASP.NET	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.C.A.(Honors) 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	<ul style="list-style-type: none"><li>• Introduction to J2EE</li><li>• Enterprise Architecture Styles:<ul style="list-style-type: none"><li>○ Two-Tier Architecture</li><li>○ Three-Tier Architecture</li><li>○ N-Tier Architecture</li></ul></li><li>• J2EE Architecture</li><li>• The J2EE Platform</li><li>• Introduction to J2EE APIs (Servlet, JSP, EJB, JMS)</li><li>• Introduction to Containers</li><li>• Introduction of JDBC</li><li>• JDBC Architecture</li><li>• Data types in JDBC</li><li>• Database Exception Handling</li><li>• JDBC Introduction and Need for JDBC</li><li>• Types of JDBC Drivers</li><li>• JDBC API for Database Connectivity (java.sql package)</li><li>• Statement, PreparedStatement, CallableStatement</li><li>• ResultSetMetaData</li><li>• DatabaseMetaData</li><li>• Connecting with Databases (MySQL, Access, Oracle)</li></ul>	15

2	<ul style="list-style-type: none"> <li>Servlet Introduction</li> <li>Architecture of a Servlet</li> <li>Servlet API (Javax.servlet and Javax.servlet.http)</li> <li>Servlet Life Cycle</li> <li>Developing and Deploying Servlets</li> <li>Handling Servlet Requests and Responses</li> <li>Session Tracking Approaches (URL Rewriting, Hidden Form Fields, Cookies, Session API)</li> <li>Servlet Collaboration</li> <li>Servlet with JDBC</li> <li>RMI overview</li> <li>RMI architecture</li> <li>Stub and Skeleton</li> </ul>	15
3	<ul style="list-style-type: none"> <li>Introduction to JSP and JSP Basics</li> <li>JSP vs. Servlet</li> <li>JSP Architecture</li> <li>Life cycle of JSP</li> <li>JSP Elements: Directive Elements, Scripting Elements, Action Elements <ul style="list-style-type: none"> <li>Directives Elements (page, include)</li> <li>Scripting Elements (Declaration, scriptlet, expression)</li> <li>Action Elements (JSP:param, JSP:include, JSP:Forward, JSP:plugin)</li> </ul> </li> <li>JSP Implicit Objects</li> <li>Including and Forwarding from JSP Pages</li> <li>include Action</li> <li>forward Action</li> <li>Working with Session &amp; Cookie in JSP</li> <li>Error Handling and Exception Handling with JSP</li> <li>JDBC with JSP</li> <li>Introduction to JavaBean</li> <li>JavaBean Properties</li> <li>JavaBean Methods</li> </ul>	15
4	<ul style="list-style-type: none"> <li>Introduction to MVC</li> <li>Implementation of MVC Architecture</li> <li>Introduction to EJB</li> <li>Benefits of EJB</li> <li>Types of EJB: Session Beans, Entity Beans, Message-driven beans</li> <li>Introduction to Hibernate</li> <li>Features of hibernate</li> <li>Exploring Hibernate Architecture</li> <li>Hibernate Configuration file</li> <li>Hibernate Mapping file</li> <li>Basic Example of Hibernate</li> <li>Annotation</li> <li>Hibernate Inheritance</li> <li>Inheritance Annotations</li> <li>Introduction of Spring Framework</li> <li>Spring Architecture</li> <li>Spring &amp; MVC</li> <li>Understanding Struts Framework</li> <li>Struts Flow of Control</li> </ul>	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	<b>JDBC and Database Connectivity for IKS</b> <ul style="list-style-type: none"><li>• <b>Setup JDBC with MySQL for IKS Content</b> Create a database for Indian heritage content such as Ayurvedic herbs, Sanskrit terms, or Panchang data using JDBC and MySQL.</li><li>• <b>Demonstration of JDBC Driver Types</b> Explore and test different JDBC driver types to understand which are best suited for connecting to heritage data storage platforms.</li><li>• <b>Querying IKS Data with JDBC API</b> Use Statement, PreparedStatement, and CallableStatement to fetch and update data related to Vedic rituals, traditional texts, and Yoga poses.</li><li>• <b>Metadata Handling for Cultural Databases</b> Use ResultSetMetaData and DatabaseMetaData to extract schema information and document the structure of IKS-based datasets.</li><li>• <b>Exception Handling in Heritage Applications</b> Implement exception handling techniques to ensure robust and ethical handling of user input and retrieval of IKS content from databases.</li></ul>	30

