PPSU

P P SAVANI UNIVERSITY

SCHOOL OF ENGINEERING

BACHOLAR OF COMPUTER APPLICATION (BCA)

WITH HONORS

SYLLABUS BOOK

AY 2025-26

INSTITUTE VISION
To emerge as an Institute of Excellence by imparting value-based education aided with Research, Innovation and Entrepreneurial skills.

INSTITUTE MISSION	
1.	To impart the holistic engineering education of highest quality & prepare socially responsible professionals with entrepreneurial skills.
2.	To prepare value-aided engineering professionals to meet up global industry requirements by imparting cutting edge professional education.
3.	To inculcate the attitude of research and innovation among the stake holders through experiential and project-based teaching-learning pedagogy.
4.	To acquire global talent pool by providing world class amenities for teaching, learning & research.

Graduates will demonstrate ability to:

PEO No	PROGRAMME EUCATIONAL OBJECTIVES
PEO 1	Apply software development skills and computing knowledge to design innovative applications that effectively and efficiently address business and societal challenges.
PEO 2	Build successful careers as application developers, software developer, system analysts, or entrepreneurs by leveraging strong programming, problem solving and analytical abilities.
PEO 3	Uphold ethical practices, demonstrate effective communication and teamwork, and contribute to the continuous advancement of knowledge in rapidly evolving fields.

PO No	PROGRAMME OUTCOMES
PO 1	Foundational & Domain Knowledge: Demonstrate a solid understanding of mathematical foundations, computing principles, and domain-specific theories to model and solve complex IT problems.
PO 2	Problem Formulation & Analysis: Decompose and critically analyse real-world IT problems using algorithmic thinking and computational reasoning to derive structured, well-justified solutions.
PO 3	Design, Development & Innovation: Design, develop, and deploy innovative IT solutions that satisfy functional and non-functional requirements while adhering to engineering best practices and user-centered design.
PO 4	Research, Inquiry & Evidence-Based Thinking: Investigate complex IT challenges through research methods, data analysis, and experimentation to reach evidence-based conclusions and actionable solutions.
PO 5	Modern Tools & Emerging Technologies: Select and apply modern tools, frameworks, and emerging technologies; including AI/ML, cloud, and DevOps to build efficient, industry-relevant IT solutions.
PO 6	Communication, Leadership & Teamwork: Convey technical concepts clearly and contribute effectively as a collaborator or leader within multidisciplinary teams in professional environments.
PO 7	Ethics, Society & Professional Responsibility: Apply ethical frameworks, legal standards, and professional norms in IT practice, with awareness of societal impact, data privacy, cybersecurity, and sustainability.
PO 8	Lifelong Learning & Adaptability: Demonstrate proactive commitment to continuous self-directed learning, staying current with technological advancements and evolving best practices throughout career.

PSO No	PROGRAMME SPECIFIC OUTCOMES (PSO) BACHELOR OF COMPUTER APPLICATION (BCA)
PSO 1	Graduates will develop competency to design computing systems that meet realistic constraints of security and applicability, collaborating effectively in multidisciplinary teams with a positive professional attitude.
PSO 2	Graduates will apply contemporary techniques, tools, and industry-relevant skills to deliver efficient and reliable computing solutions across practical domains.
PSO 3	Graduates will emerge as technically proficient professionals, researchers, and entrepreneurs, well prepared to succeed in competitive examinations and pursue higher academic studies.

Credit Guidelines (General)			
Component	**Hour/Week**	**Credit**	**Total Hours/Semester**
Theory	1	1	15
Practical	2	1	30
Tutorial	1	1	15
Note: In specific cases; extra credits can be granted for specific/important subjects.			

CO-PO Mapping Guidelines		
Mapping Level	**% age Mapping**	**Indicator**
0 / -	0	No Mapping
1	0-33	Low Level (Slightly Mapped)
2	33-66	Medium Level (Moderately Mapped)
3	>66	High Level (Strongly Mapped)

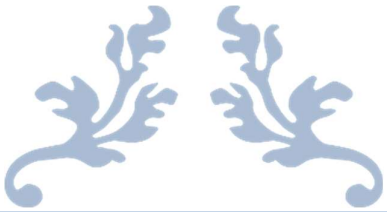
Syllabus Book

B.C.A. with Honors

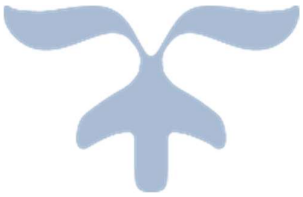
P P Savani University
School of Engineering

Effective From: 2025-26
Authored by: P P Savani University

CONTENT



FIRST YEAR B.C.A.



P P SAVANI UNIVERSITY															
SCHOOL OF ENGINEERING INSTITUTE OF COMPUTER SCIENCE AND APPLICATION															
TEACHING & EXAMINATION SCHEME FOR B.C.A. PROGRAMME with Honors (AY: 2025-26)															
Sem	**Course Code**	**Course Title**	**Offered By**	**Teaching Scheme**					**Examination Scheme**						
				Contact Hours				**Credit**	**Theory**		**Practical**		**Tutorial**		**Total**
				Theory	**Practical**	**Tutorial**	**Total**		**CE**	**ESE**	**CE**	**ESE**	**CE**	**ESE**	
1	ICSH1010	Mathematics for Computer Applications	SH	03	-	02	05	05	40	60	-	-	100	00	200
	ICCA1010	Basics of Web Designing	CA	03	04	-	07	05	40	60	40	60	-	-	200
	ICCS1010	Introduction to Computer Organization	CS	03	-	02	05	05	40	60	-	-	100	00	200
	ICIT1010	Introduction to Computer Programming	IT	03	04	-	07	05	40	60	40	60	-	-	200
	CFLS2110	Elementary Communicative English-I	CFLS	03	-	-	03	03	100	00	-	-	-	-	100
						Total	**27**	**23**							**900**
2	ICSH1020	Statistics	SH	03	-	02	05	05	40	60	-	-	100	00	200
	ICCA1021	Advanced Web Designing	CA	03	04	-	07	05	40	60	40	60	-	-	200
	ICIT1020	Digital Marketing	IT	03	04	-	07	05	40	60	40	60	-	-	200
	ICIT1031	Object Oriented Programming with C++	IT	03	04	-	07	05	40	60	40	60	-	-	200
	CFLS2120	Elementary Communicative English-II	CFLS	02	-	-	02	02	100	00	-	-	-	-	100
						Total	**28**	**22**							**900**

P P Savani University
School of Engineering
Institute of Computer Science and Application
Department of Science & Humanities

Course Code: ICSH1010
Course Name: Mathematics for Computer Applications
Prerequisite Course(s): --

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	-	02	05	40	60	-	-	100	00	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help the learners to

- provide foundation of data representation, logical implementation of data.
- educate mathematical concepts to recognize their applications in computer domain.
- demonstrate a basic understanding of a function, its inverse, composition, and notation.
- model and analyze computational processes using analytic and combinatorial methods.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Number System:** Introduction to Number System, Base, Types of Number Systems, Conversion Between Number Bases, Arithmetic Operations - Addition, Subtraction, Multiplication and Division for Binary, Octal, Hexadecimal Systems, Signed Unsigned Numbers, Binary Coding - BCD, ASCII, EBCDIC, Floating Point Representation of Numbers and Arithmetic Operation with Normalized Floating-Point Numbers.	08	18
2.	**Mathematical Logic:** Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers.	07	16
3.	**Elementary Combinatorics:** Introduction, Basic Counting Principles, Permutation and Combination, Mathematical Induction.	07	16
4.	**Matrix Algebra:** Introduction, Types of Matrices, Operations of Matrices, Adjoint Matrices, Solution of System of Equations by Matrix Inversion Method, Applications of Matrix.	07	16
5.	**Determinants:** Formation of Determinants, Minors and Cofactors of the Elements of a Determinant, Properties of Determinants, Applications of Determinants in Computer Science, Cramer's Rule.	08	17
6.	**Analytical Geometry:** Introduction to Cartesian coordinate system, Straight line, Slope of Straight line, Intersection of two straight lines,	09	17

	Equation of circle, Centre and Radius, Tangent, Equation of Parabola, Hyperbola and Ellipse.		
	TOTAL	**45**	**100**

List of Tutorials:

Sr. No	Name of Practical	Hours
1.	Number System-1	02
2.	Number System-2	04
3.	Mathematical Logic	04
4.	Elementary Combinatorics	04
5.	Matrix Algebra-1	02
6.	Matrix Algebra-2	04
7.	Determinants-1	02
8.	Determinants-2	04
9.	Analytical Geometry-1	02
10.	Analytical Geometry-2	02
	TOTAL	**30**

Text Book (s):

Title	Author/s	Publication
Discrete Mathematics	T. Veerarajan	Tata McGraw Hill

Reference Book(s):

Title	Author/s	Publication
Discrete Mathematics and its Applications	Kenneth H. Rosen	Tata McGraw Hill
Discrete Mathematical Structures with Applications to Computer Science	J. P. TremblayR. Manohar	Tata McGraw Hill
Analytical Geometry: 2D and 3D	P R Vittal	Pearson
Introduction to Computer Science	ITL ESL	Pearson

Web material link:

- https://onlinecourses.swayam2.ac.in/nou25_cm04/preview
- https://onlinecourses.swayam2.ac.in/nou25_cs01/preview?
- https://onlinecourses.nptel.ac.in/noc25_cs26/preview

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Tutorial:

- Continuous Evaluation consists of performance of tutorial which will be evaluated out of 10 marks for each practical and average of the same will be converted to 50 marks.
  Viva/ Oral performance consists of 50 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICSH1010	MATHEMATICS FOR COMPUTER APPLICATIONS
CO 1	Convert decimal to binary, octal, hexadecimal for data representation and calculate arithmetic operations.
CO 2	Compute permutations and combinations on a given set of data.
CO 3	Evaluate the solution of system if linear equations through elimination method.
CO 4	Discuss the equation of straight line in different forms and related properties.

Mapping of CO with PO

ICSH1010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	2	2	1	1				
CO 2	2	2	1	1				
CO 3	2	2	1	1				
CO 4	2	2	1	1				

Mapping of CO with PSO

ICSH1010	PSO1	PSO2	PSO3
CO 1	1	1	
CO 2	1	1	
CO 3	1	1	
CO 4	1	1	

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Number System	1,2,3,5
2	Mathematical Logic	1,2,4,6
3	Elementary Combinatorics	1,2,3,5
4	Matrix Algebra	1,2,3,5
5	Determinants	1,2,3,5
6	Analytical Geometry	1,2,3,5

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Application

Course Code: ICCA1010
Course Name: Basics of Web Designing
Prerequisite Course(s): -

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	04	-	05	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to explore the concepts of web designing and develop user interface using markup language, applying styles and usage of scripting language.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Basics of User Interface** User Interface: Importance and Benefits, Designing Principles, Design Commandments, Human Interaction with Computers, User Characteristics: Knowledge & Experience, Psychological and Physical, Design Considerations: Human and Technical	08	10
2.	**The Web and Markup Language** Basic Terminologies: Web, Web Browser, Web Server, Web Space, Intranet, Internet, Domain Name, URL, IP Address, Website and other applications, Planning Website: Types of Sites, Lifespan of Site, An Overview of Web Technologies, Markup Language Elements: Root, Metadata, Heading, Paragraph, Lines, Formatting, Form, Listing, Linking, Table, Markup Language Character Entities	07	20
3.	**Advanced Markup Language** Form Attributes, Semantic Elements, Graphics Elements, Multimedia Elements, Advance Input Element's Attributes, Types of Input Element	09	20
4.	**Applying Styles** Syntax and Structures of Applying Styles, Styling Selectors: Element, Class, ID, Pseudo and Universal, Applying Styles using Inline, Internal and External, Styling Properties: Font, Display, Box, Background and Border	08	10
5.	**Scripting Language** An Overview of Server-side and Client-side Scripting Languages, Embedding Scripting Language into Web Page,	06	20

	Variables and Data types, Conditional and Looping Statements, Array: Declaration, Initialization and Operations, User-defined Functions: Creation, Calling and Return a Value		
6.	**Scripting Language Libraries** Incorporating Scripting Language Library into webpage, Scripting Methods: Retrieving Attributes of Markup Language Elements, Traversing Markup Language Elements, Handling Mouse and Keyboard Events	07	20
	TOTAL	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Implement HTML Attributes, HTML Headings and HTML Paragraphs.	6
2.	Implement HTML Styles and HTML Text Formatting.	4
3.	Implement code to add Links in HTML.	4
4.	Implement code to add Images in HTML.	4
5.	Implement code to create different types of frame using HTML.	8
6.	Create a static web page using HTML to display P PSavani University information.	6
7.	Style a webpage using colors, borders, and padding.	6
8.	Build a layout with media queries and Flexbox.	8
9.	Add a JavaScript-based interactive form.	6
10.	Develop a responsive webpage using Bootstrap.	8
	TOTAL	**60**

Text Book (s):

Title	Author/s	Publication
The Essentials Guide to User Interface Design	Wilbert O. Galitz	Wiley
HTML 5 Black Book: Covers CSS3, Javasvript, XML, XHTML, AJAX, PHP and jQuery.	DT Editorial Services	Dreamtech Press

Reference Book (s):

Title	Author/s	Publication
Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP	Ivan Bayross,	Wiley
JavaScript and CSS Development using jQuery	Rechard Y.	Wrox [RY]
Beginning HTML5 and CSS3	Richard Clark, Oli S.	Apress
HTML5- the missing manual	Matthew MacDonald	O'REILLY

Web Material Link(s):

- https://www.codecademy.com/catalog/subject/web-development
- https://www.w3schools.com/html/default.asp
- https://developer.mozilla.org/en-US/

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICCA1010	BASICS OF WEB DESIGNING
CO 1	Summarize the necessity of user interface and applying designing principles.
CO 2	Construct and enhance user interface by using markup language features.
CO 3	Select and apply styling features to user interface.
CO 4	Analyse and apply effects using scripting language.
CO 5	Analyse and implement events using scripting language's library.

