

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>BCA SEM 5</b>						
<b>Sr. No</b>	<b>Type Of Course</b>	<b>Subject</b>	<b>Credit</b>	<b>CCE</b>	<b>SEE</b>	<b>Total</b>
1	Major-11	<b>CS –29:</b> Advance Java Programming(J2EE)	4	50	50	100
2	Major-12	<b>CS –30:</b> Programming in Python	4	50	50	100
3	Major-13	<b>CS –31:</b> Cyber Security	4	50	50	100
4	Minor-04	<b>CS –32:</b> Mini Project	4	50	50	100
5	Minor-05	<b>CS –33:</b> Practical Based on CS-29 and CS-30	4	50	50	100
6	SEC-05	<b>CS –34:</b> Introduction to AI	2	25	25	50

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>CS-29: Advance Java Programming (J2EE)</b>		
<p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Gain a deep understanding of the principles of J2EE architecture, including servlets, JSP.</li> <li>• Proficiency in frameworks and technologies like spring framework, hibernate, spring boot.</li> <li>• Learn about the Model-View-Controller (MVC) design pattern and its application in J2EE development.</li> </ul> <p><b>Prerequisites:</b></p> <ul style="list-style-type: none"> <li>• Core Java Knowledge</li> </ul>		
Unit No.	Topic	Detail
<b>1</b>	Introduction to J2EE and JDBC	<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>• Enterprise Architecture</li> <li>• The J2EE Platform</li> <li>• Introduction to J2EE APIs (Servlet, JSP, EJB, JMS, JavaMail, JSF, JNDI)</li> <li>• Introduction to Containers</li> <li>• Tomcat as a Web Container</li> </ul>
		<ul style="list-style-type: none"> <li>• JDBC Architecture,</li> <li>• Types of JDBC Drivers,</li> <li>• Introduction to major JDBC Classes and Interface,</li> <li>• Creating simple JDBC Application,</li> <li>• Types of Statement (Statement Interface, PreparedStatement, CallableStatement),</li> <li>• Creating CRUD Application</li> </ul>
<b>2</b>	Servlet	<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>• Servlet Configuration with Deployment Descriptor</li> <li>• Developing and Deploying Servlets</li> <li>• Handling Servlet Requests and Responses</li> <li>• Reading Initialization Parameters</li> <li>• Session Tracking Approaches (URL Rewriting, Hidden Form Fields, Cookies, Session API)</li> </ul>

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>3</b>	JSP	<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: <ul style="list-style-type: none"> <li>▪ Directives Elements (page, include, taglib)</li> <li>▪ Scripting Elements (Declaration, scriptlet, expression)</li> <li>▪ Action Elements (jsp:param, jsp:include, jsp:Forward, jsp:plugin, jsp:useBean, jsp:setAttribute, jsp:getAttribute)</li> </ul> </li> <li>● JSP Implicit Objects (request, response, out, session, application, pagecontext)</li> <li>● JSP Scope</li> <li>● Including and Forwarding from JSP Pages <ul style="list-style-type: none"> <li>▪ include Action</li> <li>▪ forward Action</li> </ul> </li> <li>● Working with Session &amp; Cookie in JSP</li> <li>● Error Handling and Exception Handling with JSP</li> <li>● JSP EL (Expression Language), JSP Standard Tag Libraries (JSTL)</li> </ul>
<b>4</b>	EJB, Introduction of MVC Architecture, Hibernate	<ul style="list-style-type: none"> <li>● Introduction to EJB</li> <li>● Types of EJB</li> </ul> <hr/> <ul style="list-style-type: none"> <li>● Introduction to MVC</li> <li>● Implementation of MVC Architecture</li> </ul> <hr/> <ul style="list-style-type: none"> <li>● Introduction to Hibernate</li> <li>● Features of Hibernate</li> <li>● Exploring Hibernate Architecture</li> <li>● Object Relation Mapping (ORM) with Hibernate</li> <li>● Hibernate Configuration file</li> <li>● Hibernate Mapping file</li> <li>● Hibernate Annotation</li> <li>● Hibernate Query Language (HQL)</li> <li>● Hibernate Sessions</li> </ul>
<b>5</b>	Introduction to Spring Framework & Spring Boot	<p><b>Introduction to Spring Framework</b></p> <ul style="list-style-type: none"> <li>● What is Spring Framework?</li> <li>● Importance and Benefits of Spring</li> <li>● Spring Architecture Overview (Core Container, Modules)</li> </ul> <p><b>Core Concepts of Spring</b></p> <ul style="list-style-type: none"> <li>● Dependency Injection (DI) and Inversion of Control (IoC)</li> <li>● Bean Lifecycle in Spring</li> <li>● Spring ApplicationContext and BeanFactory</li> </ul> <p><b>Spring MVC Basics</b></p> <ul style="list-style-type: none"> <li>● Overview of MVC Pattern</li> <li>● Spring MVC Architecture and Request Flow</li> <li>● Controllers, Models, Views</li> <li>● Simple Spring MVC Application Example</li> </ul>

