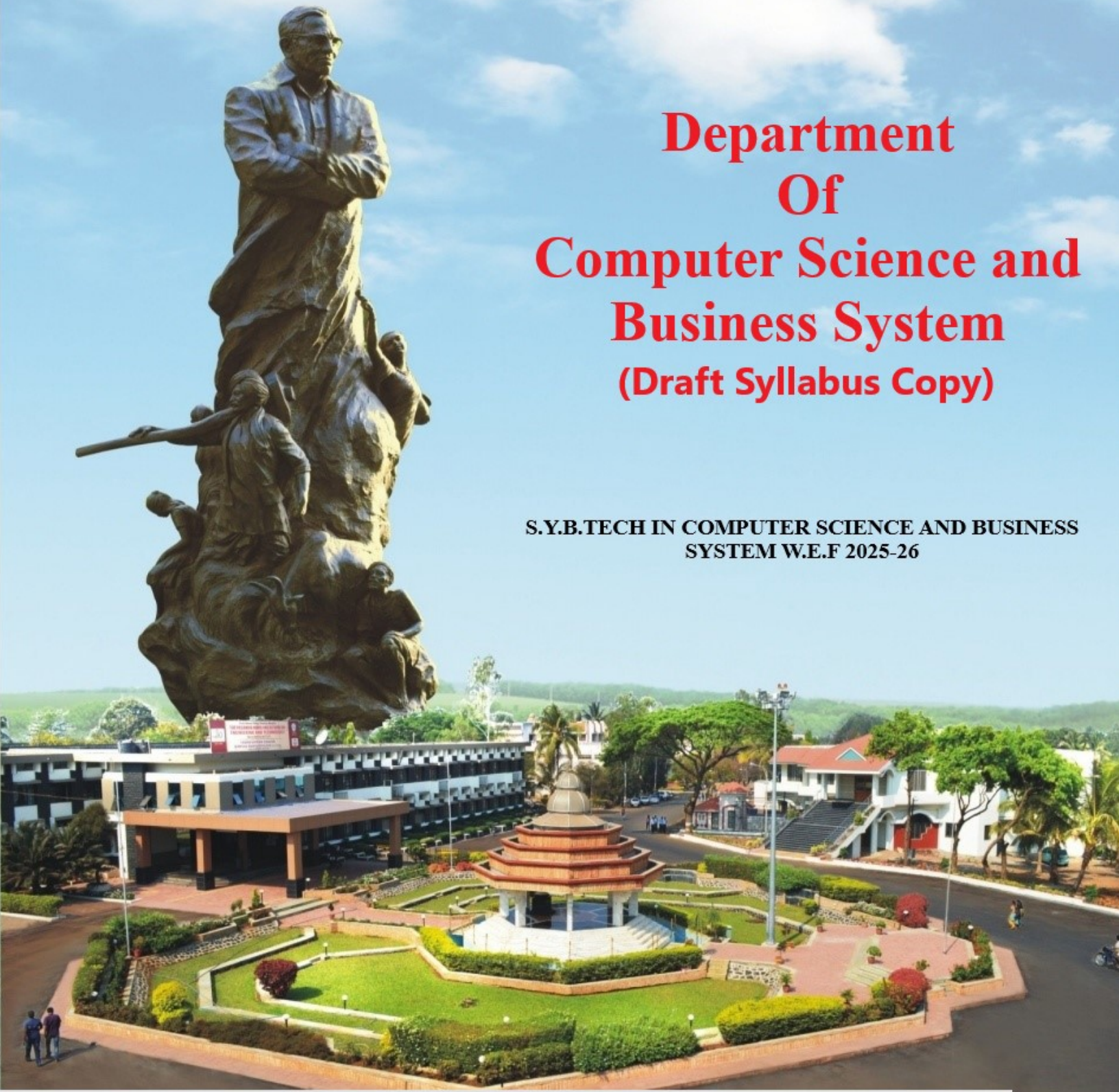
Shree Warana Vibhag Shikshan Mandal's

**Tatyasaheb Kore Institute of Engineering And Technology, Warananagar**

# Department Of Computer Science and Business System

## (Draft Syllabus Copy)

**S.Y.B.TECH IN COMPUTER SCIENCE AND BUSINESS SYSTEM W.E.F 2025-26**



**S. Y. B.Tech. (Sem-III & IV) Computer Science & Business System**

**Syllabus Structure and Curriculum as per NEP 2020**

# Department of Computer Science and Business System

## Vision

To be a leading Computer Science & Business Systems department recognized for excellence in technical and management education, fostering innovation, academic flexibility, industry readiness, and a commitment to professional ethics, shaping future leaders who drive sustainable development and societal progress.