Mapping of CO with PO

ICCA1010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	3	2					2
CO 2	3	3	3		2			2
CO 3	3	3	3		2			2
CO 4	3	3	2	3	2			2
CO 5	3	3	2	3	2			2

Mapping of CO with PSO

ICCA1010	PSO1	PSO2	PSO3
CO 1	1	2	
CO 2	1		
CO 3	1		3
CO 4	1		3
CO 5	1		3

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Basics of User Interface	1,2
2	The Web and Markup Language	2,3
3	Advanced Markup Language	3,4
4	Applying Styles	3,4,6
5	Scripting Language	3,4,5
6	Scripting Language Libraries	4,5,6

P P Savani University
School of Engineering
Institute of Computer Science and Application
Department of Computer Science

Course Code: ICCS1010
Course Name: Introduction to Computer Organization
Prerequisite Course(s): --

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	-	02	05	40	60	-	-	100	00	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- impart basic concepts of computer architecture and organization.
- explain key skills of constructing cost-effective computer systems.
- help students in understanding various memory devices.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Overview of Computer Systems** Definition, components, and classifications (desktop, server, embedded systems). Difference between Computer Organization and Computer Architecture. Basic functional units: Input unit, Output unit, Memory unit, Arithmetic and Logic Unit (ALU), Control Unit (CU). Instruction execution cycle: Fetch, decode, execute.	05	15
2.	**Basic Computer Organization** Instruction Codes and Computer Registers; Instruction Cycle and Timing; Input-Output Basics and Interrupt Handling.	04	7
3.	**Computer Data Representation** Data Representation: Decimal, Binary, Octal, and Hexadecimal Numbers; Conversion from one number system to another; Fixed-point Representation; Signed Magnitude, 1's and 2's Complement Representation; Addition, Subtraction, and Basic Logical Operations.	06	15
4.	**Boolean Algebra and Logic Gates** Basics of Boolean algebra: Boolean operations (AND, OR, NOT, NAND, NOR, XOR, XNOR). Laws of Boolean algebra. De Morgan's theorem. Simplification of Boolean expressions using Karnaugh Maps (up to 4 variables). Digital logic gates and circuits: Truth tables. Combinational circuits. Encoder,Decoder -design &	06	15

	implementation.		
5.	**Processor and Control Unit** Introduction to microprocessors: Basic components of a processor (Registers, ALU, CU). Clock speed, instruction set, and processing modes. Control Unit (CU): Hardwired control. Microprogrammed control. Basics of RISC and CISC architectures.	06	12
6.	**Memory Organization** Types of memory: Primary memory: RAM (Static and Dynamic), ROM (PROM, EPROM, EEPROM). Secondary memory: Hard drives, SSDs. Cache memory: Levels (L1, L2, L3). Virtual memory and paging. Memory hierarchy: Characteristics and organization. Concept of memory access time and performance.	07	12
7.	**Input/Output Systems** Basics of input/output operations. I/O devices: Keyboard, mouse, printers, scanners, etc. I/O data transfer techniques: Programmed I/O. Interrupt-driven I/O. Direct Memory Access (DMA).	05	12
8.	**Basics of Storage and Peripherals** storage devices: Magnetic, optical, and solid-state. RAID (Redundant Array of Independent Disks). Peripheral devices: Monitors, projectors, external storage devices.	06	12
	TOTAL	45	100

List of Tutorial:

Sr. No	Name of Tutorial	Hours
1.	Draw and explain fetch-decode-execute cycles for simple instructions.	02
2.	Practice decimal to binary, octal, and hexadecimal conversions.	04
3.	Perform addition and subtraction using 1's and 2's complement.	04
4.	Convert and represent floating-point numbers.	02
5.	Demonstrate data transfer techniques (Programmed I/O, Interrupt-driven I/O, DMA) with examples.	02
6.	Basic AND, OR, NOT, XOR operations and create truth tables & Implement D, T, and Karnaugh Maps.	02
7.	Explain how the clock speed affects the performance of a microprocessor. How does a processor's instruction set influence its operation	02
8.	Describe the difference between single-cycle and multi-cycle processing modes. Which one is more efficient and why?	02
9.	Illustrate how data is stored and accessed in cache memory.	02
10.	Demonstrate virtual memory concepts and perform simple paging exercises.	02
11.	Compare Programmed I/O, Interrupt-driven I/O, and Direct Memory Access (DMA) in terms of speed, efficiency, and application suitability.	02
12.	Describe the operation of optical storage devices such as CDs, DVDs, and Blu-ray. How do they differ from magnetic storage in terms of data reading and writing	02
13.	describe the concept of RAID and list different RAID levels (RAID 0, RAID 1,	02

	RAID 5, etc.). How do these levels enhance performance and data redundancy	
	TOTAL	**30**

Text Book (s):

Title	Author/s	Publication
Computer System Architecture	M. Morris Mano	Pearson

Reference Book (s):

Title	Author/s	Publication
Computer Architecture and Organization	Ghoshal, Subrata	Pearson
Computer Architecture &Organization	M. Murdocca & V. Heuring	WILEY

Web Material Link(s):

- https://nptel.ac.in/courses/106105163
- EE282 Computer Systems Architecture, Spring 2024
- Intel® Product Specifications

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks

Tutorial:

- Continuous Evaluation consists of performance of tutorial which will be evaluated out of 10 marks for each practical and average of the same will be converted to 50 marks.
- Viva/ Oral performance consists of 50 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICCS1010	INTRODUCTION TO COMPUTER ORGANIZATION
CO 1	Understanding Computer Systems and Classifications
CO 2	Demonstrate the conversion between different number systems and perform data representation using fixed-point and complement methods.
CO 3	Understand memory types, hierarchy, and concepts like virtual memory and paging.
CO 4	Learn the operation of I/O devices and the various data transfer techniques
CO 5	Understand different storage devices and peripherals and their role

Mapping of CO with PO

ICCS1010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2						
CO 2	3	3	2					
CO 3	1	3	3	2	1		2	
CO 4	1				2			1
CO 5		2	3	3				1

Mapping of CO with PSO

ICCS1010	PSO1	PSO2	PSO3
CO 1	2	3	
CO 2			
CO 3			2
CO 4	3	1	
CO 5	1	2	2

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Overview of Computer Systems	1,2
2	Basic Computer Organization	2,3,4
3	Computer Data Representation	1,2
4	Boolean Algebra and Logic Gates	1,2,5
5	Processor and Control Unit	1,2
6	Memory Organization	2,3
7	Input/Output Systems	1,2,4,6
8	Basics of Storage and Peripherals	2,3,5

P P Savani University
School of Engineering
Institute of Computer Science and Application
Department of Information Technology

Course Code: ICIT1010
Course Name: Introduction to Computer Programming
Prerequisite Course(s): --

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	04	-	05	40	60	40	60	00	00	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- understand how programming can help to solve real time problems.
- identify appropriate approach to computational problems.
- develop logic building and problem-solving skills.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Fundamentals of Computer Programming** Basic block diagram and functions of various components of computer, Concepts of Hardware and software, Types of software, Compiler and interpreter, Concepts of Machine level, Assembly level and high-level programming	03	05
2.	**Fundamentals of C** Features of C language, structure of C Program, Development of program, Algorithm and flowchart, Types of errors, debugging, tracing/stepwise execution of program, watching variables values in memory.	04	10
3.	**Control structure in C** Simple statements, Decision making statements, Looping statements, Nesting of control structures, break and continue, goto statement.	06	15
4.	**Array & String, Function** Concepts of array, one- and two-dimensional arrays, declaration and initialization of arrays, string, string storage, Built-instring functions, Concepts of user defined functions, prototypes, definition of function, parameters, parameter passing, calling function, recursive function,Macros, Pre-processing.	07	15
9.	**Recursion**	04	15

	Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort		
10.	**Pointers** Basics of pointers, pointer to pointer, pointer and array, pointer to array, array to pointer, function returning pointer.	06	15
11.	**Structure & Union** Basics of structure, structure members, accessing structure members, nested structures, array of structures, structure and functions, structures and pointers,Accessing Structure members, Union.	08	15
12.	**Dynamic memory allocation & File management** Introduction to Dynamic memory allocation, malloc, calloc, Introduction to file management and its functions.	07	10
	TOTAL	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Write an algorithm and draw a flowchart to find the largest number among three user inputs.	2
2.	Write an algorithm and draw a flowchart to compute the factorial of a number n.	2
3.	Write a program that performs basic arithmetic operations (addition, subtraction, multiplication, and division)and demonstrates the use of different data types.	4
4.	Create a program that uses if, else, and switch statements to implement a simple menu-driven application. Useloops (for, while, and do-while) to repeat tasks	6
5.	Develop a program that calculates the factorial of a number using both iterative and recursive functions	4
6.	Write a program to perform various operations on arrays (e.g., sorting, searching) and strings (e.g.,concatenation, comparison).	6
7.	Implement a program that uses pointers to create and manipulate dynamic arrays, demonstrating the use of malloc, calloc, realloc, and free.	4
8.	Design a student record system using structures that store and display information such as name, roll number,and grades.	6
9.	Write a program to read from and write to files, such as creating a simple text editor that performs basic file operations.	4
10.	Implement a singly linked list with operations like insertion, deletion, and traversal.	6
11.	Develop programs to simulate stack operations (push, pop, peek).queue operations (enqueue, dequeue)using arrays and linked lists.	4
12.	Develop a program to define queue operations (enqueue, dequeue)using arrays and linked lists.	4
13.	Write a program that takes a number (1–7) as input and prints the	4

	corresponding day of the week.	
14.	Provide students with a program containing intentional errors and inefficiencies. Have them use debugging tools (like gdb) to find and fix the errors and optimize the code for better performance.	4
	TOTAL	**60**

Text Book (s):

Title	Author/s	Publication
Programming in ANSI C	E. Balagurusamy	Tata McGraw Hill

Reference Book (s):

Title	Author/s	Publication
Programming in C	Ashok Kamthane	Pearson
Let Us C	Yashavant P. Kanetkar	Tata McGraw Hill
Introduction to C Programming	ReemaThareja	Oxford Higher Education

Web Material Link(s):

- https://www.javatpoint.com/c-programming-language-tutorial
- https://nptel.ac.in/courses/106105085/4
- https://fresh2refresh.com/c-programming/

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 Hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per
- guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICIT1010	INTRODUCTION TO COMPUTER PROGRAMMING
CO 1	Understand the basic concepts of programming.
CO 2	Apply fundamental programming constructs like loops, conditionals, and functions to solve problems.
CO 3	Design and develop structured programs using modular programming techniques.
CO 4	Implement algorithms for problem-solving in a high-level programming language.
CO 5	Debug and test programs to ensure correctness and efficiency.