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

		<p><b>Aspect-Oriented Programming (AOP) in Spring</b></p> <ul style="list-style-type: none"> <li>• Introduction to AOP</li> <li>• Concepts: Advice, Join Point, Pointcut, Aspect</li> <li>• Simple Logging Aspect Example</li> </ul> <p><b>Spring Boot Fundamentals</b></p> <ul style="list-style-type: none"> <li>• What is Spring Boot?</li> <li>• Features and Advantages of Spring Boot</li> <li>• Architecture of Spring Boot</li> <li>• Auto-Configuration and Starter Dependencies</li> <li>• Embedded Servers (Tomcat, Jetty)</li> <li>• Important Spring Boot Annotations (@SpringBootApplication, @RestController, @RequestMapping)</li> </ul> <p><b>Spring Boot Data Access</b></p> <ul style="list-style-type: none"> <li>• Introduction to Spring Boot JDBC</li> <li>• Using JdbcTemplate for database operations</li> <li>• Overview of Spring Boot Data JPA</li> <li>• Basic CRUD operations with Spring Boot Data JPA and Hibernate</li> </ul> <p><b>Exception Handling in Spring Boot MVC</b></p> <ul style="list-style-type: none"> <li>• Handling exceptions using @ExceptionHandler</li> <li>• Using @ControllerAdvice for global exception handling</li> </ul>
--	--	---

**Reference Books:**

- (1) Java Complete Reference 11th Edition - Herbert Schildt, Oracle Press
- (2) Java Server Programming For Professionals, Ivan Bayross, Sharanam Shah – Shroff publication
- (3) Developing Java Servlets – Techmedia
- (4) JSP Beginner’s Guide – Tata McGraw Hill by Gary Bolling, Bharathi Nataragan
- (5) Spring and Hibernate, K. Santosh Kumar, - Tata McGraw-Hill
- (6) Hibernate Made Easy: Simplified Data Persistence with Hibernate and JPA (Java Persistence API) Annotations by Cameron Wallace McKenzie, Kerri Sheehan
- (7) Spring Framework: A Step by Step Approach for Learning Spring Framework – Create Space Independent Publishing Platform
- (8) Beginning Hibernate Second Edition By Jeff Linwood, Dave Minte – Apress

**Course Outcomes:**

- Students should gain a comprehensive understanding of the architecture of J2EE, including its various tiers such as presentation, business logic, and data tiers.
- Students should be able to develop enterprise applications using J2EE technologies, including the ability to design and implement user interfaces, business logic, and data access layers.
- Students should be proficient in using various Java EE APIs for developing enterprise applications, including Servlets, JavaServer Pages (JSP).
- Understands and implements JSP and frameworks like Spring, Spring Boot etc.
- Understand and apply the concepts of MVC and tag libraries.

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>CS-30: Programming in Python</b>		
<p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Understanding basic syntax of python and emphasize the importance of writing clear and concise code documentation and comments.</li> <li>• Familiarize students with Python’s style guide and best practices for writing Pythonic code.</li> <li>• Familiarize students with built-in data structures in Python such as lists, tuples, dictionaries etc.</li> </ul> <p><b>Prerequisites:</b></p> <ul style="list-style-type: none"> <li>• Basic Computer Skills.</li> <li>• Fundamental Programming Concepts.</li> <li>• Problem-Solving Skills.</li> </ul>		
<b>Unit No.</b>	<b>Topic</b>	<b>Detail</b>
<b>1</b>	Introduction to Python	<ul style="list-style-type: none"> <li>• The basic elements of Python</li> <li>• Branching programs</li> <li>• Strings and Input</li> <li>• Iteration</li> <li>• Functions and Scoping, Specifications, Recursion</li> <li>• Global variables, Modules, Files</li> <li>• Tuples, Lists and Mutability</li> <li>• Functions as Objects, Strings</li> <li>• Tuples and Lists, Dictionaries</li> </ul>
<b>2</b>	OOP using Python	<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,</li> <li>• Encapsulation and information hiding,</li> <li>• Search Algorithms, Sorting Algorithms,</li> <li>• Hashtables</li> </ul>
<b>3</b>	Plotting using PyLab	<ul style="list-style-type: none"> <li>• Plotting using PyLab,</li> <li>• Plotting mortgages and extended examples,</li> <li>• Fibonacci sequence revisited, Dynamic programming and the 0/1 Knapsack algorithm,</li> <li>• Dynamic programming and divide and conquer</li> </ul>