## Mission

- To continuously implement a dynamic and adaptable curriculum.
- To enhance industry-institute collaboration to effectively promote internships, improve employability, and foster entrepreneurial skills among students.
- To cultivate an environment that inspires faculty and students to actively participate in impactful academic and research pursuits.
- To expand access to quality education for rural and underprivileged sections of society.
- To instill a commitment to lifelong learning enriched with human values, social responsibility

## Program Educational Objectives (PEOs)

**Graduates will be,**

- To design and develop computing systems by integrating modern technologies and addressing business intelligence challenges.
- To acquire capabilities for pursuing higher education and entrepreneurship with a strong aptitude for innovation
- To function effectively as professionals with excellent interpersonal skills, upholding ethical and social responsibilities
- To lead in their respective domains and contribute positively to societal needs, working efficiently in multidisciplinary and multicultural environments.

## Quality Policy

To promote excellence in academic and training activities by inspiring students for becoming competent professionals to cater industrial and social needs.

# Program Outcomes (POs)

**1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## Abbreviations

| Sr.No | Acronym | Defination |
|---|---|---|
| 1 | ISE | In-Semester Examination |
| 2 | ISE-I | In-Semester Examination-I |
| 3 | ISE-II | In-Semester Examination-II |
| 4 | ESE | End Semester Examination |
| 5 | ISA | In-Semester Assessment (Term Work) |
| 6 | L | Lecture |
| 7 | T | Tutorial |
| 8 | P | Practical |
| 9 | CH | Contact Hours |
| 10 | C | Credit |

## Course Categories

| Sr.No | Acronym | Defination |
|---|---|---|
| 1 | BSC | Basic Science Course |
| 2 | ESC | Engineering Science Course |
| 3 | PCC | Program Core Courses |
| 4 | HSSM | Humanities Social Science and Management |
| 5 | VSEC | Vocational and Skill Enhancement Course |
| 6 | CC | Co-curricular course |

## Course/ Subject Code for Theory and Practical

| 2 | 5 | UG | PCC | CB | 3 | 0 | 1 | T/P/A |
|---|---|---|---|---|---|---|---|---|
| Course Introduced Year | | Under Graduate | Program Code | Course Category | Semester | Course Number | | T-Term work P-POE A - Audit Course |

# Second Year B. Tech.
# (Semester – III)
# Detailed Syllabus

# In
# Computer Science & Business System

**Curriculum Structure, Credits and Evaluation Scheme as per NEP 2020**
**(To be implemented from AY 2025-26)**

Shree Warana Vibhag Shikshan Mandal's
Tatyasaheb Kore Institute of Engineering & Technology
Warananagar, Tal- Panhala, Dist- Kolhapur -416 113. Maharashtra
An Autonomous Institute, Affiliated to Shivaji University, Kolhapur

# Second Year B.Tech in Computer Science & Business System Semester-III

## To be Implemented from 2025-26

## Scheme Curriculum Structure, Credit Scheme and Evaluation

| Sr. No | Category | Sub- Category | Course Code | Course Title | Teaching Scheme | | | | | Examination & Evaluation Scheme | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | L | T | P | C | CH | Component | Marks | Min Marks for Passing | |
| 1 | Program Core Course | PCC | 25UG-PCC- CB301 | Formal Language and Automata Theory | 3* | -- | -- | 2 | 3 | ISE | 40 | 16 | 40 |
| | | | | | | | | | | ESE | 60 | 24 | |
| 2 | | PCC | 25UG-PCC- CB302 | Computer Organization and Architecture | 2 | -- | -- | 2 | 2 | ISE | 40 | 16 | 40 |
| | | | | | | | | | | ESE | 60 | 24 | |
| 3 | | PCC | 25UG-PCC- CB303 | Computational Statistics | 3* | -- | -- | 2 | 3 | ISE | 40 | 16 | 40 |
| | | | | | | | | | | ESE | 60 | 24 | |
| 4 | | PCC | 25UG-PCC- CB304 | Software Engineering | 3* | -- | -- | 2 | 3 | ISE | 40 | 16 | 40 |
| | | | | | | | | | | ESE | 60 | 24 | |
| 5 | | PCC | 25UG-PCC- CB305 | Object Oriented Programming | 3* | - | -- | 2 | 3 | ISE | 40 | 16 | 40 |
| | | | | | | | | | | ESE | 60 | 24 | |
| 6 | Multi-Disciplinary Course | MDM-1 | 25UG- MDM1-CB306 | Object-Based Programming | 2 | -- | -- | 2 | 2 | ISA | 25 | 10 | 10 |
| 7 | Humanities Social Science and Management | Entrepreneurship/Economics/ Mgmt. Course | 25UG- EEC1- CB307-1 | Financial Management | 2* | -- | -- | 1 | 2 | ISA | 25 | 10 | 10 |
| 8 | | Value Education Course | 23UG- VEC1-CB308-1 | Ethics and Values in Business and Technology | 2 | -- | -- | 2 | 2 | ISA | 25 | 10 | 10 |
| 9 | Experiential Learning Courses | Community Engg Project (CEP/FP) | 23UG-CEP- CB309 | Community Engagement Activity (CEA)/Field Project | -- | -- | -- | 2 | -- | ISA | 25 | 10 | 10 |
| 11 | Program Core Course | PCC | 23UG-PCC- CB302P | Computer Organization and Architecture Lab | | | 2 | 1 | 2 | ISA | 25 | 10 | 10 |
| 12 | | PCC | 23UG-PCC- CB305P | Object Oriented Programming Lab | -- | -- | 2 | 1 | 2 | ISA | 25 | 10 | 10 |
| | | | | | | | | | | POE | 50 | 20 | 20 |
| 13 | | PCC | 23UG-PCC- CB303P | Computational Statistics Lab | -- | | 2 | 1 | 2 | ISA | 25 | 10 | 10 |
| | | | | | | | | | | POE | 50 | 20 | 20 |
| | | PCC | 23UG-PCC- CB304P | Software Engineering Lab | | | 2 | 1 | 2 | ISA | 25 | 10 | 10 |
| | | | | | 20 | -- | 08 | 21 | 028 | -- | 800 | 320 | 320 |