Mapping of CO with PO

ICIT1010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	1		2			
CO 2	3	3	2	1	2			
CO 3	3	3	3	2	2		1	
CO 4	3	3	3	3	3		2	
CO 5	3	2	2	2	3			

Mapping of CO with PSO

ICIT1010	PSO1	PSO2	PSO3
CO 1	3	2	1
CO 2	3	3	2
CO 3	3	3	3
CO 4	3	3	3
CO 5	2	3	3

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Fundamentals of Computer Programming	1,2
2	Fundamentals of C	1,2
3	Control structure in C	1,2,3,6
4	Array & String, Function	1,2
5	Recursion	1,2,3,4
6	Pointers	2,3,6
7	Structure & Union	2,3,5
8	Dynamic memory allocation & File management	2,3,4,6

P P Savani University
School of Engineering
Institute of Computer Science and Application
Department of Science & Humanities

Course Code: ICSH1020
Course Name: Statistics
Prerequisite Course(s): --

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	-	02	05	40	60	-	-	100	00	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- Demonstrate understanding of statistical methods in support of the analysis, design and application for problem solving in the field of computer science and applications.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Introduction to Data & Descriptive Statistics** Elements, Variables, and Observations, Scales of Measurement , Categorical and Quantitative Data, Cross-Sectional and Time Series Data, Summarizing Categorial Data and Quantitative Data, Frequency Distribution, Relative Frequency and Percentage Distributions, Bar Charts and Pie Charts, Dot Plot, Histogram, Cumulative Distributions, Ogive, Measures of Location: Mean, Median, Mode, Percentiles and Quartiles, Measures of Variability: Range, Interquartile Range, Variance, Standard Deviation, Coefficient of Variation.	10	25
2.	**Exploratory Data Analysis** Distribution Shape, z - Scores, Chebyshev's Theorem, Empirical Rule, Outliners, Five Number Summary, Box Plot.	07	15
3.	**Correlation Analysis** Type and properties of Correlation, Karl-Pearson's coefficient.	05	10
4.	**Introduction to Probability** Experiments, Counting Rules, Assigning Probabilities, Events and their Probabilities, Relationships of Probabilities, Conditional Probability, Bayes' Theorem	06	10
5.	**Discrete and Continuous Probability Distribution**		

	Random Variables, Discrete Probability Distributions, Expected Values and variance, Binomial Probability Distribution, Poisson Probability Distribution, Uniform Probability Distribution, Normal Probability Distribution.	10	25
6.	**Testing of Hypothesis** Introduction, Sampling, Tests of Significance, Null Hypothesis, Alternative Hypothesis, Type 1 and Type 2 errors, Level of Significance, Chi-square test, Student's t-test, Seducer's F-test.	07	15
	TOTAL	**45**	**100**

List of Tutorial:

Sr. No.	Name of Tutorial	Hours
1.	Introduction to various statistical tools.	02
2.	Implementation of Data & Descriptive Statistics-1.	02
3.	Implementation of Data & Descriptive Statistics-2.	02
4.	Implementation of Data & Descriptive Statistics-3.	02
5.	Implementation &Exploratory Data Analysis.	02
6.	Correlation Analysis	02
7.	Introduction to Probability	02
8.	Discrete and Continuous Probability Distribution-1.	02
9.	Discrete and Continuous Probability Distribution-2.	02
10.	Discrete and Continuous Probability Distribution-3.	06
11.	Testing of Hypothesis	06
	TOTAL	**30**

Text Book(s):

Title	Author/s	Publication
Basic Statistics	B L AGRAWAL	New Age International

Reference Book(s):

Title	Author/s	Publication
Understandable Statistics Concepts and Methods	Charles Henry Brase Corrinne PellilloBrase	Houghton Mifflin Company

Web Material Link(s):

- https://onlinecourses.nptel.ac.in/noc25_ma04/preview?
- https://onlinecourses.swayam2.ac.in/nou24_cm19/preview?

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests, each of 30 marks and 1 hour of duration and average of the same will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the Course Coordinator.

- End Semester Examination consists of 60 marks.

Tutorial:

- Continuous Evaluation consists of the performance of tutorial which will be evaluated out of 10 marks for each tutorial and average of the same will be converted to 50 marks.
- Assignment consists of 20 marks.
- Internal viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICSH1020	STATISTICS
CO 1	Elaborate analysis of categorial data and quantitative data.
CO 2	Examine the box plot for real data and able to find the outliers.
CO 3	Adapt the knowledge of various probability distribution and their applications in mathematical models, sport strategies and insurance.
CO 4	Adapt the knowledge of various probability distribution and their applications in insurance, banking and sentiment analysis.

Mapping of CO with PO

ICSH1020	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	3	1	2				1
CO 2	3	2	1	2				1
CO 3	3	2	1	2				1
CO 4	3	2	1	2				1

Mapping of CO with PSO

ICSH1020	PSO1	PSO2	PSO3
CO 1	2	2	1
CO 2	1	1	1
CO 3	2	1	1
CO 4	1	2	1

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to Data & Descriptive Statistics	1,2,3,5
2	Exploratory Data Analysis	1,2,3,4
3	Correlation Analysis	2,3,4,5
4	Introduction to Probability	2,3,5
5	Discrete & Continuous Probability Distribution	1,2,3,4
6	Testing of Hypothesis	1,2,3,4

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Application

Course Code: ICCA1021
Course Name: Advanced Web Designing
Prerequisite Course(s): ICCA1010

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	04	-	05	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
Aim to integrate frontend, backend, and database programming concepts in details to prepare students for real-world full-stack development.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Refreshing Java Script and CSS** CSS syntax, benefits, Responsive design, Bootstrap introduction, Java Script syntax, Java script inbuilt objects, Error handling and event handling, DOM, Asynchronous Programming.	07	10
2.	**Introduction to XML, AJAX and ES6:** Working with Basics of XML, Implementing Advanced Features of XML, Converting XML Documents in Other Format, Overview of AJAX, Consuming Web Services Using AJAX, Working with jQuery, Introduction to ES6,ES6 const and let, ES6 arrow functions, ES6 classes,.	08	20
3.	**React Component and Elements:** Introduction to React, Introduction to JSX, Lists and functional component in React, React DOM, Property validation, validating props with create class, Default props, Custom property validation, React state management -Introducing component state, Initializing state from properties, State within the component tree, Passing properties down the component tree, Passing data back up the component tree; Style component, dynamic style, Setting styles and class name component dynamically, Radium for media queries .	08	20
4.	**Introduction to Node.js**	07	15

	Installing Node.js, Executing Node.js scripts, Types of variables, Functions in Node.js, Default values, Closures, Exact equality, Modules, this keyword, Prototype, Node package manager; Popular node package manager modules - Handling command line arguments, Handling date/time using moment.		
5.	**Node JS in details** Events and Event Loop, timers, Error Handling, Buffers, Streams, Work with File System, Networking with Node (TCP, UDP and HTTP clients and servers), Web Module, Debugging, Node JS REST API, Sessions and Cookies, Design patterns, caching, scalability	07	20
6.	**Database Programming with Node JS and MongoDB** Basics of MongoDB, Data types, Connect Node JS with MongoDB, Operations on data (Insert, Find, Query, Sort, Delete, Update) using Node JS	08	15
	Total	45	100

List of Practical:

Sr. No	Name of Practical	Hours
1.	Create a responsive webpage with a navbar, carousel, and footer using Bootstrap. Ensure the layout adjusts for mobile and desktop views.	6
2.	Build a dynamic to-do list application where users can add, edit, and delete tasks. Use JavaScript for DOM manipulation and event handling.	6
3.	Develop a webpage that fetches and displays data (e.g., user profiles) from a public API (like GitHub API) using fetch() or axios.	6
4.	Create a React application that displays a list of products. Use functional components, JSX, and props to render product details dynamically.	6
5.	Develop a React app for a counter with increment, decrement, and reset buttons. Implement state management within components.	6
6.	Build a basic REST API in Node.js for managing a library system (CRUD operations for books).	10
7.	Connect a Node.js application to a MongoDB database. Perform CRUD operations (Insert, Read, Update, Delete) on a "Students" collection.	10
8.	Develop a full-stack web application (e.g., a task management system) with the following features: • **Frontend:** React for UI. • **Backend:** Node.js REST API. • **Database:** MongoDB for storing task details.	10
	TOTAL	**60**

Text Book (s):

Title	Author/s	Publication
The Road to Learn React Your Journey to Master Plain Yet Pragmatic React.js	Robin Wieruch	Zaccheus Entertainment

HTML 5 Black Book: Covers CSS3, Javasvript, XML, XHTML, AJAX, PHP and jQuery.	DT Editorial Services	Dreamtech Press

Reference Book (s):

Title	Author/s	Publication
Beginning Node.js	Basarat Syed	Apress
Practical Node.js: Building Real-World Scalable Web Apps	AzatMardan	Apress
Learning React: Functional Web Development with React and Redux	Alex Banks, Eve Porcello	O'Reilly Media, Inc.
Advanced Web Development with React: SSR and PWA with Next.js using React with advanced concepts (English Edition)	Mehul Mohan	BPB Publication.

Web Material Link(s):

- https://react.dev/
- https://getbootstrap.com/docs/5.3/getting-started/introduction/
- https://nodejs.org/docs/latest/api/

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICCA1021	Advanced Web Designing
CO 1	Apply modern CSS and JavaScript for responsive, event-driven web interfaces.
CO 2	Utilize ES6 features and React for developing single-page applications.
CO 3	Manage React components, state, props, and dynamic styling.
CO 4	Execute Node.js scripts, manage modules, and handle command-line arguments.
CO 5	Design scalable Node.js applications with REST APIs, sessions, and file systems.

Mapping of CO with PO

ICCA1021	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	3	3	2	3			2
CO 2	3	3	3		3			2
CO 3	3	3	3		3			2
CO 4	3	3	2	3	3			2
CO 5	3	3	3	3	3			2

Mapping of CO with PSO

ICCA1021	PSO1	PSO2	PSO3
CO 1	1	2	3
CO 2	1		3
CO 3	1	3	2
CO 4	1		
CO 5	1	2	3

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Refreshing Java Script and CSS	1,2
2	Introduction to React and ES6	2
3	React Component and Elements:	2.3
4	Introduction to Node.js	1,2
5	Node JS in details	2,3,4,6
6	Database Programming with Node JS and MongoDB	3,4,5,6

P P Savani University
School of Engineering
Institute of Computer Science and Application
Department of Information Technology

Course Code: ICIT1020
Course Name: Digital Marketing
Prerequisite Course(s): --

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	04	-	05	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- equip students with a comprehensive understanding of digital marketing concepts
- develop practical skills in using digital marketing platforms
- expose students to emerging trends and technologies

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Introduction to Digital Marketing** Definition and Importance of Digital Marketing. Traditional Marketing vs Digital Marketing Current Trends and Career Opportunities in Digital Marketing	3	6
2.	**Search Engine Optimization (SEO) & Web Analytics** Search Engine Optimization Fundamentals. Keywords and SEO Content Plan. Writing SEO Content. On-site & Off-site SEO. Optimize Organic Search Ranking. Google Analytics Tools. Web Analytics Tools	8	20
3.	**Display Marketing** Display Ads – Concepts and Types. Buying Models (CPC, CPM, CPL, CPA). Targeting Display Ads. Programmable Digital Marketing. Analytical Tools. YouTube Marketing.	6	12
4.	**Search Engine Marketing (SEM)** Pay-Per-Click (PPC) Advertising. Understanding Quality Score and Ad Rank Budget Planning and Bidding Strategies. Social Media Advertising (Facebook Ads, Instagram Ads, LinkedIn Ads). Analytics for Social Media Campaigns	6	12
5.	**Email Marketing and Content Marketing** Email Marketing Basics and Best Practices. Tools for Email Campaigns. Content Creation for Social Media. Content Marketing	7	15

	Strategies. Blog Writing and Content Optimization.		
6.	**Mobile Marketing** Mobile Advertising Concepts. Forms of Mobile Marketing & Features. Mobile Campaign Development. Mobile Advertising Analytics. Mobile-Friendly Content Strategies. Google Analytics & Google AdWords	8	20
7.	**Emerging Trends in Digital Marketing** Artificial Intelligence and Automation in Marketing. Voice Search Optimization. Introduction to Affiliate Marketing. Overview of Influencer Marketing. Data-Driven Marketing. Social Commerce.	7	15
	TOTAL	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Practical Task: Develop a digital marketing strategy for a fictional business, outlining goals, target audience, and key performance indicators (KPIs).	04
2.	Compare a traditional marketing campaign with a digital marketing campaign. Analyze the differences in reach, cost, and audience engagement.	04
3.	Conduct a SWOT analysis for a business considering digital marketing. Identify potential opportunities and challenges in the digital landscape.	04
4.	Research and present a case study on a brand that successfully leveraged current digital trends (e.g., influencer marketing, user-generated content).	04
5.	Explore and discuss the ethical considerations of digital marketing. Create guidelines for responsible and ethical marketing practices.	04
6.	Develop a timeline showcasing the evolution of digital marketing tools and technologies. Highlight key milestones and their impact.	04
7.	Analyze how technological advancements drive changes in consumer behavior. Propose strategies for adapting marketing efforts to these changes.	04
8.	Apply a digital marketing framework (e.g., RACE model) to a real-world scenario. Develop a campaign using the framework and measure its effectiveness.	04
9.	Set up a Google Ads campaign for a business. Choose relevant keywords, set a budget, and monitor campaign performance.	04
10.	Optimize a pay-per-click (PPC) campaign by adjusting bids, improving ad relevance, and implementing ad extensions.	04
11.	Analyze the performance of a YouTube marketing campaign using YouTube Analytics. Identify trends and areas for optimization.	04
12.	Develop a location-based marketing strategy. Consider features like push notifications or mobile wallet integration to enhance user experience.	04
13.	Use mobile advertising analytics tools to track user engagement and conversion rates. Adjust the campaign based on the insights gained.	04
14.	Implement tracking codes and tags for a website using Google Analytics. Verify the accuracy of data collection.	04

15.	Analyze a marketing campaign using a multichannel attribution model. Determine the contribution of each channel to conversions.	04
	TOTAL	**60**

Text Book (s):

Title	Author/s	Publication
Digital Marketing	Seema Gupta	Mc-GrawHill,1st Edition – 2017

Reference Book (s):

Title	Author/s	Publication
Digital Marketing: A Practical Approach	Alan Charlesworth	Routledge-2023
Fundamentals of Digital Marketing	Puneet Singh Bhatia	Pearson 1st Edition-2017
The Art of Digital Marketing	Ian Dodson	Wiley

Web Material Link(s):

- https://neilpatel.com/what-is-digital-marketing/
- https://www.investopedia.com/terms/d/digital-marketing.asp

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests of 30 marks and 1 hour of duration and average of the same will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by Course Coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks during End Semester Examination.
- Viva/Oral performance consists of 30 marks during End Semester Examination.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICIT1020	DIGITAL MARKETING
CO 1	Gain a deep understanding of digital marketing concepts, tools, and strategies.
CO 2	Develop the skills to create and manage effective digital marketing campaigns across various platforms.
CO 3	Learn to analyze and optimize marketing performance using web analytics and SEO techniques.
CO 4	Able to design mobile-friendly and location-based marketing strategies for enhanced user engagement.