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>4</b>	Network Programming and GUI using Python	<ul style="list-style-type: none"> <li>• Network Programming: <ul style="list-style-type: none"> <li>▪ Protocol, Sockets,</li> <li>▪ Knowing IP Address,</li> <li>▪ URL, Reading the Source Code of a Web Page,</li> <li>▪ Downloading a Web Page from Internet,</li> <li>▪ Downloading an Image from Internet,</li> <li>▪ A TCP/IP Server, A TCP/IP Client,</li> <li>▪ A UDP Server, A UDP Client,</li> <li>▪ File Server, File Client,</li> <li>▪ Two-Way Communication between Server and Client,</li> <li>▪ Sending a Simple Mail.</li> </ul> </li> <li>• GUI Programming: <ul style="list-style-type: none"> <li>▪ Event-driven programming paradigm;</li> <li>▪ creating simple GUI;</li> <li>▪ buttons, labels, entry fields, dialogs;</li> <li>▪ widget attributes - sizes, fonts, colors, treeview, layouts, nested frames</li> </ul> </li> </ul>
<b>5</b>	Connecting with Database	<ul style="list-style-type: none"> <li>• Verifying the MySQL dB Interface Installation,</li> <li>• Working with MySQL Database,</li> <li>• Using MySQL from Python,</li> <li>• Retrieving All Rows from a Table,</li> <li>• Inserting Rows into a Table,</li> <li>• Deleting Rows from a Table,</li> <li>• Updating Rows in a Table,</li> <li>• Creating Database Tables through Python</li> </ul>

**ReferenceBooks:**

- “Core Python Programming” by Dr.R. NageswaraRao– 2017 Edition, Dreamtech Press
- JohnVGuttag.“Introduction to Computation and Programming Using Python”, Prentice Hall of India
- Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in python, Pearson
- WesleyJ Chun, CorePython Applications Programming, 3rd Edition. Pearson
- Michael Bowles, Machine Learning in Python, Essential techniques for predictive analysis, Wiley

**Course Outcomes:**

- Understand the concept of programming with Python
- Understand the OOP using Python
- Implementing the plotting using PyLab
- Understand the Network Programming and GUI
- Understand and Implement database connectivity

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>CS-31: Cyber Security</b>		
<p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Learn the foundations of Cyber Security and threat landscape.</li> <li>• To equip students with the technical knowledge and skills needed to protect and defend against cyber threats.</li> <li>• To expose students to governance, regulatory, legal, economic, environmental, social and ethical contexts of cyber security.</li> <li>• To develop skills in students that can help them plan, implement, and monitor cyber security mechanisms to ensure the protection of information technology assets.</li> </ul> <p><b>Prerequisites:</b></p> <ul style="list-style-type: none"> <li>• Basic Computer Skills.</li> <li>• OS and Programming knowledge, Networking Fundamentals</li> <li>• Critical Thinking and Problem-Solving Skills</li> </ul>		
<b>Unit No.</b>	<b>Topic</b>	<b>Detail</b>
<b>1</b>	Introduction to Cyber Security	<ul style="list-style-type: none"> <li>• Defining Cyberspace and Overview of Computer and Web-technology</li> <li>• Architecture of cyberspace,</li> <li>• Communication and web technology,</li> <li>• Internet, World wide web,</li> <li>• Advent of internet,</li> <li>• Internet infrastructure for data transfer and governance,</li> <li>• Internet society,</li> <li>• Regulation of cyberspace</li> <li>• Concept of cyber security</li> <li>• Issues and challenges of cyber security</li> </ul>
<b>2</b>	Cyber Crime and Cyber law	<ul style="list-style-type: none"> <li>• Classification of cyber crimes</li> <li>• Common cyber crimes <ul style="list-style-type: none"> <li>▪ cyber crime targeting computers and mobiles</li> <li>▪ Cyber crime against women and children</li> <li>▪ Financial frauds</li> <li>▪ Social engineering attacks</li> <li>▪ Malware and ransomware attacks</li> <li>▪ Zero day and zero click attacks</li> </ul> </li> <li>• Cybercriminals modus-operandi</li> <li>• Reporting of cyber crimes</li> <li>• Remedial and mitigation measures</li> <li>• Legal perspective of cyber crime</li> <li>• IT Act 2000 and its amendments</li> <li>• Cyber crime and offences</li> <li>• Organisations dealing with Cyber crime and Cyber Security in India</li> </ul>

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

		<ul style="list-style-type: none"> <li>• Case studies</li> </ul>
<b>3</b>	Social Media Overview and Security	<ul style="list-style-type: none"> <li>• Introduction to Social networks</li> <li>• Types of Social media</li> <li>• Social media platforms</li> <li>• Social media monitoring</li> <li>• Hashtag</li> <li>• Viral content</li> <li>• Social media marketing</li> <li>• Social media privacy</li> <li>• Challenges, opportunities and pitfalls in Online Social Network</li> <li>• Security issues related to social media</li> <li>• Flagging and reporting of inappropriate content</li> <li>• Laws regarding posting of inappropriate content</li> <li>• Best practices for the use of Social media</li> <li>• Case studies</li> </ul>
<b>4</b>	E-commerce and Digital Payments	<ul style="list-style-type: none"> <li>• Definition of E-Commerce</li> <li>• Main components of E-Commerce</li> <li>• Elements of E-Commerce security</li> <li>• E-Commerce threats</li> <li>• E-Commerce security best practices</li> <li>• Introduction to digital payments</li> <li>• Components of digital payment and stake holders</li> <li>• Modes of digital payments: <ul style="list-style-type: none"> <li>○ Banking Cards</li> <li>○ Unified Payment Interface (UPI)</li> <li>○ e-Wallets</li> <li>○ Unstructured Supplementary Service Data (USSD)</li> <li>○ Aadhar enabled payments</li> </ul> </li> <li>• Digital payments related common frauds and preventive measures</li> <li>• RBI guidelines on digital payments and customer protection in unauthorized banking transactions</li> <li>• Relevant provisions of Payment Settlement Act, 2007.</li> </ul>
<b>5</b>	Digital Devices Security, Tools and Technologies for Cyber Security	<ul style="list-style-type: none"> <li>• End Point device and Mobile Phone security</li> <li>• Password policy</li> <li>• Security patch management</li> <li>• Data backup</li> <li>• Downloading and management of third party software</li> <li>• Device security policy</li> <li>• Cyber Security best practices</li> <li>• Significance of host firewall and Anti-virus</li> <li>• Management of host firewall and Anti-virus</li> </ul>