2	<b>Servlets for Dynamic IKS Applications</b> <ul style="list-style-type: none"> <li><b>Create a Panchang Viewer Using Servlet</b> Develop a web application that fetches and displays daily Panchang details (tithi, nakshatra, sunrise time) using Java Servlets.</li> <li><b>Servlet Collaboration for Multi-Module IKS Applications</b> Design a system where multiple servlets share data, such as linking anchatantra stories with related Sanskrit shlokas.</li> <li><b>Session Tracking for Personalized IKS Apps</b> Use session tracking techniques (cookies, session API, hidden fields) to personalize IKS apps such as favorite Ayurveda herbs or saved Yoga routines.</li> <li><b>Servlet with JDBC for Ayurvedic Remedies</b> Build a form-based system using Servlet and JDBC that provides remedies for common ailments based on traditional Ayurvedic knowledge.</li> <li><b>Remote Access to IKS Data using RMI</b> Implement a basic RMI-based application that allows remote access to an IKS dataset, such as a herb knowledge base or Sanskrit glossary.</li> </ul>	30
3	<b>JSP and JavaBeans for IKS Web Interfaces</b> <ul style="list-style-type: none"> <li><b>Sanskrit Shloka Display using JSP</b> Create a web page that displays a Sanskrit Shloka or philosophical quote of the day using JSP and JDBC.</li> <li><b>Session and Cookie Management in IKS Sites</b> Store and retrieve user preferences in an IKS learning portal (e.g., preferred Vedic topic or language setting) using cookies and sessions in JSP.</li> <li><b>Use of JSP Action Elements in Cultural Apps</b> Apply jsp:include, jsp:forward, and jsp:param to modularize components in applications like Panchatantra Story Readers or Gurukul admission forms.</li> <li><b>Exception Handling in JSP for Jyotish Application</b> Develop a Zodiac generator using JSP that includes proper error and exception handling when user input (date/time) is invalid.</li> <li><b>Integration of JavaBeans for Yoga Posture Viewer</b> Create reusable JavaBeans to manage data related to Yoga asanas, including name, category, Sanskrit translation, and benefits.</li> </ul>	30
4	<b>MVC and Advanced Frameworks for IKS Projects</b> <ul style="list-style-type: none"> <li><b>Mini Project using MVC Pattern</b> Implement a full-stack mini project such as a Gurukul Student Registration System or Sanskrit Learning Portal using the MVC architectural pattern.</li> <li><b>Hibernate-Based Application for IKS Content</b> Use Hibernate ORM to map and retrieve data like Yoga asanas, Vedic sutras, or Ayurveda formulations from the database.</li> <li><b>Struts Framework for Panchang Calendar Application</b> Develop a Panchang calendar app using Struts MVC framework to demonstrate enterprise-level project structure.</li> <li><b>Spring MVC for Sanskrit Content Delivery</b> Build an app to upload, store, and search Sanskrit verses or texts using the Spring MVC framework.</li> <li><b>Hibernate Annotations for Modeling Cultural Hierarchies</b> Apply annotations in Hibernate to represent structured Indian knowledge hierarchies, such as Vedas → Samhitas → Mandalas.</li> </ul>	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 Zamboni
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 : Web Development Using ASP.NET**

#### Objectives:

- Understand the ASP.NET framework architecture and its components, including server-side controls, page lifecycle, and validation techniques.
- Gain hands-on knowledge in designing web forms using standard and advanced controls in ASP.NET.
- Implement client-side and server-side state management techniques for maintaining user session data.
- Utilize ADO.NET for database connectivity, perform CRUD operations, and bind data to various controls using SQL data sources.
- Design structured web applications using Master Pages, Themes, CSS3, and Bootstrap for responsive UI/UX.
- Apply techniques for web caching, including output and data caching for performance optimization.
- Integrate XML datasets and Web Services for data handling and interoperability.
- Understand and configure web.config settings, including security (authentication & authorization), error handling, and deployment best practices.

#### Outcomes:

- Describe the ASP.NET framework and identify the role of key server controls and life cycle stages.
- Design and develop interactive, data-driven web applications using ASP.NET Web Forms.
- Implement client-side and server-side validation to enhance form input accuracy and usability.
- Demonstrate the use of AJAX and state management to improve the performance and user experience of web applications.
- Connect to databases using ADO.NET and perform operations with both connected and disconnected architecture.
- Develop master pages and responsive web interfaces using CSS3 and Bootstrap to maintain a uniform layout.
- Apply caching techniques to enhance web application speed and efficiency.