# Multidisciplinary Minor (MDM) Courses Bucket

## Branch: Computer Science & Buisness System

## Fundamentals of Computing and Software Systems

| Sr. No | Semester | Category | Course Code | Course Title | Teaching Scheme | | | | | Examination & Evaluation Scheme | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | L | T | P | C | CH | Component | Marks | Min Marks for Passing | |
| 1 | III | MDM-1 | 25UG- MDM1- CB306 | Object Oriented Programming | 2 | -- | -- | 2 | 2 | ISA | 50 | 20 | 20 |
| 2 | IV | MDM-II | 25UG- MDM2- CB405 | Operations Research | 2 | -- | -- | 2 | 2 | ISA | 50 | 20 | 20 |
| 3 | V | MDM-III | 25UG-PCC- CB505 | Data Structure & Algorithm | 3 | -- | -- | 3 | 3 | ISE | 40 | 16 | 40 |
| | | | | | | | | | | ESE | 60 | 24 | |
| 4 | VI | MDM-IV | 25UG-PCC- CB605 | Operating System | 2 | -- | -- | 2 | 2 | ISA | 50 | 20 | 20 |
| 5 | VII | MDM-V | 25UG-PCC- CB705 | Database Management Systems | 3 | -- | -- | 3 | 3 | ISA | 50 | 20 | 20 |
| 6 | VIII | MDM-VI | 25UG-PCC- CB804 | System Analysis and Software Testing | 2 | -- | -- | 2 | 2 | ISA | 50 | 20 | 20 |
| | | | | | 14 | | | 14 | 14 | -- | 350 | 140 | 140 |

# 25UG-PCC- CB301- Formal Language and Automata Theory

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | 3* Hrs/Week | **ISE** | **:** | 40 Marks |
| **Credits** | **:** | 2 | | | |
| **Tutorials** | **:** | -- | **ESE** | **:** | 60 Marks |

| **Course Objective:** The objective of this course is to | |
|---|---|
| **1** | To introduce the fundamental concepts of formal languages, automata, and grammar classifications, including the Chomsky hierarchy. |
| **2** | To develop the ability to design and analyze finite automata, regular expressions, context-free grammars, and pushdown automata for recognizing different language classes. |
| **3** | To explore the capabilities and limitations of Turing machines and understand the concepts of decidability, undecidability, and computational universality. |
| **4** | To provide a basic understanding of computational complexity, including the classes NP, NP, NP-completeness, and key theoretical results like Cook's Theorem. |

**Course Outcomes:**

| **COs** | **At the end of successful completion of the course, the students will be able to** | **Bloom's Taxonomy** |
|---|---|---|
| **CO1** | Describe the fundamental concepts of formal languages, grammars, and Chomsky hierarchy. | Understand |
| **CO2** | Construct deterministic and non-deterministic finite automata, regular expressions, and context-free grammars for language recognition. | Apply |
| **CO3** | Demonstrate the equivalence between different models of computation such as DFA, NFA, regular expressions, and regular grammars.. | Analyze |
| **CO4** | Use pumping lemma and related techniques to determine language properties and limitations of automata.. | Apply |
| **CO5** | Differentiate the behavior of Turing machines and **categorize** problems based on decidability and complexity classes like P and NP. | Analyze |