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

	<ul style="list-style-type: none"><li>• Wi-Fi security</li><li>• Configuration of basic security policy and permissions</li></ul>
--	---

**Reference Books:**

- Cyber Crime Impact in the New Millenium, by R. C. Mishra, Auther Press. Edition 2010.
- Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Sumit Belapure and nina Godbole, Wiley India Pvt. Ltd. (First Edition, 2011)
- Security in the Digital Age: Social Media Security Threats and Vulnerabilities by Henry A. Oliver, Create Space Independent Publishing Platform (Pearson, 13<sup>th</sup> November, 2001)
- Electronic Commerce by Elias M. Awad, Prentice Hall of India Pvt. Ltd.
- Cyber Laws: Intellectual Property & E-Commerce Security by Kumar K, Dominant Publishers
- Network Security Bible, Eric Cole, Ronald Krutz, James W. Conley, 2<sup>nd</sup> Edition, Wiley India Pvt. Ltd.
- Fundamentals of Network Security by E. Maiwald, McGraw Hill.

**Course Outcomes:**

- After completion of this module, students would be able to understand the concept of Cyber security and issues and challenges associated with it.
- Understand the cyber crimes, their nature, legal remedies and as to how report the crimes through available platforms and procedures.
- Able to appreciate various privacy and security concerns on online Social media and understand the reporting procedure of inappropriate content, underlying legal aspects and best practices for the use of Social media platforms.
- Understand the basic concepts related to E-commerce and digital payments.
- Familiar with various digital payment modes and related cyber security aspects, RBI guidelines and preventive measures against digital payment frauds.
- Understand the basic security aspects related to Computer and Mobiles.
- Able to use basic tools and technologies to protect their devices.

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>CS-32: Mini Project</b>			
Objectives:			
<ul style="list-style-type: none"> <li>• To gain hands-on experience in using programming languages, tools, and technologies.</li> <li>• To understand and implement the complete Software Development Life Cycle (SDLC).</li> <li>• To enhance problem-solving and analytical skills by designing solutions to real-life problems.</li> <li>• To improve project planning and time management abilities.</li> <li>• To practice documentation and reporting skills.</li> </ul>			
No.	Topic	Guidelines for Mini Project	
1	General	<ul style="list-style-type: none"> <li>• Choose a project topic</li> <li>• Projects can be individual or group-based (2 Students)</li> <li>• Submit a project proposal with the title, objective, scope ,tools/ technologies to be used, and team members.</li> <li>• Get the proposal approved by the project guide/faculty coordinator.</li> <li>• Use any suitable programming language or platform</li> <li>• Project must be under the supervision of college faculties.</li> </ul>	
2	Documentation	<ul style="list-style-type: none"> <li>• Maintain a project report including the following:</li> <li>• Title Page</li> <li>• Certificate</li> <li>• Acknowledgment</li> <li>• Table of Contents</li> <li>• Introduction &amp; Objective</li> <li>• System Analysis (Problem Definition, Feasibility Study)</li> <li>• System Design (DFD, ER Diagram, etc.)</li> <li>• Data Dictionary</li> <li>• Implementation (Screenshots)</li> <li>• Testing (Test cases and results)</li> <li>• Conclusion &amp; Future Scope</li> </ul>	
3	Submission	<ul style="list-style-type: none"> <li>• Submit a soft copy (CD/Pen drive) and hard copy of the project report for SEE. Project report must be printed on both side of the page.</li> </ul>	
4	Assessment Criteria	CCE –Continuous and comprehensive evaluation done by institute.	50 Marks
		SEE –Exam is conducted by Saurashtra University using external examiner (examiner will evaluate the executable project, hard copy of the project and take the viva voce)	50 Marks

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>CS-33: Practical Based on CS – 29 and CS – 30</b>	
<p><b>CCE- Continuous and comprehensive Evaluation as follow</b></p> <ul style="list-style-type: none"> <li>• The continuous Comprehensive Evaluation (CCE) for each subject will be conducted by the teacher of that subject. The teacher will decide how the evaluation will be done. Usually CCE includes things like class participation, case studies and presentation, assignments, tutorials, small test (announced or surprised), quizzes and attendance or a mix of these.</li> <li>• Students must submit their work for internal evaluation on time to time.</li> <li>• Another part of CCE is the mid-term exam, which is compulsory for all students. This exam will be conducted internally by the college.</li> </ul>	50 Marks
<p><b>SEE – Semester End Examination as per the following</b></p> <ul style="list-style-type: none"> <li>• Practical exams may be scheduled before or after the theory examinations. Exam is conducted by Saurashtra University using external examiner (3 hours duration)</li> <li>• Students must prepare a practical notebook/book for the final practical examination. (The practical book serves as a record of all practical work, observations, procedures and results performed during the semester in lab. It is essential for evaluation during the final practical examination)</li> </ul>	50 Marks