CO 5	Equipped to adapt marketing strategies based on emerging trends such as AI, voice search, and influencer marketing.

Mapping of CO with PO

ICIT1020	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2				2		
CO 2	3	3	2	2				2
CO 3			3	2		1		2
CO 4	1		2	3		2		
CO 5	2	3	3	3	2			

Mapping of CO with PSO

ICIT1020	PSO1	PSO2	PSO3
CO 1	3		
CO 2	3	2	
CO 3		3	
CO 4		2	2
CO 5	2		3

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to Digital Marketing	1,2
2	Search Engine Optimization (SEO) & Web Analytics	3,4
3	Display Marketing	3,4
4	Search Engine Marketing (SEM)	3,4
5	Email Marketing and Content Marketing	3,5
6	Mobile Marketing	3,4
7	Emerging Trends in Digital Marketing	2,5,6

P P Savani University
School of Engineering
Institute of Computer Science and Application
Department of Information Technology

Course Code: ICIT1031
Course Name: Object Oriented Programming with C++
Prerequisite Course(s): ICIT1010

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	04	-	05	40	60	40	60	0	0	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- define & describe the basic concepts of the Object-Oriented Programming Paradigm.
- understand functions in C++ and the different types of Constructors in C++.
- understand on Operator Overloading, Inheritance

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Basic concepts of Object-Oriented Programming** Introduction OOP, Procedural Vs. Object Oriented Programming, Principles of OOP, Benefits and applications of OOP	3	10
2.	**Introduction to C++** Overview, Program structure, namespace, identifiers, variables, constants, enum, operators, typecasting, control structures	4	13
3.	**Functions** Simple functions, Call and Return by reference, Inline functions, Macro Vs. Inline functions, Overloading of functions, default arguments, friend functions, virtual functions	6	15
4.	**Object and classes** Basics of object and class in C++, Private and public members, static data and function members, constructors and their types, destructors, operator overloading, type conversion	5	15
5.	**Inheritance** Concept of Inheritance, types of inheritance: single, multiple, multilevel, hierarchical, hybrid, protected members, overriding, virtual base class,	8	13
6.	**Polymorphism** Pointers in C++, Pointes and Objects, this pointer, virtual and pure virtual functions, Implementing polymorphism	5	10
7.	**I/O and File management**	7	12

	Concept of streams, cin and cout objects, C++ stream classes, Unformatted and formatted I/O, manipulators, File stream, C++ File stream classes, File management functions, File modes, Binary and random Files		
8.	**Templates, Exceptions and STL** What is template? function templates and class templates, Introduction to exception, try-catch throw, multiple catch, catch all, rethrowing exception, implementing user defined exceptions, Overview and use of Standard Template Library	7	12
	TOTAL	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Introduction to C++ basic input/output functions, library files.	4
2.	Implementation of C++ programs with classes and objects.	4
3.	Implement C++ programs to demonstrate use of data types, tokens and constants.	4
4.	Implementation of C++ programs to demonstrate dynamic initialization of Variables, Reference Variables, Operators in C++, Scope Resolution Operator.	4
5.	Implementation of C++ programs to demonstrate use of member referencing, operators – Memory Management Operators – manipulators.	4
6.	Implementation of C++ programs for call by reference and return by reference	4
7.	Implement of C++ programs for use of inline function	4
8.	Implementation of C++ programs to demonstrate use of function overloading.	4
9.	Implementation of C++ programs to demonstrate use of virtual function.	4
10.	Implementation of C++ programs to demonstrate static data members, friend function.	4
11.	Implementation of C++ programs to demonstrate constructors and destructors.	4
12.	Implementation of file handling operations.	6
13.	Implement a program for type conversion between objects and basic types.	4
14.	Create a program demonstrating exception handling using try-catch-throw.	6
	TOTAL	**60**

Text Book (s):

Title	Author/s	Publication
Object Oriented Programming with C++	E Balagurusamy	McGraw Hill Education (India) Private Limited

Reference Book (s):

Title	Author/s	Publication
C++: The Complete Reference	Herbert Schildt	McGraw-Hill Education

Web Material Link(s):

- **https://www.tutorialspoint.com/cplusplus/index.htm**
- **https://www.w3schools.com/CPP/default.asp**
- **https://www.javatpoint.com/cpp-tutorial**

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 Hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practicals, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICIT1031	OBJECT-ORIENTED PROGRAMMING WITH C++
CO1	Use advanced features like templates and exceptions to make programs, standard template library for faster development.
CO2	Use features of c++ like type conversion, inheritance, polymorphism, i/o streams and files to develop programs for real life problems.
CO3	Develop the applications using object-oriented programming with c++.
CO4	Use advance features like temples and exception to make programs supporting reusability and template library for faster development.
CO5	Develop the applications using object-oriented programming with c++.

Mapping of CO with PO

ICIT1031	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	2			1	2	1	1	1
CO 2		1	2	2	1	2	1	2
CO 3	2	1	1	1	2	2	2	1
CO 4	1	2	2	1	1	1	2	1
CO 5	2	3	2	2	3	3	2	2

Mapping of CO with PSO

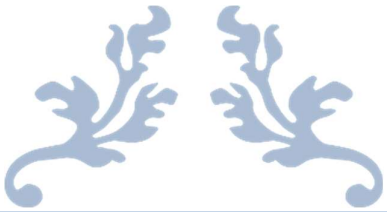
ICIT1031	PSO1	PSO2	PSO3
CO 1	3	3	2
CO 2	3	3	2
CO 3	3	3	3
CO 4	3	3	2

CO 5	3	3	3

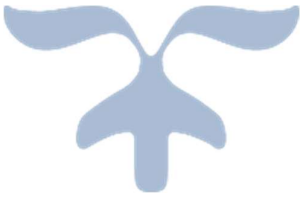
Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Basic concepts of Object-Oriented Programming	1,2
2	Introduction to C++	2,3
3	Function	2,3.5.6
4	Objects and Classes	1,2,3
5	Inheritance	1,2,6
6	polymorphism	2,3,4,6
7	I/O and File Management	1,2,5
8	Templates, Exceptions and STL	1,2,3,6



SECOND YEAR B.C.A.



P P SAVANI UNIVERSITY														
SCHOOL OF ENGINEERING														
INSTITUTE OF COMPUTER SCIENCE AND APPLICATION														
TEACHING & EXAMINATION SCHEME FOR B.C.A. PROGRAMME with Honors (AY: 2025-26)														
Sem	**Course Code**	**Course Title**	**Offered By**	**Teaching Scheme**					**Examination Scheme**						
				Contact Hours				**Credit**	**Theory**		**Practical**		**Tutorial**		**Total**
				Theory	**Practical**	**Tutorial**	**Total**		**CE**	**ESE**	**CE**	**ESE**	**CE**	**ESE**	
3	ICCA2010	Database Management System	CA	3	2	0	5	4	40	60	40	60	0	0	200
	ICCS2010	Programming with Python	CS	3	2	0	5	4	40	60	40	60	0	0	200
	ICIT2010	PHP Studio	IT	2	2	0	4	3	40	60	40	60	0	0	200
	ICCA2020	Cloud Computing	CA	3	2	0	5	4	40	60	40	60	0	0	200
	ICCS2020	Data Structures	CS	3	4	0	7	5	40	60	40	60	0	0	200
	CLSC2020	IPDC -I	CLSC	2	0	0	2	2	100	0	0	0	0	0	100
						Total	**28**	**22**							**1100**
4	ICCA2031	Relational Database Management System	CA	3	4	0	7	5	40	60	40	60	0	0	200
	ICCA2041	Gaming with Python	CA	3	2	0	5	4	40	60	40	60	0	0	200
	ICIT2040	Operating Systems	IT	3	2	0	5	4	40	60	40	60	0	0	200
	ICCA2050	Object oriented Programming with JAVA	CA	3	4	0	7	5	40	60	40	60	0	0	200
	ICCA2060	Online Course-I	CA	3	0	0	3	3	100	0	0	0	0	0	100
	CLSC2030	IPDC-II	CLSC	2	0	0	2	2	100	0	0	0	0	0	100
							Total	**29**	**23**						**1000**

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Application

Course Code: ICCA2010
Course Name: Database Management System
Prerequisite Course(s): -

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	02	-	04	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- Learn the fundamental theory and practice necessary for conceptual and logical database design and DBMS development.
- Develop a thorough competency in Structured Query Language (SQL), specifically its processing, optimization, and application.
- Explore the core architectural principles and design considerations of Relational Database Management Systems (RDBMS) back-ends along with few industrial aspects.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Introduction to DBMS and Data Models:** File Organization vs. DBMS, Purpose & Application of DBMS, Data Independence, Architecture, Users & Administrators. Data Models. Relational Model: Structure, Domains, Relations, Relational Algebra (Operators & Queries).	7	16
2.	**Relational Database Concepts and Integrity:** Components of DBMS, Query Languages (DDL, DML, TCL), Data Independence, Keys (Super, Candidate, Primary, Foreign, etc.), Integrity Constraints (Domain, Referential, Entity).	8	18
3.	**SQL Query Fundamentals and Functions:** Basic SELECT, WHERE, ORDER BY. IN operator, Aggregate functions, Built-in functions (numeric, date, string). Set operations. INSERT, UPDATE & DELETE queries.	6	13
4.	**Advanced SQL:** Sub-queries and Correlated Sub-queries. Joins (Inner, Outer, Self), Exist, Any, All. Views and its types.	7	16
5.	**Entity-Relationship (E-R) Modelling:** E-R Model: Basic concepts, Design process, Constraints, Keys, Design issues, E-R diagrams, Weak Entity Sets. Extended E-R Features: Generalization, Specialization, Aggregation, Reduction to Database Schema.	7	16
6.	**Relational Database Design and Normalization:** Functional Dependency & its types, Consequences of Bad Design (Anomalies), Normalization Need. Normal Forms: First (1NF), Second (2NF), Third (3NF), BCNF.	6	13
7.	**Transaction Management & Concurrency Control**	2	4

	Definition of transaction, Transaction states, TCL Commands, ACID properties, Concurrency Problems, Concurrency Control Techniques		
8.	**NoSQL and Distributed Databases:** MongoDB/Cassandra Architecture, CAP Theorem, Data consistency models, Use-cases in modern web applications.	2	4
	Total	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Create and manage databases using DDL commands (CREATE, ALTER, DROP) and explore data types.	2
2.	Implement integrity constraints including primary key, foreign key, unique, check, and not null.	2
3.	Implement basic SQL queries using SELECT, WHERE, ORDER BY, BETWEEN, LIKE, IN.	2
4.	Apply aggregate and built-in functions (COUNT, SUM, AVG, MIN, MAX, string, date functions).	2
5.	Perform set operations such as UNION, INTERSECT, and MINUS on sample datasets.	2
6.	Execute DML operations using INSERT, UPDATE, DELETE with realistic sample tables.	2
7.	Write SQL queries using different types of JOINs (Inner, Left, Right, Full, Self Join).	2
8.	Implement subqueries and correlated subqueries for complex data retrieval.	4
9.	Create and manage views (simple view, complex view, updatable/non-updatable views).	2
10.	Design an Entity-Relationship (ER) diagram for a mini case study and convert it into relational schema.	2
11.	Identify functional dependencies in a dataset and perform normalization up to 3NF/BCNF.	2
12.	Perform Transaction Control (TCL) using COMMIT, ROLLBACK, SAVEPOINT and observe transaction behavior.	2
13.	Simulate concurrency issues like lost update, dirty read, and non-repeatable read using transaction sessions.	2
14.	Work with MongoDB/Cassandra: Create collections, insert documents, and run basic NoSQL queries.	2
15.	Demonstrate CAP theorem concepts by configuring a small distributed or simulated NoSQL setup and observing consistency–availability trade-offs.	4
	TOTAL	**30**