### **Course Contents**

Unit	Contents	Hours
1	<ul style="list-style-type: none"><li>• Overview of Asp.NET Framework</li><li>• Client Server Architecture</li><li>• ASP.NET Life Cycle</li><li>• Types of Files in Asp.NET</li><li>• Types of controls in Asp.NET</li><li>• Page Architecture</li><li>• Introduction to standard Controls (Buttons, Textbox, Checkbox, Label, Panel, Listbox, Dropdownlist, FileUpload, AdRotator, CheckBoxLayout, RadioButtonList, ImageMap, Wizard, Calendar, File Upload Control)</li><li>• What is Validation?</li><li>• Client Side Validation</li><li>• Server Side Validation</li><li>• Types (RequiredField Validator, Range Validator, CompareField Validator, RegularExpression Validator, Custom Validator, ValidationSummary Control)</li></ul>	10



2	<ul style="list-style-type: none"> <li>• Setting up Ajax</li> <li>• ASP.NET Ajax Control Toolkit With Basic Control</li> <li>• What is State?</li> <li>• Why is it Required in Asp.Net?</li> <li>• Client Side State Management</li> <li>• Server Side State Management</li> <li>• Various State Management Techniques <ul style="list-style-type: none"> <li>○ View State,</li> <li>○ Query String,</li> <li>○ Cookie,</li> <li>○ Session State,</li> <li>○ Application State</li> </ul> </li> <li>• What is Master Page?</li> <li>• Requirement Of a Master Page in an Asp.NET application</li> <li>• Overview of CSS3</li> <li>• Overview of Bootstrap</li> <li>• Designing Website with Master Page, Theme and CSS Overview</li> </ul>	10
3	<ul style="list-style-type: none"> <li>• Architecture of ADO.NET</li> <li>• Connected Architecture</li> <li>• DisConnected Architecture</li> <li>• ADO.NET Classes (Connection, Command, DataReader, DataAdapter, DataSet, DataColumn, DataRow, DataConstraints, DataView etc.)</li> <li>• The Gridview Control, The Repeater Control</li> <li>• Binding Data to DataBound Controls,</li> <li>• Displaying Data in a webpage using SQLDataSource Control</li> <li>• DataBinding Expressions</li> </ul>	15
4	<ul style="list-style-type: none"> <li>• Page Output Caching</li> <li>• Partial Page Caching, Absolute Cache Expiration</li> <li>• Sliding Cache Expiration</li> <li>• Data Caching.</li> <li>• Reading Datasets From XML</li> <li>• Writing DataSets With XML</li> <li>• WebServices (Introduction, HTTP, SOAP, UDDI,XML, Creating a Web Service)</li> <li>• Introduction To Web.Config</li> <li>• Common Configuration Sections</li> <li>• AppSettings</li> <li>• Tracing</li> <li>• Custom Errors</li> <li>• Authentication And Authorization</li> <li>• Deployment of Application in web server</li> </ul>	10
	<b>PRACTICAL</b>	<b>30</b>

#### References Book:

1. Asp.Net – Unleashed
2. Asp.Net – Wrox Publication
3. Programming With ASP.NET [ISBN: 978 - 81 - 909634 - 7 - 3] by Bharat & Company
4. Beginning.ASP.NET.3.5.in.C.Sharp.2008.From.Novice.to.Professional – Apress

#### Web site References:

1. <https://www.tutorialspoint.com/asp.net/>
2. <https://www.guru99.com/asp-net-tutorial.html>
3. <https://www.javatpoint.com/asp-net-tutorial>
4. <https://www.w3schools.com/asp/>
5. <https://docs.microsoft.com/en-us/aspnet/tutorials>
6. <https://www.tutorialsteacher.com/core>
7. <https://www.tutorialsteacher.com/mvc/asp.net-mvc-tutorials>
8. <http://www.pragimtech.com/free-aspnet-video-tutorial.aspx>
9. <https://www.ajaxcontroltoolkit.net/>