<b>CS-33: Practical Based on CS – 29 and CS – 30</b>	<b>Total Marks - 100</b>	
<b>Topics</b>	<b>CCE</b>	<b>SEE</b>
<b>CS-29: Advance Java Programming (J2EE)</b>	<b>25</b>	<b>25</b>
<b>CS-30: Programming in Python</b>	<b>25</b>	<b>25</b>

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>CS-34: Introduction to AI</b>		
<p><b>Objectives:</b></p> <ul style="list-style-type: none"> <li>• Develop a comprehensive understanding of the fundamental concepts and applications of Artificial Intelligence.</li> <li>• Gain knowledge of the major techniques and technologies used in Machine Learning and their applications in various domains.</li> <li>• Develop an understanding of Natural Language Processing and its applications in fields such as chatbots, sentiment analysis, and language translation.</li> <li>• Explore the applications and techniques of Computer Vision in real-world scenarios and understand the ethical considerations related to its use.</li> <li>• Stay up-to-date with emerging trends and advancements in AI, and understand their implications for society and the workforce</li> </ul> <p><b>Prerequisites:</b></p> <ul style="list-style-type: none"> <li>•</li> </ul>		
Unit No.	Topic	Detail
1	Introduction to AI	<ul style="list-style-type: none"> <li>• Definition of AI</li> <li>• Brief History of AI</li> <li>• Applications of AI</li> <li>• Ethical considerations in AI</li> <li>• Overview of AI Technologies and techniques</li> </ul>
2	Computer Vision	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Basic techniques of Computer Vision</li> <li>• Applications of Computer Vision</li> <li>• Computer Vision Libraries and Tools</li> <li>• Ethical Considerations in Computer Vision</li> </ul>
3	Emerging Trends in AI	<ul style="list-style-type: none"> <li>• Advanced AI technologies and techniques</li> <li>• AI and IOT</li> <li>• AI and Robotics</li> <li>• Future directions of AI research and development</li> <li>• Implications of AI for society and the workforce</li> </ul>

**ReferenceBooks:**

- Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
- Shane, M. (2018). Artificial intelligence and ethics. Morgan & Claypool Publishers.
- Russell, S. J., & Norvig, P. (2020). Artificial intelligence: A modern approach. Pearson.

**Course Outcomes:**

- Students will be able to define Artificial Intelligence, describe its history and applications,

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

and analyze ethical considerations related to AI.

- Students will be able to understand the basics of Machine Learning, including the different types of algorithms, data preparation, and processing. They will also be able to identify successful Machine Learning projects.
- Students will be able to identify the different techniques used in Computer Vision, understand the applications of Computer Vision, and identify the ethical considerations related to Computer Vision.
- Students will be able to identify emerging trends in Artificial Intelligence, including advanced AI technologies and techniques, AI and IoT, AI and Robotics, and future directions of AI research and development. They will also be able to analyze the implications of AI for society and the workforce.

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>BCA-5</b>	
<b>CS –29: Advance Java Programming(J2EE)</b>	
<b>Minimum following exercise should be performed by the students during the semester</b>	
Consider below tables for JDBC Programs:	
Table Name: emp	
Fields : empno, empnm, designation, city, salary, department	
Table Name: stud	
Fields : rollno, firstname, lastname, course, semester	
(1)	Write a program to insert a record of an employee into the emp table.
(2)	Write a program to display all the records of employees.
(3)	Write a program to display employees whose salary is greater than 50000.
(4)	Write a JDBC program to display employees who are from the city 'Rajkot'.
(5)	Write a program to display employees whose name starts with 'A'.
(6)	Write a program to display employees whose designation is manager.
(7)	Write a program to count the number of employees in the table.
(8)	Write a program to display the employee with the highest salary.
(9)	Write a program to sort employee records by empnm.
(10)	Write a program which accepts empno from the user and displays the corresponding employee record.
(11)	Write a program which accepts a department name from the user and displays the employee name along with their designation.
(12)	Write a Program which inserts a record of a student by using Prepared Statement.
(13)	Write a Program which updates a record of a student by using Prepared Statement.
(14)	Write a Program to delete a record of a student whose rollno is given by the user.
(15)	Write a program which inserts a default record of an employee using a callable statement. (Procedure without parameter)
(16)	Write a program which inserts a record of an employee using a callable statement. (Procedure with Parameter)
(17)	Write a program which displays employee's designation by providing empno using callable statement.
(18)	Write a program which displays all the records of employees whose designation is provided by the user using a callable statement.
(19)	Write a program that performs CRUD operation on product table having fields like pid, productname, price and quatity.
(20)	Write a program to connect to a database and retrieve metadata.
(21)	Write a servlet to print Hello World.
(22)	Write a servlet that displays today's date and time.
(23)	Write a program that accepts username and after submitting the form, welcomes the user by writing "Welcome Username".
(24)	Write a program that accepts Employee Number, Employee Name, Designation,