Text Book (s):

Title	Author/s	Publication
PL/SQL-The Programming Language of Oracle	Ivan Bayross	BPB Publications

Reference Book (s):

Title	Author/s	Publication
Database System Concept	Abraham Silberschatz, Henry F. Korth, S. Sudarshan	McGraw Hill
An Introduction to Database System	CJ Date	Addition-Wesley

Fundamental of Database System	R. Elmasri and S.B Navathe	Benjamin/Cummin g
Oracle: The Complete Reference	George Koch, Kevin Loney	TMH/oracle press

Web Material Link(s):

- https://www.geeksforgeeks.org/dbms/
- https://onlinecourses.nptel.ac.in/noc18 cs15
- https://www.tutorialcup.com/dbms

Course Evaluation:
Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):
After the completion of the course, the following course outcomes will be able to:

ICCA2010	Database Management System
CO 1	Understand fundamental DBMS concepts, data models, architectures, and relational algebra operations.
CO 2	Apply relational database concepts, keys, and integrity constraints to design consistent database structures.
CO 3	Construct SQL queries using basic, advanced, and functional operations for data retrieval and manipulation.
CO 4	Design ER models and normalize relational schemas using functional dependencies and normal forms.
CO 5	Explain transaction management, concurrency control, and modern database systems including NoSQL and distributed databases.

Mapping of CO with PO

ICCA2010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	2	1	2	1		1
CO 2	3	2	1	2				
CO 3	3	2	3	1	2	1		2
CO 4	2	3	3	2	3	2	1	
CO 5	3	2	3	1	3	1		2

Mapping of CO with PSO

ICCA2010	PSO1	PSO2	PSO3
CO 1	2	1	2
CO 2	2	1	2
CO 3	2	1	2
CO 4	3	2	2
CO 5	3	2	1

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to DBMS and Data Models	1,2
2	Relational Database Concepts and Integrity	1,2,3
3	SQL Query Fundamentals and Functions	1,2,3,4,5
4	Advanced SQL	2,3,4,5
5	Entity-Relationship (E-R) Modelling	2,3,4,5
6	Relational Database Design and Normalization	2,3,4,5
7	Transaction Management & Concurrency Control	2,3,4,5
8	NoSQL and Distributed Databases	4,5,6

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Science

Course Code: ICCS2010
Course Name: Programming with Python
Prerequisite Course(s):

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	02	-	04	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- Understand the syntax, semantics, and core concepts of Python programming.
- Develop Python programs using data types, control structures, functions, and modules.
- Implement object-oriented programming concepts and exception handling.
- Apply Python libraries for data analysis and visualization.

Co**urse Content:**

Module No.	Content	Hours	Weightage in %
1.	**Introduction to Python:** Overview of Python, Installation, Python IDEs, Syntax, Variables, Data Types, Operators, Type Conversion, Input/Output Functions	06	10
2.	**Control Structures:** Decision-making (if, if-else, nested if), Loops (for, while), Loop control statements, Range function, Iterations	05	10
3.	**Strings and Lists:** String operations, slicing, string methods, list operations, list comprehension, nested lists	05	10
4.	**Tuples, Sets, and Dictionaries:** Tuple operations, immutability, set operations, dictionary methods, nested dictionary, comprehension	05	10
5.	**Functions and Modules:** Function definition, parameters, return values, variable scope, recursion, lambda functions, importing modules, math, random, datetime	06	15
6.	**File Handling and Exception Handling:** File read/write operations, file modes, handling text and binary files, exceptions, try-except-else-finally blocks	05	10
7.	**Object-Oriented Programming in Python**: Classes and objects, constructors, inheritance, polymorphism, encapsulation, method overriding	07	15
8.	**Python Libraries and Applications:** NumPy, Pandas, Matplotlib basics, data visualization, Introduction to machine learning applications. scikit, scipy-learn basics.	06	20
	TOTAL	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Write a Python program to demonstrate variable types and basic operators.	02
2.	Implement decision-making and looping constructs in Python.	02
3.	Demonstrate string and list manipulations.	04
4.	Implement programs using tuples, sets, and dictionaries.	04
5.	Write programs using user-defined functions and lambda expressions.	02
6.	Demonstrate importing and using Python standard modules.	02
7.	Perform file read/write operations.	02
8.	Demonstrate exception handling in Python.	04
9.	Create classes and objects to demonstrate OOP concepts.	04
10.	Perform data analysis using NumPy and Pandas.	04
	TOTAL	**30**

Text Book(s):

Title	Author/s	Publication
Python Programming: A Modern Approach	Vamsi Kurama	Pearson Education
Python Programming; A Modular Approach	Taneja, Sheetal	Pearson

Reference Book(s):

Title	Author/s	Publication
Learning Python	Mark Lutz	O'Reilly Media
Python for Everybody	Charles Severance	Coursera Press
Think Python	Allen B. Downey	Green Tea Press

Web Material Link(s)

- https://onlinecourses.nptel.ac.in/noc24_cs57/preview
- https://nptel.ac.in/courses/106/106/106106182/
- https://www.coursera.org/learn/python

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 Hour of duration and average of the same will be converted to 30 marks.
- Faculty Evaluation consists of 10 marks as per the guidelines provided by Course Coordinator.
- End Semester Examination consists of 60 marks.

Practical/Tutorial:

- Continuous Evaluation consists of performance of Practical/Tutorial which should be evaluated out of 10 for each practical and average of the same will be converted to 20 Marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/drawing/test of 30 marks during End Semester Exam.
- Viva/Oral performance of 30 marks during End Semester Exam.

Course Outcome(s):

After completion of the course, the student will be able to

ICCS2010	Programming with Python
CO1	Understand Python basics, syntax, and fundamental programming constructs.
CO2	Implement programs using functions, loops, and control structures effectively.
CO3	Apply file handling, exception management, and data structures (lists, tuples, sets, dictionaries).
CO4	Develop programs using object-oriented programming and modules in Python.
CO5	Analyze and solve real-world problems using Python libraries like NumPy, Pandas, and Matplotlib.

Mapping of CO with PO

ICCS2010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1					
CO2	2	3	2	2	2	3	1	
CO3	2	3	2	3	2	2	1	2
CO4	3	2	3	2	3		1	
CO5	2	2	1	2	3	2	3	2

Mapping of CO with PSO

ICCS2010	PSO1	PSO2	PSO3
CO1	2	1	
CO2	2	2	2
CO3	3	3	3
CO4	3	3	2
CO5	2	2	3

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1.	Introduction to Python	1,2
2.	Control Structures	2,3
3.	Strings and Lists	2,4
4.	Tuples, Sets, and Dictionaries	3,4,5
5.	Functions and Modules	2,3,4
6.	File Handling and Exception Handling	2,3,4
7.	Object-Oriented Programming in Python	3,4,6
8.	Python Libraries and Applications	4,5,6

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Information Technology

Course Code: ICIT2010
Course Name: PHP Studio
Prerequisite Course(s): -

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
02	02	-	03	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To develop students' skills in server-side programming using PHP, integrating frontend, backend, and database concepts for building dynamic and interactive web applications..

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Introduction to PHP:** What is PHP?, Loosely typed language vs Strongly type language, Basic PHP Syntax, Comments in PHP.	03	05
2.	**Constants, Variables, data types and Operators and Expression:** Module keyword, Constants and Variables, Datatypes-Declaration and Initialization, Basic input and output operations, Symbolic constants, Arithmetic Operator, Increment and Decrement operator, Assignment Operator, String Operator	06	20
3.	**Conditional Statement and Branching:** Decision Making & branching: Decision making with If & If... else statements, If - Else statements (Nested Ladder) and Looping: The while statement. The break statement & The Do. While loop, The FOR loop, FOREACH, break and continue	05	20
4.	**User-Defined Functions , Array & Strings:** prototypes, definition of function, parameters, parameter passing and calling, A function. Introduction to array, Numeric Array and in-built string functions. PHP Forms: S_GET and $_POST function	05	20
5.	**PHP & MySQL:** Introduction to PHP MyAdmin, connection to MySQL server from PHP, execution of MySQL queries from PHP, receiving data from database server and processing it on webserver using PHP.	06	20
6.	**Web Sockets:**	05	15

	Introduction to Web sockets, Web socket URIs, Web socket APIs, Opening Handshake, Data Framing, Sending and Receiving Data, Closing the Connections, Error Handling, Web socket Security		
	Total	30	100

List of Practical:

Sr. No	Name of Practical	Hours
1.	Introduction to PHP. Installation of XAMPP Server and write a PHP Script to display Welcome message	02
2.	HelloWorld Example, finding errors present in the program, Insert Comments in Program, ECHO and PRINT statements in PHP, PHP Variable Example, Global and locally-scoped variables -Example, Constant string Example, PHP Example to calculate the area of the circle	04
3.	Example on Arithmetic Operators, Increment and Decrement Operators, Assignment Operators and String Operators	02
4.	Example on Conditional Statements (if, if...else Statement, if...else if...else and Switch)	02
5.	Example on branching Statements (For loop, Declaring multiple variables in for loop, While loop and Do-While loop), Example on break and Continue Statement	04
6.	User Defined Function Example (How to Add parameters and How to Return values?).Date() and time() function in PHP-Example	04
7.	Array in PHP, Numeric array in PHP-Example Associative array in PHP - Example Loop through an Associative	02
8.	PHP Forms The $_GET Function-Example, The $_POST Function-Example PHP Global Variables-Superglobals $_GOBALS Variables	02
9.	Develop a small project in with database connectivity	06
	TOTAL	**30**

Text Book (s):

Title	Author/s	Publication
PHP: The Complete Reference	Steven Holzner	Tata McGraw Hill

Reference Book (s):

Title	Author/s	Publication
Learning PHP, MySQL	Michele Davis, Jon Phillips	'O'riley Press
Web Technologies Black Book	Kogent Learning Solutions Inc	Dreamtech Press

Web Material Link(s):

- https://onlinecourses.swayam2.ac.in/aic20_sp32/preview
- https://www.w3schools.com/php/
- https://www.geeksforgeeks.org/php/php-tutorial/

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICIT2010	PHP Studio
CO 1	Understand the basic syntax, structure, and features of PHP as a server-side scripting language.
CO 2	Apply variables, constants, data types, and operators to perform operations and manage data efficiently.
CO 3	Implement decision-making and looping constructs to control program flow in PHP scripts.
CO 4	Develop dynamic web applications using user-defined functions, arrays, strings, and PHP forms with GET and POST methods.
CO 5	Integrate PHP with MySQL and Web Sockets to create interactive, database-driven, and real-time web applications.