**Course Description :**

This course provides a theoretical foundation of computer science through the study of formal languages, automata, computability, and complexity. It covers key models of computation including finite automata, pushdown automata, and Turing machines. Students will explore the classification of languages through the Chomsky hierarchy and study properties of regular, context-free, and context-sensitive languages. The course also introduces fundamental concepts of undecidability and computational complexity, including P, NP, and NP-complete problems. It lays the groundwork for understanding what problems can be solved using computation and the inherent limitations of computational models.

| **Prerequisites :** | • Discrete Mathematics<br>• Data Structures and Algorithms<br>• Basics of C Programming. |
|---|---|

| Section – I | | |
|---|---|---|
| **Unit-1** | **Introduction:**<br>Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.<br>Regular languages and finite automata**:**<br>Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, Kleene's theorem, pumping lemma for regular languages, Myhill- Nerode theorem and its uses, minimization of finite automata. | **06Hr** |
| **Unit-2** | **Context-free languages and pushdown automata:**<br>Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs. | **06Hr** |
| **Unit-3** | **Context-sensitive languages:**<br>Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. | **06Hr** |
| **Section-II** | | |
| **Unit-4** | **Turing machines:**<br>The basic model for Turing machines (TM), Turing recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators | **6Hr** |
| **Unit-5** | **Undecidability:**<br>Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages. | **6Hr** |
| **Unit-6** | **Basic Introduction to Complexity:**<br>Introductory ideas on Time complexity of deterministic and nondeterministic Turing machines, P and NP, NP- completeness, Cook's Theorem, other NP - Complete problems | **6Hr** |

**References**

**Text Books :**

1. Introduction to Automata Theory, Languages, and Computation John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman.

**Reference Books :**

1. Elements of the Theory of Computation, Harry R. Lewis and Christos Papadimitriou.
2. Automata and Computability, Dexter C. Kozen.
3. Introduction to the Theory of Computation, Michael Sipser.
4. Introduction to Languages and the Theory of Computation, John Martin.

# 25UG-PCC- CB302- Computer Organization and Architecture

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | 2 Hrs/Week | **ISE** | **:** | 40 Marks |
| **Credits** | **:** | 2 | | | |
| **Tutorials** | **:** | -- | **ESE** | **:** | 60 Marks |

| **Course Objective:** The objective of this course is to | |
|---|---|
| **1** | This course aims to introduce the fundamental concepts of computer organization, including Boolean logic, digital circuits, and the basic functional units of a computer system. It covers instruction set architecture, data representation, and arithmetic operations such as addition, multiplication, and floating-point calculations. Students will learn about CPU control unit design, memory organization, and input-output mechanisms like interrupts and DMA. The course also introduces pipelining, cache memory, and parallel processing concepts to understand system performance and efficiency. |

**Course Outcomes:**

| **COs** | **At the end of successful completion of the course, the students will be able to** | **Bloom's Taxonomy** |
|---|---|---|
| **CO1** | Explain the structure and function of basic computer components including CPU, memory, and I/O systems. | Understand |
| **CO2** | Describe the instruction set architecture, addressing modes, and instruction execution cycle of a typical CPU. | Understand |
| **CO3** | Perform number system conversions and computations using signed numbers, fixed-point, and floating-point representations. | Apply |
| **CO4** | Illustrate the working of arithmetic units such as adders and multipliers, and describe techniques for division and floating-point operations. | Apply |
| **CO5** | Discuss the functioning of control units, memory hierarchy, cache mapping, pipelining, and basics of parallel processing. | Apply |

**Course Description :**

This course covers the fundamentals of computer organization, including digital logic, CPU architecture, memory systems, and input-output mechanisms. It introduces instruction sets, data representation, arithmetic operations, and control unit design. Students will also learn about pipelining, cache memory, and parallel processing to understand modern computer performance.

| | |
|---|---|
| **Prerequisites :** | • Digital Logic and Circuits.<br>• Computer Fundamentals and Programming Concepts |

| Section – I | | |
|---|---|---|
| **Unit-1** | **Basics of Digital Circuits and Computer Architecture**<br>Basics in Boolean logic and Combinational/Sequential Circuits. Functional blocks of a computer: CPU, memory, input-output subsystems, control unit**.** | **4 Hr** |
| **Unit-2** | **Instruction set architecture of a CPU**: Registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Outlining instruction sets of some common CPUs.<br>**Data representation**: Signed number representation, fixed and floating point representations, character representation. | **08Hr** |
| **Unit-3** | **Computer arithmetic:**<br>Integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and-add, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic, IEEE 754 format. | **06Hr** |
| **Section-II** | | |
| **Unit-4** | **Introduction to x86 architecture.**<br>CPU control unit design: Hardwired and micro-programmed design approaches, design of a simple hypothetical CPU.<br>**Memory system design**: Semiconductor memory technologies, memory organization**.** | **6Hr** |
| **Unit-5** | **Peripheral devices and their characteristics:**<br>Input-output subsystems, I/O device interface, I/O transfers – program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes – role of interrupts in process state transitions, I/O device interfaces – SCII, USB | **6Hr** |
| **Unit-6** | **Pipelining:** Basic concepts of pipelining, throughput and speedup, pipeline hazards.<br>Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.<br>**Memory organization:** Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies. | **6Hr** |