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

	Qualifications and after submitting the form display that information on a page using GET method.
(25)	Write a program that registers a student by providing rollno, name, course, semester, hobbies and after submitting the form display the information on a page using POST method. (Hint: By using <code>getParameterNames()</code> and <code>getParameterValues()</code> )
(26)	Demonstrate the Servlet lifecycle with appropriate messages in each stage.
(27)	Write a servlet program that displays the Basic Header Information.
(28)	Write a program that creates a login form and after clicking on login button if login details are correct then displays welcome screen else display appropriate error message.(Hint: take <code>username="admin"</code> , <code>password="admin"</code> )
(29)	Write a Servlet Program that gives the following option using the radio button and perform appropriate operation (Addition, Subtraction, Multiplication, Division).
(30)	Write a program that redirects a page to google.com.
(31)	Write a program that sends a customized error message for "404 - Page Not Found".
(32)	Write a servlet program that demonstrates the use of URL Rewriting.
(33)	Write a servlet program that demonstrates the use of Hidden Form Field.
(34)	Write a servlet program that checks if a cookie exists; if not, it creates a new cookie and adds it to the response. The servlet should then display all the cookies sent by the browser.
(35)	Write a servlet that demonstrates the use of cookies in a web application. The servlet should perform the following tasks: <ul style="list-style-type: none"> <li>• Create a cookie with the name "username" and yourname as a value.</li> <li>• Set the cookie's maximum age to 1 day (24 hours).</li> <li>• Add the cookie to the response.</li> <li>• Retrieve any cookies sent by the client's browser and display their names and values.</li> <li>• If no cookies are found, display a message indicating no cookies were sent.</li> </ul>
(36)	Write a servlet that displays a "Welcome" message if the user is visiting the site for the first time, and "Welcome back" if they have visited before. (Using Cookie)
(37)	Develop a servlet that allows the user to select a background color from a dropdown list. Save the selected color in the cookie and apply it to the response page.
(38)	Write a servlet that displays a "Welcome" message if the user is visiting the site for the first time, and "Welcome back" if they have visited before. (Using session)
(39)	Write a servlet program that accepts a user's name through a form and stores it in an HTTP session. On subsequent visits during the session, greet the user by name.
(40)	Write a Java Servlet application that authenticates users using a login form. Upon successful login, store the user's information (like username) in the HTTP session. Display a personalized welcome page using session data. For the unsuccessful attempt display appropriate message. Also, implement a logout function that invalidates the session.
(41)	Create a servlet that counts how many times a user has visited the page during a session. Display the visit count on each load.
(42)	Develop a servlet that allows the user to select a background color from a dropdown list. Save the selected color in the session and apply it to the response page.
(43)	Write a servlet program that checks if a user has logged in. If not, redirect to a login form. Once logged in, store the username in the session and display a welcome message.
(44)	Write a JSP to print Hello World.

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

(45)	Write a JSP that accepts Student's details from the html page and display all the information.
(46)	Write a JSP that accepts a year from the user and print whether it is a leap year or not.
(47)	Write a JSP program to calculate simple interest by accepting values from the user.
(48)	Write a JSP page that performs a division operation. If the user enters zero as the divisor, handle the exception using a custom error page. Use the isErrorPage="true" attribute in the error page to display the exception details.
(49)	Configure an error page in web.xml that catches HTTP 404 errors (page not found). Display a user-friendly message using a JSP error page without exposing technical details.
(50)	Write a JSP application that uses <jsp:include> to include reusable pages such as header and footer into a main content page.
(51)	Create a JSP application where the user enters a favorite color; use <jsp:forward> in process.jsp to redirect to blue.jsp if the input is "blue", otherwise forward to default.jsp—ensure no output is sent before forwarding.
(52)	Create a JSP page that uses the <jsp:useBean> tag to instantiate a User JavaBean with properties name, email, and age, and display the user details on the page.
(53)	Create a JavaBean called EmployeeBean with properties for name, designation, and salary. Use the <jsp:useBean> tag to instantiate the bean with the ID employee and set its scope to session. Then, set the name, designation, and salary properties using <jsp:setProperty> tags. Finally, retrieve and display these properties (name, designation, and salary) on the page using Java expression tags (\${}).
(54)	Write a JSP program to demonstrate the use of application implicit objects. On the first JSP page, use the application object to maintain a global visitor counter that increments every time the page is accessed. Display the total number of users who have visited the site since the server started.
(55)	Write a JSP program that allows the user to log in by entering their username (login.jsp). Once the form is submitted, store the username in the session object and display a personalized welcome message on a separate JSP page (welcome.jsp). The application should retain the username across multiple pages using the session and also provide an option to log out (logout.jsp), which will invalidate the session and redirect the user back to the login page (login.jsp). This program demonstrates how session tracking is used to manage user state across multiple JSP pages.
(56)	Write a JSP program that stores the user's preferred theme (e.g., "light" or "dark") using cookies. When the user selects a theme from a form and submits it, set a cookie with the selected value and redirect the user to a welcome page. On subsequent visits, retrieve the cookie and automatically apply the selected theme to the page layout.
(57)	Write a JSP that displays a "Welcome" message if the user is visiting the site for the first time, and "Welcome back" if they have visited before. (using cookie)
(58)	Write a JSP program to display current date and time using JSP Expression Language (EL).
(59)	Write a JSP application that demonstrates the use of JSP Expression Language (EL) for dynamic content rendering across multiple JSP pages. The application should begin with a form in index.jsp where the user can input their name. Upon submitting the form, the user's name should be passed to welcome.jsp using the request object. In welcome.jsp, the name passed from index.jsp should be displayed using EL.
(60)	Write a JSP program that calculates addition, subtraction, multiplication and division using expression language and display the results dynamically.
(61)	Write a JSP program to check whether it is a leap year or not using JSTL.