Mapping of CO with PO

ICIT2010	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2						
CO 2	2	3						
CO 3	2	2						1
CO 4	2	2	3		2		2	1
CO 5	3	3	3	3	3	2	2	2

Mapping of CO with PSO

ICIT2010	PSO1	PSO2	PSO3
CO 1	3	2	
CO 2		2	2
CO 3	2		
CO 4		3	3
CO 5	3	2	3

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	**Content**	**RBT Level**
1	Introduction to PHP	1,2,3
2	Constants, Variables, data types and Operators and Expression	1.2.3.4
3	Conditional Statement and Branching:	1,2,3
4	User-Defined Functions, Arrays and Strings	2,3,4
5	PHP & MySQL	2,3,4,5,6
6	Web Sockets	2,3,4,5,6

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Application

Course Code: ICCA2020
Course Name: Cloud Computing
Prerequisite Course(s): --

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	02	-	04	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- To establish the foundational concepts and architectural frameworks of distributed computing paradigms leading to Cloud Computing.
- To impart practical knowledge in analyzing, deploying, and managing various service and deployment models (IaaS, PaaS, SaaS).
- To explore the critical issues of security, economics, and service governance necessary for successful enterprise cloud adoption.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Computing Paradigms and Overview:** Overview of Computing Paradigm, Recent trends: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Introduction to Cloud Computing, History, Benefits and Limitations.	4	9
2.	**Cloud Architecture and Deployment Models:** Comparison with traditional computing architecture (client/server), Deployment Models: Public, Private, Hybrid, Community cloud. Case study of NIST architecture.	6	13
3.	**Service Models: IaaS, PaaS, and SaaS:** Services provided at various levels, detailed study of Service Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), How Cloud Computing Works.	6	13
4.	**Enabling Technology: Virtualization:** Concept of Virtualization, Types (Hardware, OS, Application), Virtual Machines, Hypervisors, Virtual Machine Life Cycle, Migration, and related concepts (essential for IaaS).	8	16
5.	**Service Management and Cloud Economics:**	5	12

	Service Level Agreements (SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.		
6.	**Security and Access Control:** Infrastructure security, Network level security, Host level security, Application-level security. Identity, Access Management, Access control, Trust, and Risk Authentication in cloud computing.	6	13
7.	**Data Management and Jurisdictional Issues:** Data security and storage, Jurisdictional issues (data location), Data lifecycle in cloud, Data encryption strategies.	5	12
8.	**Industry Case Studies and Trends:** Case study of Service models using Google App Engine, Microsoft Azure, Amazon EC2, Eucalyptus. Review of Serverless and Edge Computing trends.	5	12
	Total	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Study the NIST architecture and prepare a report on the differences between Public, Private, and Hybrid Clouds. Set up a basic VPN for secure remote access.	2
2.	Study different types of Hypervisors (Type 1 and Type 2). Install a Type 2 Hypervisor (e.g., VirtualBox) and create a basic virtual machine (VM) instance.	2
3.	Create a Virtual Machine (VM) instance on a Public Cloud Platform (e.g., AWS EC2/Azure). Choose an appropriate OS and machine configuration.	2
4.	Study and install/configure Storage as a Service (e.g., Amazon S3 or Azure Blob Storage) and upload/retrieve objects.	2
5.	Study and implement a basic application (e.g., "Hello World" in Python/Node.js) on a Platform as a Service platform (e.g., Heroku or Google App Engine).	2
6.	Perform a comparative study (Mini Task) on the services offered by major cloud providers (AWS, Azure, GCP) based on their pricing and features.	2
7.	Choose a specific Cognitive Service (e.g., Azure Text Analytics or AWS Recognition) and develop a simple script to interact with the chosen service.	2
8.	Demonstrate User Management and Administrative features of a Cloud Management Console (e.g., creating groups, assigning basic roles).	2
9.	Implement Network Level Security by configuring a Security Group (Firewall rules) to restrict inbound/outbound traffic (e.g., only allowing SSH/RDP access).	2
10.	Implement a basic level of data encryption for a data directory or storage service using a platform utility or command-line tool.	2
11.	Configure a running Cloud Web Server (e.g., Apache/Nginx on a VM) to secure the data directory using configuration files.	2
12.	Study the concept of Containerization (e.g., Docker) and compare its resource usage and deployment model with traditional Virtual Machines.	2
13.	Analyze the billing dashboard of a cloud account to understand usage patterns, cost breakdowns, and identify potential areas for optimization.	2

14.	Prepare a detailed Case Study report on a specific service model implementation, such as using Google App Engine or another open system for cloud deployment.	2
15.	Deploy a simple web application (provided by the instructor) using either the IaaS or PaaS model, and document the process of scaling and monitoring the service.	2
	TOTAL	**30**

Text Book (s):

Title	Author/s	Publication
Cloud Computing: Principles and Paradigms	Rajkumar Buyya, James Broberg, Andrzej Goscinski	John Wiley & Sons, Inc.

Reference Book (s):

Title	Author/s	Publication
Cloud computing a practical approach	Anthony T.Velte, Toby J. Velte, Robert Elsenpeter	TATA McGraw-Hill
Cloud Computing: Concepts, Technology & Architecture	Thomas Erl, Ricardo Puttini, Zaigham Mahmood	Prentice Hall
Cloud Computing Bible	Barrie Sosinsky	Wiley Publishing

Web Material Link(s):

- https://nptel.ac.in/courses/106105167
- https://onlinecourses.nptel.ac.in/noc21_cs14/preview
- https://www.tutorialspoint.com/cloud_computing/index.htm
- https://www.geeksforgeeks.org/cloud-computing/cloud-computing-tutorial/

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICCA2020	Cloud Computing
CO 1	Analyze the technical and economic foundations of distributed and utility computing leading to the cloud paradigm.
CO 2	Apply the principles of virtualization and resource abstraction to configure and manage different Infrastructure as a Service (IaaS) environments.

CO 3	Differentiate and implement various service and deployment models (PaaS, SaaS) to develop and host scalable applications.
CO 4	Formulate comprehensive strategies for service governance, including managing SLAs, scaling, and cost optimization.
CO 5	Evaluate and integrate robust security, identity, and access control mechanisms to safeguard data and applications within a multi-tenant cloud environment.

Mapping of CO with PO

ICCA2020	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	2	3	1	2	1	1		
CO 2	3	3	3	2	3	2	1	
CO 3	3	2	3	2	3	2	1	
CO 4	2	3	2	2	2	2	3	1
CO 5	2	3	3	3	3	3	2	2

Mapping of CO with PSO

ICCA2020	PSO1	PSO2	PSO3
CO 1	1	2	
CO 2	2	3	1
CO 3	2	3	1
CO 4	1	2	1
CO 5	2	2	2

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Computing Paradigms and Overview	1,2
2	Cloud Architecture and Deployment Models	1,2
3	Service Models: IaaS, PaaS, and SaaS	1,2,3
4	Enabling Technology: Virtualization	2,3
5	Service Management and Cloud Economics	1,2,3,4,6
6	Security and Access Control	1,2,3,4,5,6
7	Data Management and Jurisdictional Issues	2,3,4,5,6
8	Industry Case Studies and Trends	1,2,4,6

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Science

Course Code: ICCS2020
Course Name: Data Structures
Prerequisite Course(s): -

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
3	4	-	5	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- Understand the basic concepts of data structures and algorithms.
- Learn how to represent and manipulate data using arrays, stacks, queues, linked lists, trees, and graphs.
- Develop problem-solving skills using data structure operations.
- Analyze the performance of different algorithms in terms of time and space complexity.
- Apply appropriate data structures in real-world problem-solving and software development.

Co**urse Content:**

Module No.	Content	Hours	Weightage in %
1.	**Introduction to Data Structures:** Basics of data organization, abstract data types (ADT), need and importance of data structures, algorithm analysis, time and space complexity, Big O notation	06	10
2.	**Arrays and Strings:** Representation, operations (insertion, deletion, traversal, searching, sorting), multidimensional arrays, applications	05	10
3.	**Stacks:** Definition and operations, implementation using arrays and linked lists, applications – infix, prefix, postfix expressions, recursion	06	10
4.	**Queues:** Concept, implementation using arrays and linked lists, circular queue, priority queue, deque, applications	05	10
5.	**Linked Lists:** Singly, doubly, and circular linked lists; operations (insertion, deletion, traversal, searching); applications	07	15
6.	**Trees:** Binary tree representation, traversal (inorder, preorder, postorder), binary search tree (BST), AVL tree basics, applications	06	15
7.	**Graphs:** Representation (adjacency matrix, list), graph traversal (BFS, DFS), shortest path algorithms, spanning tree concepts	05	15
8.	**Searching and Sorting Techniques:** Linear and binary search, bubble sort, selection sort, insertion sort, quick sort, merge sort, comparison and complexity analysis	05	15
	TOTAL	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Implement basic operations on arrays and strings.	02
2.	Implement stack operations using arrays and linked lists.	04
3.	Write a program to convert infix expression to postfix and evaluate it.	04
4.	Implement queue, circular queue, and priority queue.	04
5.	Develop programs for singly and doubly linked list operations.	02
6.	Implement binary tree traversals (inorder, preorder, postorder).	02
7.	Write a program for binary search tree insertion and deletion.	02
8.	Implement graph traversal algorithms (BFS and DFS).	04
9.	Write programs for searching and sorting algorithms.	02
10.	Mini Project: Implement a real-life data structure-based application.	04
	TOTAL	**30**

Text Book(s):

Title	Author/s	Publication
Data Structures Using C	Reema Thareja	Oxford University Press

Reference Book(s):

Title	Author/s	Publication
Fundamentals of Data Structures	Ellis Horowitz, Sartaj Sahni	Galgotia Publications
Data Structures Through C	Yashavant Kanetkar	BPB Publications
Data Structures and Algorithm Analysis in C	Mark Allen Weiss	Pearson Education

Web Material Link(s)

- https://nptel.ac.in/courses/106105085
- https://www.geeksforgeeks.org/data-structures/
- https://www.javatpoint.com/data-structure-tutorial
- https://www.coursera.org/learn/data-structures

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 Hour of duration and average of the same will be converted to 30 marks.
- Faculty Evaluation consists of 10 marks as per the guidelines provided by Course Coordinator.
- End Semester Examination consists of 60 marks.

Practical/Tutorial:

- Continuous Evaluation consists of performance of Practical/Tutorial which should be evaluated out of 10 for each practical and average of the same will be converted to 20 Marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/drawing/test of 30 marks during End Semester Exam.
- Viva/Oral performance of 30 marks during End Semester Exam.

Course Outcome(s):

After completion of the course, the student will be able to

ICCS2020	Data Structures
CO1	Understand the fundamentals and classifications of data structures.
CO2	Apply arrays, stacks, queues, and linked lists for data management.
CO3	Analyze and construct trees and graphs to solve computational problems.
CO4	Implement searching and sorting algorithms and evaluate their performance.
CO5	Design efficient programs using appropriate data structures for real-world applications.

Mapping of CO with PO

ICCS2020	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1				
CO2	2	3	2	2	2	1	1	2
CO3	2	3	2	3	2		1	
CO4	3	2	3	2	3	1	1	2
CO5	2	2	3	2	3	1	2	3

Mapping of CO with PSO

ICCS2020	PSO1	PSO2	PSO3
CO1	2	1	
CO2	2	2	2
CO3	3	3	3
CO4	3	3	2
CO5	2	2	3

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1.	Introduction to Data Structures	1,2,3
2.	Arrays and Strings	1,2,3,4
3.	Stacks	1,2,3,4
4.	Queues	1,2,3,4
5.	Linked Lists	2,3,4,5,6
6.	Trees	2,3,4,5
7.	Graphs	2,3,4,5
8.	Searching and Sorting	2,3,4,5,6

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Application

Course Code: ICCA2031
Course Name: Relational Database Management System
Prerequisite Course(s): - ICCA2010

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	04	-	05	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners:

- To establish competency in procedural programming methodologies for database applications, focusing on the design and implementation of modular, business-logic routines.
- To impart a functional understanding of transaction atomicity and control mechanisms, alongside advanced techniques for iterative data manipulation and exception handling.
- To analyze and apply event-driven automation features and persistent program objects to ensure both back-end process efficiency and structural integrity.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Introduction to Procedural Database Programming:** Introduction to PL/SQL (Definition & Block Structure) , Variables, Constants and Data Types, Assigning Values, User-Defined Records. Introduction to Stored Procedures and Functions.	5	10
2.	**Flow Control Structures:** Conditional Statements: IF...THEN, IF...Else, multiple conditions, Nested IF statements, CASE statements. Iterative Statements: Loop. End Loop, For Loop, While Loop, EXIT Loop, Continue.	8	11
3.	**Data Retrieval Techniques in Procedural Code:** Overview of Cursors , Types of Cursors (Implicit/Explicit), Cursor Declaration, OPEN, CLOSE, and FETCH operations, Cursor Attributes, Advantages/Disadvantages.	8	12
4.	**Error Handling and Fault Tolerance:** Introduction to Exception Handling , Types of Exceptions: Named System Exceptions, Unnamed System Exceptions, User-defined Exceptions , Implementation of Exception Handling.	6	13
5.	**Automating Database Operations:**	6	14