**References**

**Text Books :**

1. *Computer System Architecture* M. M. Mano:, 3rd ed., Prentice Hall of India, New Delhi, 1993.
2. *Computer Organization and Design: The Hardware/Software Interface*, David A. Patterson and John L. Hennessy.
3. *Computer Organization and Embedded Systems*, Carl Hamacher.

**Reference Books :**

1. *Computer Architecture and Organization*, John P. Hayes.
2. *Computer Organization and Architecture: Designing for Performance*, William Stallings.
3. *Computer System Design and Architecture*, Vincent P. Heuring and Harry F. Jordan.

# 25UG-PCC- CB303- Computational Statistics

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | 3 Hrs/Week | **ISE** | **:** | 40 Marks |
| **Credits** | **:** | 2 | | | |
| **Tutorials** | **:** | -- | **ESE** | **:** | 60 Marks |

| **Course Objective:** The objective of this course is to | |
|---|---|
| **1** | Introduce the theoretical foundations and applications of multivariate techniques. |
| **2** | Develop competence in estimation and diagnostics for multivariate models. |
| **3** | Enable effective interpretation of complex datasets using statistical tools. |
| **4** | Guide the selection and validation of appropriate multivariate methods for given problems |

| **Course Outcomes:** | | |
|---|---|---|
| **COs** | **At the end of successful completion of the course, the students will be able to** | **Bloom's Taxonomy** |
| **CO1** | **Explain** the multivariate normal distribution and its relationship to regression models. | Understand |
| **CO2** | **Estimate** parameters and **assess** assumptions in multiple and multivariate regression models. | Analyze |
| **CO3** | **Construct** and **evaluate** linear discriminant functions for classification problems. | Analyze |
| **CO4** | **Construct** principal component and factor analysis for data reduction and interpretation. | Analyze |
| **CO5** | **Implement** clustering algorithms and **interpret** cluster structures in multivariate data. | Analyze |

**Course Description :**

This course provides a comprehensive understanding of multivariate statistical methods used in data analysis. Topics include multivariate normal distribution, multiple and multivariate regression models, discriminant analysis, principal component analysis, factor analysis, and cluster analysis. Emphasis is placed on model assumptions, estimation, interpretation, and diagnostics in real-world data contexts.

| | |
|---|---|
| **Prerequisites :** | • Probability and Statistics<br>• Linear Algebra |

| Section – I | | |
|---|---|---|
| **Unit-1** | **Multivariate Normal Distribution:** Multivariate Normal Distribution Functions, Conditional Distribution and its relation to regression model, Estimation of parameters. | **06Hr** |
| **Unit-2** | **Multiple Linear Regression Model**: Standard multiple regression models with emphasis on detection of collinearity, outliers, non-normality and autocorrelation, Validation of model assumptions. | **06Hr** |
| **Unit-3** | **Multivariate Regression**: Assumptions of Multivariate Regression Models, Parameter estimation, Multivariate Analysis of variance and covariance | **06Hr** |
| **Section-II** | | |
| **Unit-4** | **Discriminant Analysis**: Statistical background, linear discriminant function analysis, Estimating linear discriminant functions and their properties. | **6Hr** |
| **Unit-5** | **Principal Component Analysis & Factor Analysis**: Principal components, Algorithm for conducting principal component analysis, deciding on how many principal components to retain, H-plot.<br>Factor analysis model, Extracting common factors, determining number of factors, Transformation of factor analysis solutions, Factor scores. | **6Hr** |
| **Unit-6** | **Cluster Analysis**: Introduction, Types of clustering, Correlations and distances, clustering by partitioning methods, hierarchical clustering, overlapping clustering, K-Means Clustering- Profiling and Interpreting Clusters | **6Hr** |

**References**

**Text Books :**

1. *An Introduction to Multivariate Statistical Analysis,* T.W. Anderson.
2. *Applied Multivariate Data Analysis, Vol I & II,* J.D. Jobson.

**Reference Books :**

1. *Statistical Tests for Multivariate Analysis,* H. Kris.
2. *Programming Python,* Mark Lutz.
3. *Multivariate Statistical Analysis*, D.F. Morrison.
4. *Python for Data Analysis,* Wes Mc Kinney.