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

(62)	Write a JSP program to check whether it is an even or odd number using JSTL.
(63)	Write a JSP program to print prime numbers from 1 to 100 using JSTL.
(64)	Write a JSP program to display whether a number is divisible by 2 using JSTL.
(65)	Write a program with a Servlet that handles user input, Accepts Username as an input. Use a JSP page to welcome User by providing a username. Use the MVC pattern to separate concerns.
(66)	Write a login program. Use "admin" as the username and "admin" as the password. If the username and password are correct, display a JSP page that welcomes the user. If they are incorrect, display a JSP page that says "Incorrect username or password." Use the MVC pattern to separate concerns. (Create 2 separate views)
(67)	Write a login program. Fetch the username and password from the database. If the username and password are correct, display a JSP page that welcomes the user. If they are incorrect, display a JSP page that says "Incorrect username or password." Use the MVC pattern to separate concerns and create two separate views. (Table: Login, Fields: username, password)
(68)	Develop a Registration module using JSP, Servlet, and MVC architecture with database connectivity. Collect user details like Username, Password, and Mobile. Store the data in a database upon form submission. (Table: User, Fields : username, password, mobile)
(69)	Write a program that allows a user to submit a request to view a list of students by providing their course and semester. Use a Servlet to retrieve data from a model and display it dynamically on a JSP page. (Table: Student)
(70)	Write a Contact Management System using MVC where users can add contacts with names and phone numbers, and view all saved contacts. The Model handles contact data and interactions with the data store, the View includes a form for adding contacts and a display page for listing them, and the Controller processes input from the form, updates the model, and directs users to the appropriate view.
(71)	Write an MVC application using Servlet, JSP, and JavaBean to insert and display employee data. The application works on an emp table with fields empno, empnm, designation, and dept.
(72)	Develop a full CRUD application using the MVC model with Servlet, JSP, and JavaBean to manage student records. The application operates on studnt table containing rollno, name, course, and semester. JSP pages are used for data entry, display, update, and delete views. Servlets control the flow and interact with Beans for database transactions.
(73)	Create a simple Hibernate application that connects to a MySQL database. Define an entity class (e.g., Student with attributes rollno, name, course and semester), use Hibernate to save a new Student object, and then retrieve it using a session to display its details.
(74)	Create an entity class called Employee with attributes empno, empnm and department. Use Hibernate to insert an Employee record in the database, and then retrieve and display all employees from the database.
(75)	Create a Hibernate application with an entity class called Product that has attributes id, name, and price. Use Hibernate Query Language (HQL) to fetch all products from the database and display their id, name, and price in a list.
(76)	Create a simple login application using the Spring framework. Implement a LoginController to manage user authentication and define the necessary mappings using Spring MVC. Additionally, create templates for the login form, a success page upon successful login, and a failure page for incorrect credentials.
(77)	Develop an application using Spring Boot and Spring Data JPA to insert and display

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

	employee data. It manages an Employee table with empno, empnm, designation, and dept fields.
(78)	Develop an application using Spring Boot and Spring Data JPA to perform CRUD operations on a Student table without. Fields include rollno, name, course, and semester. Spring Boot simplifies the project setup, and JPA handles all database actions.

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

<b>BCA-5</b>	
<b>CS –30: Programming in Python</b>	
<b>Minimum following exercise should be performed by the students during the semester</b>	
(1)	Write a Python program to display as following: Hello World. This is Python.
(2)	Write a Python program to work with data types (i.e., int, float, complex, boolean string types)
(3)	Write a Python program to work with following functions: id(), type(),range()
(4)	Write a Python program to work with type conversion functions.
(5)	Write a Python program to work with the following Operators in Python with suitable examples. i) Arithmetic Operators ii) Relational Operators iii) Assignment Operator iv) Logical Operators v) Bit wise Operators vi) Ternary Operator
(6)	Write Python programs to work with the following: i) input( ) ii) print( ) iii) 'sep' attribute iv) 'end' attribute v) replacement Operator ( { })
(7)	Write a Python program to work with the Conditional statements in Python with suitable examples. i) if statement ii) if else statement iii) if – elif – else statement
(8)	Write a Python program to work with the Iterative statements in Python with suitable examples. i) while loop ii) for loop
(9)	Write a Python program to work with the control transfer statements in Python with suitable examples. i) break ii) continue iii) pass
(10)	Write a Python program to work with the various ways of accessing the string. i) By using Indexing (Both Positive and Negative) ii) By using Slice Operator
(11)	Write a Python program to work with the read and write operations on a file.
(12)	Write a Python program to copy the contents of a file to another file.
(13)	Write a Python program to work with the count frequency of characters in a given file.
(14)	Write a Python program to print each line of a file in reverse order.
(15)	Write a Python program to compute the number of characters, words and lines in a file.