	Triggers and their Features , Types of Triggers (Row/Statement, BEFORE/AFTER), Syntax for creation, Trigger Events (DML/DDL), Implementation of BEFORE and AFTER Trigger.		
6.	**Persistent Code Objects: Functions and Procedures:** Detailed design and implementation of Stored Procedures and User-Defined Functions, emphasizing parameter passing, modularity, and return values.	6	12
7.	**Optimizing Procedural Code:** Best practices for writing secure and performant code (minimizing context switching, SQL injection prevention, bulk operations).	3	13
8.	**Advanced RDBMS Features:** Advanced Indexing, Query Tuning, In-Memory features, and Replication/High-Availability solutions in RDBMS.	3	15
	Total	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Implement DDL (CREATE, ALTER) to set up a minimum of 5 tables with diverse datatypes. Implement DML (INSERT, UPDATE, DELETE) with a minimum of 10 rows per table.	6
2.	Execute SELECT commands using various SQL Operators (LIKE, BETWEEN, IN). Implement queries utilizing WHERE, HAVING, GROUP BY, and ORDER BY clauses.	6
3.	Write and execute basic standalone PL/SQL programs, defining variables, constants, and demonstrating assignment operations within the block structure.	6
4.	Write PL/SQL programs utilizing various forms of the Conditional Statement structure (IF-THEN-ELSE, ELSIF, Nested IF) to implement branching logic.	2
5.	Write PL/SQL programs to solve problems using both the Simple CASE statement and the Searched CASE expression for multi-way conditional logic.	2
6.	Write PL/SQL programs using the fundamental LOOP...END LOOP and the WHILE LOOP constructs, incorporating EXIT conditions.	4
7.	Write PL/SQL programs utilizing the FOR LOOP construct for sequential iteration, particularly for indexed operations and ranges.	4
8.	Demonstrate the function and handling of Implicit Cursors created automatically by DML operations, checking Cursor Attributes (%FOUND, %ROWCOUNT).	4
9	Write PL/SQL code to implement a basic Explicit Cursor process: DECLARE, OPEN, single FETCH, and CLOSE.	2
10	Implement Explicit Cursors within loops (e.g., Cursor FOR loop) for efficient multi-row processing and dynamic data retrieval.	4
11	Write PL/SQL programs to handle Predefined/Named System Exceptions (e.g., TOO_MANY_ROWS, NO_DATA_FOUND) using the WHEN clause.	4
12	Implement User-Defined Exceptions for custom error conditions, including declaring, raising, and handling them.	4

13	Write PL/SQL code for creating Database Triggers that fire on DML events (INSERT, UPDATE, DELETE), including both BEFORE and AFTER timing.	4
14	Write PL/SQL code for creating a reusable Stored Procedure and a User-Defined Function, demonstrating parameter passing and return values.	4
15	Write SQL commands to implement various types of Index on data	4
	TOTAL	**60**

Text Book (s):

Title	Author/s	Publication
Database System Concepts	Henry Korth	Tata McGraw Hill

Reference Book (s):

Title	Author/s	Publication
SQL, PL/SQL the Programming language of Oracle	Ivan Bayross	BPB Publication
"An introduction to Database Systems"	CJ Date.	Pearson
Oracle: The Complete Reference	George Koch, Kevin Loney	TMH/oracle press
Fundamental of Database System	R. Elmasri and S.B Navathe	Benjamin/Cummin g

Web Material Link(s):

- https://www.tutorialcup.com/rdbms
- https://www.geeksforgeeks.org/rdbms/
- https://onlinecourses.nptel.ac.in/

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICCA2031	Relational Database Management System
CO 1	Understand PL/SQL block structure, data types, variables, and procedural programming fundamentals in databases.

CO 2	Apply flow-control and iterative constructs to implement logical operations within PL/SQL programs.
CO 3	Use cursors and data-retrieval mechanisms effectively within procedural code for efficient data processing.
CO 4	Implement exception handling, triggers, stored procedures, and user-defined functions for automated and modular database operations.
CO 5	Apply optimization techniques and advanced RDBMS features such as indexing, query tuning, in-memory processing, and high-availability solutions.

Mapping of CO with PO

ICCA2031	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	2	3	3	1	2	1		
CO 2	2	3	3	2	3	1		2
CO 3	3	2	3	1	3	2	1	
CO 4	2	3	2	2	2	1		2
CO 5	2	2	2	3	3	3	1	2

Mapping of CO with PSO

ICCA2031	PSO1	PSO2	PSO3
CO 1	1	3	2
CO 2	3	2	2
CO 3	2	3	2
CO 4	2	2	3
CO 5	3	3	2

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to Procedural Database Programming	1,2
2	Flow Control Structures	2,3
3	Data Retrieval Techniques in Procedural Code	1,2,3,4
4	Error Handling and Fault Tolerance	1,2,3,4
5	Automating Database Operations	2,3,4,5
6	Persistent Code Objects: Functions and Procedures	3,4,5
7	Optimizing Procedural Code	4,5,6
8	Advanced RDBMS Features	3,4,5

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Application

Course Code: ICCA2041
Course Name: Gaming with Python
Prerequisite Course(s): -ICCS2010

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	02	-	04	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:

- To establish foundational proficiency in the Python programming language and its application in rapid software prototyping.
- To impart practical skills in leveraging multimedia libraries to design and develop interactive 2D graphical game environments.
- To analyze and implement key concepts of game physics, event handling, and collision detection within a structured development framework.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Python Fundamentals for Gaming** Python refresher (Data Structures, Functions, Classes). Setting up Pygame. Basic game loop structure. Event handling basics.	4	10
2.	**Graphics, Screen, and Assets** Creating the display window. Loading and displaying images and sound. Managing sprite surfaces. Color management (RGB).	5	10
3.	**Event Handling and User Input** Detailed analysis of the Pygame event system. Handling keyboard and mouse input. Custom events and timers.	5	12
4.	**Sprite Animation and Movement** Creating sprite classes. Implementing basic linear and diagonal movement. Frame-based animation techniques (sprite sheets).	6	12
5.	**Collision Detection and Game State** Rect object usage for bounding boxes. Implementing simple collision logic (AABB). Managing game state (Start, Playing, Paused, Game Over).	6	15
6.	**Basic Game Physics and Scoring** Implementing gravity and jumps. Velocity and acceleration concepts. Managing score, health, and lives. Displaying text (fonts).	6	15

7.	**Advanced Graphics and Sound Design** Introduction to Particle Systems, basic Camera mechanics (scrolling backgrounds), and advanced sound mixing.	5	10
8.	**Project Implementation & Debugging** Final Project Focus: Game packaging (executable creation), Debugging strategies, Performance profiling in Pygame, Licensing basics.	8	16
	TOTAL	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Install Pygame. Create a basic window, implement the game loop, and handle the quit event to ensure a clean exit.	2
2.	Load two different image files (sprites) and display them at fixed coordinates on the screen. Implement background color changes.	2
3.	Implement keyboard event handling (KEYDOWN/KEYUP) to move a single rectangular object across the screen in four directions (up, down, left, right).	2
4.	Implement mouse motion handling to dynamically draw circles or lines on the screen where the cursor moves.	2
5.	Define a Python class for a game character (Sprite). Initialize its image, rect, and define a method for its basic movement logic.	2
6.	Modify the Sprite class to use velocity variables (dx, dy) and update its position continuously within the game loop, simulating diagonal movement.	2
7.	Implement a simple frame-based animation by cycling through a list of images/surfaces over time to animate a running or walking character.	2
8.	Apply basic simulated physics by implementing gravity (downward acceleration) and a controlled jump function for the main player sprite.	2
9.	Implement collision detection between two rectangular sprites (.colliderect()) and change the color or stop movement upon impact.	2
10.	Use Pygame Groups or a list to manage multiple static obstacle sprites and check for collisions between the player and any obstacle in the list.	2
11.	Implement a game_state variable and use IF/ELIF logic to display separate screens for "START MENU" and "GAME OVER."	2
12.	Implement a scoring system that increments based on an event (e.g., collision). Display the current score on the screen using a custom font.	2
13.	Load background music and a sound effect. Play the music on loop and trigger the sound effect upon a user action (e.g., shooting or jumping).	2
14.	Design and implement a "bullet" sprite that travels linearly when the player presses a specific key. Implement collision between the bullet and a target enemy.	2
15.	Submit a fully functional 2D mini-game (e.g., Pong, Flappy Bird, or simple platformer) that incorporates all learned concepts (movement, collision, scoring, and state).	2
	TOTAL	**30**

Text Book (s):

Title	Author/s	Publication
Python Crash Course, 3rd Edition *(Focusing on Pygame chapters)*	Eric Matthes	No Starch Press

Reference Book (s):

Title	Author/s	Publication
Invent Your Own Computer Games with Python, 4th Edition	Al Sweigart	No Starch Press
Beginning Game Development with Python and Pygame	Will McGugan	Apress
Game Programming with Python, Lua, and C++	Tom Miller	Jones & Bartlett Learning

Web Material Link(s):

- https://www.geeksforgeeks.org/blogs/python-game-development/
- https://onlinecourses.nptel.ac.in/noc21_cs32/preview
- https://nptel.ac.in/courses/106106182
- https://www.tutorialspoint.com/pygame/index.htm

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICCA2041	Gaming with Python
CO 1	Apply core Python programming principles to architect the fundamental structure of a 2D game environment and manage continuous program flow.
CO 2	Construct interactive game elements by implementing effective event handling mechanisms and managing the state transitions of a multi-stage game.
CO 3	Develop dynamic visual components and character interactions using sprite classes, frame animation, and basic simulated physics.
CO 4	Analyze and implement algorithms for collision detection, scoring, and level progression to create a functionally complete game experience.

CO 5	Evaluate and integrate advanced visual, audio, and performance optimization techniques, culminating in the creation of a polished, distributable game project.

Mapping of CO with PO

ICCA2041	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	2	1	2	1		
CO 2	3	2	3	2	3	2		2
CO 3	2	3	2	2	3	2		
CO 4	3	3	3	2	2	2	1	
CO 5	2	3	2	3	3	3	2	1

Mapping of CO with PSO

ICCA2041	PSO1	PSO2	PSO3
CO 1	3	3	1
CO 2	2	3	1
CO 3	3	3	1
CO 4	2	2	1
CO 5	3	3	3

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Python Fundamentals for Gaming	1,2
2	Graphics, Screen, and Assets	1,2,3
3	Event Handling and User Input	2,3
4	Sprite Animation and Movement	3,4,5,6
5	Collision Detection and Game State	3,4,6
6	Basic Game Physics and Scoring	3,4,5
7	Advanced Graphics and Sound Design	2,4,6
8	Project Implementation & Debugging	3,5,6

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Information Technology

Course Code: ICIT2040
Course Name: Operating Systems
Prerequisite Course(s): -

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	02	-	04	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:

To help learners to

- Learn the principles of operating system design.
- Understand architecture of computer based operating systems and its components.
- Understand various software and hardware processes and its lifecycle.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Introduction to Operating System:** Overview, What is OS? Brief history of OS, Computer Hardware- Processor, Memory, Non-Volatile Storage, I/O Devices, Buses, Types of OS, Functions of OS, Modern OS trends- Cloud OS, Mobile OS, Real-time embedded OS	5	10
2.	**Processes & Threads:** Overview of Process, Process Model, Process Creation & Termination, Process States. Introduction of Thread, Thread Model, Single Thread vs Multi thread system, Thread scheduling on multicore systems	6	13
3.	**Synchronization & Inter-process Communication:** Overview, Rare Conditions, Critical Regions, Mutual Exclusion, Sleep & Wakeup, Semaphores, Mutexes, Monitors, Message Passing, Barriers, Priority Inversion, Avoiding Locks, Synchronization in distributed systems / multicore processors.	6	13
4.	**Scheduling:** Introduction, Scheduling in Interactive Systems & Real time Systems, Thread Scheduling, Multi-core scheduling, Cloud and container scheduling (Kubernetes pod scheduling – conceptual)	5	11
5.	**Memory Management:**	8	18

	Role and importance of memory management in OS, Overview of memory abstraction, Logical vs Physical Address Space, Memory Hierarchy, Virtual Memory- Paging: concept, page tables, translation lookaside buffer (TLB), Page table structures: single-level, multi-level, inverted page tables, Page faults and handling, Modern page replacement algorithms: Least Recently Used, Working Set, Not Recently Used, Introduction to Non-Uniform Memory Access (NUMA)		
6.	**File Systems:** Files- Naming, Structure, Types, File Access, Attributes, File Operations. Directories- Hierarchical directory systems, path names, operations. Disk Space Management, Modern world Examples of File System (ext4, NTFS, APFS, ZFS).	6	13
7.	**Input/Output Operations:** I/O Devices, Device Controller, Memory mapped I/O, DMA. Goal of I/O Software, Programmed I/O, Interrupt driven I/O, I/O using DMA, Solid State Drives (SSDs), NVMe, Virtual I/O in cloud.	5	10
8.	**Deadlocks:** Resources types – preemptable & non-preemptable, Resource Acquisition. The Dining Philosopher problem, Introduction to Deadlock, Condition, Deadlock Modeling, The Ostrich Algorithm, Deadlock Detection & Recovery, Deadlock Avoidance, Deadlock Prevention, Two phase Locking, Communication deadlock, Live lock, Starvation.	4	12
	Total	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Install a Linux OS (Ubuntu) in VirtualBox and explore its GUI & shell. Identify and list system hardware information (processor, memory, I/O) using Linux commands (lscpu, lsblk, free, etc.).	2
2.	Write a program to create a new process using fork() and print parent/child PIDs. Demonstrate process states using ps, top, and kill commands. Implement a basic multithreaded program. Compare single-threaded and multithreaded execution time for a simple task.	4
3.	Implement critical section problem and show race conditions. Use semaphores to synchronize two threads updating a shared counter.	2
4.	Implement FCFS scheduling algorithm and compute waiting & turnaround time. Implement SJF scheduling and compare with FCFS using same input. Simulate Round Robin scheduling with varying time quantum. Implement Priority Scheduling (preemptive and non-preemptive). Compare scheduling algorithms using Gantt charts and performance metrics.	6
5.	Implement a simple paging mechanism with page table lookup. Explore NUMA topology on Linux using numactl or system info commands.	4
6.	Create and manipulate files using basic Linux commands (touch, cat, chmod).	4