# 25UG-PCC- CB304- Software Engineering

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | 3* Hrs/Week | **ISE** | **:** | 40 Marks |
| **Credits** | **:** | 2 | | | |
| **Tutorials** | **:** | -- | **ESE** | **:** | 60 Marks |

| **Course Objective:** The objective of this course is to | |
|---|---|
| **1** | Understand the significance of software engineering in large-scale system development. |
| **2** | Gain knowledge of project management, lifecycle models, and cost estimation techniques. |
| **3** | Explore quality assurance models and reliability estimation. |
| **4** | Learn effective methods for requirements analysis, modeling, and documentation. |

| **Course Outcomes:** | | |
|---|---|---|
| **COs** | **At the end of successful completion of the course, the students will be able to** | **Bloom's Taxonomy** |
| **CO1** | Describe the evolution, scope, and importance of software engineering in project success. | Understand |
| **CO2** | Explain lifecycle models, planning strategies, cost estimation, and risk management in software projects. | Understand |
| **CO3** | Identify and analyze factors affecting software quality and reliability using standard models. | Analyze |
| **CO4** | Develop and document software requirements and design using structured and object-oriented techniques. | Create |
| **CO5** | Perform systematic software testing and validation techniques to ensure product quality. | Evaluate |

| **Course Description :** | |
|---|---|
| This course offers a structured approach to software development, covering essential principles, models, and practices in software engineering. It includes topics such as project management, quality assurance, requirements engineering, object-oriented design, and testing. The focus is on ensuring software reliability, maintainability, and timely delivery in real-world projects. | |
| **Prerequisites :** | • Basics of Programming<br>• Data Structure |

| Section – I | | |
|---|---|---|
| **Unit-1** | **Introduction:**<br>Programming in the small vs. programming in the large; software project failures and importance of software quality and timely availability; engineering approach to software development; role of software engineering towards successful execution of large software projects; emergence of software engineering as a discipline. | **06Hr** |
| **Unit-2** | **Software Project Management:**<br>Basic concepts of life cycle models – different models and milestones; software project planning – identification of activities and resources; concepts of feasibility study; techniques for estimation of schedule and effort; software cost estimation models and concepts of software engineering economics; techniques of software project control and reporting; introduction to measurement of software size; introduction to the concepts of risk and its mitigation; configuration management. | **06Hr** |
| **Unit-3** | **Software Quality and Reliability:**<br>Internal and external qualities; process and product quality; principles to achieve software quality; introduction to different software quality models like McCall, Boehm, FURPS / FURPS+, Dromey, ISO – 9126; introduction to Capability Maturity Models (CMM and CMMI); introduction to software reliability, reliability models and estimation.<br>concepts of inspection. | **06Hr** |
| **Section-II** | | |
| **Unit-4** | **Software Requirements Analysis, Design and Construction:**<br>Introduction to Software Requirements Specifications (SRS) and requirement elicitation techniques; techniques for requirement modeling – decision tables, event tables, state transition tables, Petri nets; requirements documentation through use cases; introduction to UML, introduction to software metrics and metrics based control methods; measures of code and design quality. | **6Hr** |
| **Unit-5** | **Object Oriented Analysis, Design and Construction**<br>Concepts -- the principles of abstraction, modularity, specification, encapsulation and information hiding; concepts of abstract data type; Class Responsibility Collaborator (CRC) model; quality of design; design measurements; concepts of design patterns; Refactoring; object oriented construction principles; object oriented metrics. | **6Hr** |
| **Unit-6** | **Software Testing: Introduction to faults and failures:**<br>basic testing concepts; concepts of verification and validation; black box and white box tests; white box test coverage – code coverage, condition coverage, branch coverage; basic concepts of black-box tests – equivalence classes, boundary value tests, usage of state tables; testing use cases; transaction based testing; testing for non-functional requirements – volume, performance and efficiency; | **6Hr** |

**References**

**Text Books :**

1. *Software Engineering*, Ian Sommerville.

**Reference Books :**

1. *Fundamentals of Software Engineering,* Carlo Ghezzi, Jazayeri Mehdi, Mandrioli Dino
2. *Software Requirements and Specification: A Lexicon of Practice, Principles and Prejudices*, Michael Jackson
3. *The Unified Development Process*, Ivar Jacobson, Grady Booch, James Rumbaugh
4. *Design Patterns: Elements of Object-Oriented Reusable Software*, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