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

(16)	Write a Python program to work with the different ways of creating list objects with suitable example programs.
(17)	Write a Python program to work with the following functions/methods which operates on lists in Python with suitable examples: i) list( ) ii) len( ) iii) count( ) iv) index( ) v) append( ) vi) insert( ) vii) extend() viii) remove( ) ix) pop( ) x) reverse( ) xi) sort( ) xii) copy( ) xiii) clear( )
(18)	Write a Python program to work with the different ways of creating tuple objects with suitable example programs.
(19)	Write Python programs to print the following Patterns: 1 22 333 4444 55555  A A B A B C A B C D A B C D E  ***** **** *** ** *
(20)	Write a Python program to work with the the following functions/methods which operates on tuples in Python with suitable examples: i) len( ) ii) count( ) iii) index( ) iv) sorted( ) v) min ( )vi)max( ) vii) cmp( ) viii) reversed( )
(21)	Write a Python program to work with the different ways of creating set objects with suitable example programs.
(22)	Write a Python program to work with the following functions/methods which operates on sets in Python with suitable examples: i) add( ) ii) update( ) iii) copy( ) iv) pop( ) v) remove( )vi)discard( ) vii) clear( ) viii) union( ) ix) intersection( ) x) difference( )
(23)	Write a Python program to work with the different ways of creating dictionary objects with suitable example programs.
(24)	Write a Python program to work with the following functions/methods which operates on dictionary in Python with suitable examples: i) dict( ) ii) len( ) iii) clear( ) iv) get( ) v) pop( )vi)popitem( ) vii) keys( ) viii) values( ) ix) items( ) x) copy( ) xi) update( )
(25)	Write a Python program to return multiple values at a time using a return statement.
(26)	Write a Python program to demonstrate Local and Global variables.

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

(27)	Write a Python program to work with the lambda functions in Python with suitable examples.
(28)	Write a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 60 years old.
(29)	Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.
(30)	Write a python program to generate the Fibonacci series.
(31)	Write a function that reverses the user defined value using python.
(32)	Write a function to check if the input value is Armstrong or not.
(33)	Write the function for the Input number is Palindrome or not.
(33)	Write a recursive function to print the factorial for a given number
(34)	Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.
(35)	Define a function that computes the length of a given list or string.
(36)	Write a Python program to print a specified list after removing the 0th, 2nd, 3rd and 5th elements.
(37)	Write a Python script to sort (ascending and descending) a dictionary by value
(38)	Write a Python program to sum all the items in a dictionary
(39)	Implement the following Searching and Sorting techniques in Python by using functions. i) Linear Search ii) Binary Search iii) Selection Sort iv) Bubble Sort v) Insertion vi) Merge Sort viii) Quick Sort
(40)	Write a Python program to implement encapsulation concept
(41)	Generate different plotting using PyLab. Such as Line plot, Bar chart, Pie chart, Histogram, Scatter plot.
(42)	Write a Python script to plotting a curve
(43)	Write a Python program to implement 0/1 knapsack algorithm
(44)	Write a Python program to use divide and conquer algorithm
(45)	Write a Python program to create a socket
(46)	Write a Python program to identify IP address
(47)	Write a Python program to download source code of a web page
(48)	Write a Python program to download a web page from internet
(49)	Write a Python program to download an image from internet
(50)	Create a TCP/IP Server and client

**B.C.A. (Honours) & B.C.A. (Honours with Research)**  
**(Semester - 5 and Semester - 6)**  
**To be effective from June – 2025**  
**Saurashtra University**

(51)	Create two-way communication such as server to client and client to server
(52)	Create a UDP server and client
(53)	Create a file server and client
(54)	Write a Python program to sending an email
(55)	Write a Python program to create GUI with use of buttons, labels, entry fields.
(56)	Write a Python program to ask confirmation or give alerts using dialogs.
(57)	Write a Python program to create a simple calculator
(58)	Write a Python program to get a list of existing databases.
(58)	Write a Python program to insert a row into a table.
(59)	Write a Python program to update a row into a table.
(60)	Write a Python program to delete a row from a table.
(61)	Create a database named dbStudent and a table named tblStudInfo: Table structure: student_id, student_name, stream, college_name, contact_number, remarks  Write a Python program to insert student information.
(62)	Write a Python program to update student information.
(63)	Write a Python program to delete student information.