## **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	<ul style="list-style-type: none"> <li>• Python History, Features &amp; Installation</li> <li>• Introduction to Python IDLE</li> <li>• Basic Syntax, Indentation, Reserved Words, Naming Conventions</li> <li>• Python Variables</li> <li>• Literals, Operators and Comments</li> <li>• Input and Output, print() function</li> <li>• Python Conditional and looping Statements</li> <li>• Python Functions</li> <li>• Python Modules</li> </ul>	15
2	<ul style="list-style-type: none"> <li>• Python Files I/O</li> <li>• Python Data Types                             <ul style="list-style-type: none"> <li>○ Python Strings</li> <li>○ Python Lists</li> <li>○ Python Tuples</li> <li>○ Python Sets</li> <li>○ Python Dictionary</li> </ul> </li> <li>• Special Symbols and Characters</li> <li>• Regexes and Python</li> <li>• Text Processing:                             <ul style="list-style-type: none"> <li>○ Comma Separated Values (CSV files)</li> <li>○ JavaScript Object Notation (JSON)</li> </ul> </li> </ul>	15
3	<ul style="list-style-type: none"> <li>• Handling Exceptions</li> <li>• Exceptions as a control flow mechanism</li> <li>• Assertions, Abstract Data Types and Classes</li> <li>• Inheritance, Encapsulation and Information Hiding</li> <li>• Search Algorithms (Linear, Binary)</li> <li>• Sorting Algorithms (Selection, Bubble, Insertion, Shell, Quick)</li> </ul>	15
4	<ul style="list-style-type: none"> <li>• Python Tkinter                             <ul style="list-style-type: none"> <li>○ Tkinter Introduction</li> <li>○ Tkinter Button, Canvas, Checkbutton, Entry, Frame, Label</li> <li>○ Tkinter Listbox, Menubutton, Menu, Message, Radiobutton</li> <li>○ Tkinter Scale, Scrollbar, Text, Toplevel, Spinbox</li> <li>○ Tkinter PanedWindow, LabelFrame, MessageBox</li> </ul> </li> <li>• Basic Python Library                             <ul style="list-style-type: none"> <li>○ <b>NumPy</b> : array(), mean(), max(), reshape(), arange()</li> <li>○ <b>Pandas</b> : read_csv(), head(), describe(), dropna(), sort_values()</li> <li>○ <b>OpenCV – Image Functions</b> : imread(), imshow(), imwrite(), resize(), cvtColor()</li> <li>○ <b>OpenCV – Video/Webcam Functions</b> : VideoCapture(), read(), waitKey(), destroyAllWindows(), release()</li> <li>○ <b>OpenCV – Drawing Functions</b> : line(), rectangle(), circle(), putText(), polylines()</li> <li>○ <b>Matplotlib (plt)</b> : plot(), bar(), scatter(), xlabel(), show()</li> <li>○ <b>Seaborn (sns)</b> : histplot(), boxplot(), countplot(), heatmap(), scatterplot()</li> <li>○ <b>Plotly Express (px)</b> : bar(), line(), scatter(), pie(), show()</li> </ul> </li> </ul>	15

### 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, Development 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
<b>1</b>	<ul style="list-style-type: none"> <li>• Introduction to System               <ul style="list-style-type: none"> <li>• Types of Systems</li> <li>• Role and Responsibilities of System Analyst</li> </ul> </li> <li>• SDLC               <ul style="list-style-type: none"> <li>• Phases: Planning, Analysis, Design, Implementation, Maintenance</li> </ul> </li> <li>• Fact – finding techniques/ Requirement Analysis               <ul style="list-style-type: none"> <li>• Interview, Questionnaire, Record review and observation</li> <li>• Introduction to Quality assurance (QA) and Quality Control (QC)</li> </ul> </li> <li>• Software Faults and Failures</li> <li>• Introduction to software Testing</li> <li>• Techniques of software Testing Testing Tools</li> </ul>	<b>15</b>
<b>2</b>	<ul style="list-style-type: none"> <li>• Software Development Life Cycle Models</li> <li>• Open-source Software, Freeware, Shareware</li> <li>• Timeline chart</li> <li>• Data Flow Analysis Tool               <ul style="list-style-type: none"> <li>• Data Dictionary</li> <li>• Types of Data Dictionary</li> </ul> </li> <li>• DFD (Data Flow Diagram), E-R Diagram, Use-Case Diagram (Using Visio, draw.io, etc)</li> </ul>	<b>15</b>

**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 – Rajib Mall (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](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

**◀ 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](mailto:ccsitjunagadh@gmail.com)

Phone : 92280 06940, 79906 61530