# 25UG-PCC- CB305 - Object Oriented Programming

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | 3* Hrs/Week | **Practical's** | **:** | -- |
| **Credits** | **:** | 2 | **ISE** | **:** | 40 Marks |
| **Tutorials** | **:** | -- | **ESE** | **:** | 60 Marks |

| **Course Objective:** The objective of this course is to | |
|---|---|
| **1** | To develop proficiency in procedural programming using C. |
| **2** | To introduce object-oriented programming concepts using C++. |
| **3** | To explore advanced features like inheritance, polymorphism, and templates. |
| **4** | To build problem-solving skills through structured and modular coding. |
| **5** | To apply concepts of file handling, memory management, and exception handling. |

**Course Outcomes:**

| **COs** | **At the end of successful completion of the course, the students will be able to** | **Bloom's Taxonomy** |
|---|---|---|
| **CO1** | Explain the syntax, semantics, and features of C and C++ programming languages. | Understand |
| **CO2** | Apply procedural and object-oriented programming techniques to solve real-life problems. | *Apply* |
| **CO3** | Develop modular programs using functions, pointers, and classes. | *Apply* |
| **CO4** | Analyze and implement features like inheritance, polymorphism, and templates in C++. | Analyze |
| **CO5** | Design and debug robust applications using exception handling and file operations. | *Create* |

**Course Description :**

This course introduces the principles of procedural and object-oriented programming using C and C++. It emphasizes core programming constructs, memory management, functions, and object-oriented features such as encapsulation, inheritance, and polymorphism. Students gain practical skills through structured programming tasks and application development using standard libraries, templates, and error-handling mechanisms.

| **Prerequisites :** | • Computers and algorithms<br>• Logic development and flowcharting skills |
|---|---|

| Section – I | | |
|---|---|---|
| **Unit-1** | **Procedural programming, An Overview of C:**<br>Types Operator and Expressions, Scope and Lifetime, Constants, Pointers, Arrays, and References, Control Flow, Functions and Program Structure, Namespaces, error handling, Input and Output (C-way), Library Functions (string, math, stdlib), Command line arguments, Pre-processor directive | **04Hr** |
| **Unit-2** | **Some difference between C and C++:**<br>Single line comments, Local variable declaration within function scope, function declaration, function overloading, stronger type checking, Reference variable, parameter passing – value vs reference, passing pointer by value or reference, #define constant vs const, Operator new and delete, the typecasting operator, Inline Functions in contrast to macro, default arguments | **04Hr** |
| **Unit-3** | **The Fundamentals of Object Oriented Programming**:<br>Necessity for OOP, Data Hiding, Data Abstraction, Encapsulation, Procedural Abstraction, Class and Object. | **04Hr** |
| | | |
| **Unit-4** | **More extensions to C in C++ to provide OOP Facilities:**<br>Scope of Class and Scope Resolution Operator, Member Function of a Class, private, protected and public Access Specifier, this Keyword, Constructors and Destructors, friend class, error handling (exception) | **04Hr** |
| **Unit-5** | **Essentials of Object Oriented Programming:**<br>Operator overloading, Inheritance – Single and Multiple, Class Hierarchy, Pointers to Objects, Assignment of an Object to another Object, Polymorphism through dynamic binding, Virtual Functions, Overloading, overriding and hiding, Error Handling. | **04Hr** |
| **Unit-6** | **Generic Programming:**<br>Template concept, class template, function template, template specialization<br>**Input and Output**: Streams, Files, Library functions, formatted output. | **04Hr** |

**References**

**Text Books :**

1. *The C++ Programming Language*, Bjarne Stroustrup, Addison Wesley.
2. *C++ and Object-Oriented Programming Paradigm*, Debasish Jana, PHI Learning Pvt. Ltd.

**Reference Books :**

1. . *Programming – Principles and Practice Using C++*, Bjarne Stroustrup, Addison Wesley.
2. *The Design and Evolution of C++*, Bjarne Stroustrup, Addison Wesley

# 25UG- MDM1- CB306-Object-Based Programming

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | 2 Hrs/Week | **ISA** | **:** | 25 Marks |
| **Credits** | **:** | 2 | | | |
| **Tutorials** | **:** | -- | **ESE** | **:—** | |

| **Course Objective:** The objective of this course is to | | |
|---|---|---|
| **1** | To develop foundational programming skills using procedural programming techniques in C. | |
| **2** | To understand and implement structured program design including memory management, pointers, and functions. | |
| **3** | To introduce and apply object-oriented concepts using C++ for developing modular and maintainable software. | |
| **Course Outcomes:** | | |
| **COs** | At the end of successful completion of the course, the students will be able to | **Bloom's Taxonomy** |
| **CO1** | Write efficient C programs using appropriate types, operators, expressions, control structures, and modular functions. | Understand |
| **CO2** | Distinguish between procedural and object-oriented approaches, and identify key differences between C and C++. | Analyze |
| **CO3** | Develop C++ programs utilizing classes, objects, constructors/destructors, access specifies, and member functions. | Create |
| **CO4** | Use **O**OP concepts such as encapsulation, abstraction, and function overloading to solve real-world programming problems. | Analyze |