	Demonstrate hierarchical directory structure using mkdir, tree, ls. Write a C/Python program to read/write from files using system calls. Explore different file attributes using stat command. Simulate file allocation techniques (contiguous, linked, indexed).	
7.	List and categorize I/O devices using Linux commands. Explore NVMe device information using nvme CLI tool. Explore virtual I/O in VirtualBox (shared folders, virtual disks).	4
8.	Simulate the Dining Philosophers Problem with and without deadlock avoidance. Implement Banker's Algorithm for deadlock avoidance.	4
	TOTAL	**30**

Text Book (s):

Title	Author/s	Publication
Modern Operating Systems	Andrew S. Tanenbaum	Pearson Education

Reference Book (s):

Title	Author/s	Publication
Operating System Concepts	Silberschatz A., Galvin P. and Gagne G	Wiley
Operating Systems: Internals and Design Principles	William Stallings	Pearson

Web Material Link(s):

- https://nptel.ac.in/courses/106108101/
- https://nptel.ac.in/courses/106106144/
- https://computer.howstuffworks.com/operating-system.htm
- https://www.tutorialspoint.com/computer_fundamentals/computer_operating_system.htm
- https://www.geeksforgeeks.org/operating-systems-need-and-functions/

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which will be converted to 30 marks.
- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will beconverted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks during End Semester Examination.
- Viva/Oral performance consists of 30 marks during End Semester Examination.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICIT2040	Operating Systems
CO 1	Explain the fundamental concepts, types, functions, and modern trends of operating systems
CO 2	Describe process and thread management concepts including creation, states, scheduling, and multithreading on multicore systems.
CO 3	Apply synchronization mechanism to solve critical section and inter-process communication problems.
CO 4	Analyse memory management techniques and page replacement algorithms for efficient utilization of memory.
CO 5	Demonstrate understanding of file systems, I/O operations, and deadlock handling techniques used in modern operating systems.

Mapping of CO with PO

ICIT2040	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	2	2	2	1		
CO2	3	3	3	3	2			2
CO3	3	3	3	3	3			
CO4	2	3	3	3	3	1	1	2
CO5	2	2	3	3	3	1	1	2

Mapping of CO with PSO

ICIT2040	PSO1	PSO2	PSO3
CO1	2	2	
CO2	3	2	1
CO3	3	2	1
CO4	3	3	2
CO5	3	3	2

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1	Introduction to Operating System	1,2
2	Processes & Threads	1,2,3
3	Synchronization & Inter-process Communication	2,3,4
4	Scheduling	2,3,5,6
5	Memory Management	2,3,5,6
6	File Systems	3,4
7	Input/Output Operations	1,2,3
8	Deadlocks	4,5,6

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Application

Course Code: ICCA2050
Course Name: Object oriented Programming with JAVA
Prerequisite Course(s): --

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	04	-	05	40	60	40	60	-	-	200

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help learners to

- To establish robust programming and debugging skills using a modern, platform-independent language environment.
- To impart mastery over the principles of object-oriented design (OOP), including inheritance, encapsulation, and abstraction.
- To enable the construction of reliable, concurrent, and modular applications by utilizing exception handling and multi-process execution.

Course Content:

Module No.	Content	Hours	Weightage in %
1.	**Java Ecosystem and Basic Constructs:** Programming language Paradigms, History of Java, Features, JVM and Bytecode. Java Environment, Compilation, Execution. Constants, Variables, Comments, Data Types, Operators, and Control Structures (Implicit). Introduction to Debugging: Process, Common Errors, Techniques.	5	12
2.	**Object-Oriented Programming (OOP) Fundamentals** Class, Object, Object reference, Garbage Collection and finalize() method. Constructor & types of Constructors, initialization block. Access Control Modifiers, Use of this keyword, Polymorphism-Method overloading	5	12
3.	**Data Structures: Arrays and Strings** Defining, Initializing & Accessing Arrays (Multi-Dimensional). Operations on String, String Tokenizing, StringBuffer.	5	9

4.	**Inheritance and Run-time Polymorphism** Benefits of Inheritance, Types of Inheritance, Inheriting Members, Role of Constructors in inheritance, Polymorphism- Method Overriding, Dynamic Method Dispatch, use of super keyword.	6	13
5.	**Abstraction and Modularity** Abstract Class and Interfaces. Nested/Inner/Anonymous Classes. Java Packages: Organizing Classes/Interfaces, CLASSPATH, JAR Files, Creating package, nested packages, Importing package.	6	16
6.	**Exception Handling** Idea behind Exception, Types (Errors, Checked/Un-Checked), Control Flow (try, catch, finally, throw, throws). In-built and User Defined Exceptions.	5	12
7.	**Concurrency: Threads** Understanding Threads, Need for Multi-Threaded Programming, Thread Life Cycle, Thread Priorities, Synchronizing Threads, Inter-Thread Communication.	7	14
8.	**Generics** What are Generics? Example, Generic Class, Generic Constructors, Generic Methods, Generic super class & sub class, Generic method overriding, Some Generic Restrictions	6	12
	Total	**45**	**100**

List of Practical:

Sr. No	Name of Practical	Hours
1.	Install Java SDK/IDE. Write programs using variables, operators, and control flow. Demonstrate basic debugging techniques for correcting common syntax and runtime errors.	4
2.	Implement Java programs with multiple classes and objects, demonstrating object creation and method invocation. Implement programs for variables, data types, and operators.	2
3.	Demonstrate the use of parameterized Constructors, Static members, and the application of different access specifiers (public, private, protected, default).	2
4.	Implement programs to showcase method overloading (Compile-time) and method overriding (Run-time) within a class structure.	4
5.	Implement programs demonstrating single-level, multi-level, and hierarchical inheritance. Use the super keyword effectively.	4
6.	Implement Java programs to demonstrate the use of Interfaces by defining and implementing multiple methods across different classes.	4

7.	Implement programs demonstrating the usage of Abstract Classes (partial implementation) and Anonymous Classes for concise object creation.	4
8.	Implement programs using multi-dimensional Arrays (e.g., matrix operations) and solve a problem requiring array searching/sorting.	4
9.	Implement programs to perform various string operations and contrast the efficiency of the String class with StringBuffer for mutable sequences.	4
10.	Implement Java programs demonstrating the usage and scope of Inner Classes (member, static nested, local, and anonymous).	4
11.	Create and demonstrate a Java Package structure containing multiple classes/interfaces. Implement code that uses import and static import.	4
12.	Write programs to demonstrate exception handling using all keywords. Handle at least two common Predefined Exceptions.	4
13.	Implement a User-Defined Exception class. Additionally, implement basic File Handling operations (read/write), which requires handling a Checked Exception.	4
14.	Implement programs to create threads using both the Thread class and the Runnable interface. Demonstrate the states of the Thread Life Cycle.	4
15.	Implement a program to demonstrate thread synchronization & Inter-Thread Communication.	4
16.	Implement Generic Class along with generic constructors & methods.	4
	TOTAL	**60**

Text Book (s):

Title	Author/s	Publication
The Complete Reference, Java	Herbert Schildt	McGraw-Hill Education

Reference Book (s):

Title	Author/s	Publication
Thinking in Java	Bruce Eckel	Pearson
Learning Java	Patrick Niemeyer and Jonathan Knudsen	O'reilly Media

Web Material Link(s):

- https://onlinecourses.nptel.ac.in/noc22_cs47/preview
- https://nptel.ac.in/courses/106105191
- https://www.tutorialspoint.com/java/java_oops_concepts.htm
- https://www.geeksforgeeks.org/java/object-oriented-programming-oops-concept-in-java/

Course Evaluation:

Theory:

- Continuous Evaluation consists of two tests each of 30 marks and 1 hour of duration, which

will be converted to 30 marks.

- Faculty evaluation consists of 10 marks as per the guidelines provided by the course coordinator.
- End Semester Examination consists of 60 marks.

Practical:

- Continuous Evaluation consists of the performance of practical, which will be evaluated out of 10 per each practical. At the end of the semester, the average of the entire practical will be converted to 20 marks.
- Internal viva consists of 20 marks.
- Practical performance/quiz/test consists of 30 marks.
- External viva consists of 30 marks.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICCA2050	Object oriented Programming with JAVA
CO 1	Understand the Java ecosystem, basic constructs, compilation process, and fundamental debugging techniques.
CO 2	Apply object-oriented principles including classes, objects, constructors, access control, and method overloading.
CO 3	Work with arrays, strings, inheritance, and run-time polymorphism to develop modular Java applications.
CO 4	Implement abstraction using abstract classes, interfaces, packages, and handle exceptions effectively.
CO 5	Develop concurrent and type-safe programs using threads, synchronization mechanisms, and generics in Java.

Mapping of CO with PO

ICCA2050	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	3	2	2	1	2	1		
CO 2	3	3	3	2	3	1	1	
CO 3	3	2	3	3	3	2	1	
CO 4	2	3	3	2	2	2		2
CO 5	3	3	2	3	3	3	1	2

Mapping of CO with PSO

ICCA2050	PSO1	PSO2	PSO3
CO 1	3	3	1
CO 2	3	2	2
CO 3	2	3	2
CO 4	2	2	3
CO 5	3	3	2

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	**Content**	**RBT Level**
1	Java Ecosystem and Basic Constructs	1,2
2	Object-Oriented Programming (OOP) Fundamentals	2,3
3	Data Structures: Arrays and Strings	2.3,4
4	Inheritance and Run-time Polymorphism	1,2,4
5	Abstraction and Modularity	3,4,6
6	Exception Handling for Robustness	3,5,6
7	Concurrency: Threads	3,4,5
8	Generics	3,4,6

P P Savani University
School of Engineering
Institute of Computer Science and Application

Department of Computer Application

Course Code: ICCA2060
Course Name: Online Course-I
Prerequisite Course(s): --

Teaching & Examination Scheme:

Teaching Scheme (Hours/Week)				Examination Scheme (Marks)						
Theory	Practical	Tutorial	Credit	Theory		Practical		Tutorial		Total
				CE	ESE	CE	ESE	CE	ESE	
03	-	-	03	100	00	-	-	-	-	100

CE: Continuous Evaluation, ESE: End Semester Exam

Objective(s) of the Course:
To help the learners to
- Learn new subjects as per recent trends in the industry from national experts.

Course Content:
Performance analysis will be based on any one of the following subjects:
1. Deep Learning
2. Computer Graphics
3. R Programming
4. Artificial Intelligence
5. Machine Learning
6. Virtual Reality
7. Advanced Distributed Systems
8. Software Project management
9. Ethical Hacking
10. UI & UX
11. Data Analytics
12. Introduction to Embedded System Design
13. Algorithmic Game Theory
14. Natural Language Processing
15. Big Data
16. Generative AI
17. Blockchain Technology
18. Digital Image Processing
19. Internet of Things
20. Augmented Reality
21. Real time systems
22. Data Visualization
23. Reinforcement Learning
24. DevOps and Agile Foundation

Or any other NPTEL course; available from time to time.

Course Evaluation:
Theory:
- Continuous Evaluation as per the guidelines of NPTEL assignments and tests.

- The NPTEL score will be directly fetched and converted out of 100.

Course Outcome(s):

After the completion of the course, the following course outcomes will be able to:

ICCA2060	Online Course-I
CO 1	Inculcate mode of self-learning.
CO 2	Exposure to relevant and newest tools and technologies.
CO 3	Value addition when the student is applying for jobs.
CO 4	Use NPTEL program for GATE and high studies preparation.
CO 5	Facilitate students to attain certificate and to make them employable in the industry or pursue higher education program.

Mapping of CO with PO

ICCA2060	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO 1	1		1		2		1	1
CO 2	1		2	2	2		1	2
CO 3		2	2	2	2	1		
CO 4	2	1		1	1			2
CO 5						1		3

Mapping of CO with PSO

ICCA2060	PSO1	PSO2	PSO3
CO 1	2	2	3
CO 2	3	2	2
CO 3	2	3	3
CO 4	3	3	2
CO 5	3	3	3

Level of Bloom's Revised Bloom's Taxonomy in Assessment

1: Remember	2: Understand	3: Apply
4: Analyze	5: Evaluate	6: Create

Module No	Content	RBT Level
1.	Online Course	1,2,3,4,5,6