| **Course Description :** | |
|---|---|
| This course offers a comprehensive introduction to procedural programming using the C language, followed by a transition to the object-oriented paradigms of C++. Students will explore fundamental programming constructs such as variables, data types, control flow, functions, pointers, and memory management in C. The course then moves on to the advanced features and extensions provided by C++, emphasizing differences from C and the core concepts of Object-Oriented Programming (OOP), such as encapsulation, abstraction, inheritance, and polymorphism. Students will also learn to use the standard library, handle errors effectively, and manage input/output operations. | |
| **Prerequisites :** | • Computer fundamentals and problem-solving techniques.<br>• Algorithms and flowcharts. |

| Section – I | | |
|---|---|---|
| **Unit-1** | **Procedural programming, An Overview of C:**<br>Types Operator and Expressions, Scope and Lifetime, Constants, Pointers, Arrays, and References, Control Flow, Functions and Program Structure, Namespaces, error handling, Input and Output (C-way), Library Functions (string, math, stdlib), Command line arguments, Pre-processor directive | **04Hr** |
| **Unit-2** | **Some difference between C and C++:**<br>Single line comments, Local variable declaration within function scope, function declaration, function overloading, stronger type checking, Reference variable, parameter passing – value vs reference, passing pointer by value or reference, #define constant vs const, Operator new and delete, the typecasting operator, Inline Functions in contrast to macro, default arguments | **04Hr** |
| **Unit-3** | **The Fundamentals of Object Oriented Programming**:<br>Necessity for OOP, Data Hiding, Data Abstraction, Encapsulation, Procedural Abstraction, Class and Object. | **04Hr** |
| **Unit-4** | **More extensions to C in C++ to provide OOP Facilities:**<br>Scope of Class and Scope Resolution Operator, Member Function of a Class, private, protected and public Access Specifier, this Keyword, Constructors and Destructors, friend class, error handling (exception) | **04Hr** |

**References**

**Text Books :**

1. *The C++ Programming Language*, Bjarne Stroustrup, Addison Wesley.
2. *C++ and Object-Oriented Programming Paradigm*, Debasish Jana, PHI Learning Pvt. Ltd.

**Reference Books :**

1. *Programming – Principles and Practice Using C++*, Bjarne Stroustrup, Addison Wesley.
2. *The Design and Evolution of C++*, Bjarne Stroustrup, Addison Wesley

# 25UG- EEC1- CB3071- Financial Management

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | 2* Hrs/Week | **ISA** | **:** | 25 Marks |
| **Credits** | **:** | 1 | | | |
| **Tutorials** | **:** | -- | **ESE** | **:** | -- |

| **Course Objective:** The objective of this course is to | |
|---|---|
| **1** | To introduce the basic concepts, functions, and goals of financial management. |
| **2** | To develop an understanding of financial planning, capital budgeting, and working capital management. |
| **3** | To enable students to evaluate investment options and financial performance using key tools and techniques. |

| **Course Outcomes:** | | |
|---|---|---|
| **COs** | **At the end of successful completion of the course, the students will be able to** | **Bloom's Taxonomy** |
| **CO1** | Understand the concepts and functions of financial management and its role in decision-making. | Understand |
| **CO2** | Apply financial tools for budgeting, working capital, and capital investment decisions. | Apply |
| **CO3** | Analyze financial performance using ratio analysis and evaluate investment alternatives. | Analyze |

| **Course Description :** |
|---|
| This course introduces students to the fundamental concepts of financial management, including financial analysis, planning, budgeting, investment decisions, and capital management. It aims to equip students with essential tools for sound financial decision-making in business and technology-driven environments. |

| **Prerequisites :** | • . |
|---|---|

| Section – I | | |
|---|---|---|
| **Unit-1** | **Fundamentals of Financial Management:**<br>Nature and scope of financial management, Objectives and functions, Role of financial manager, Financial decision areas – Investment, Financing, Dividend, Financial goal vs. wealth maximization, Time value of money – Present and Future Value, Risk and return trade-off. | **04Hr** |
| **Unit-2** | **Financial Analysis and Planning:**<br>Financial statements – Balance Sheet, Profit & Loss Account, Cash Flow Statement; Ratio analysis – Liquidity, Solvency, Profitability, Efficiency ratios; Fund Flow and Cash Flow analysis; Financial forecasting techniques; Budgeting and Budgetary control. | **04Hr** |
| **Unit-3** | **Investment and Capital Budgeting Decisions:**<br>Capital budgeting process and significance, Investment appraisal techniques – Payback Period, NPV, IRR, ARR, Profitability Index; Risk analysis in capital budgeting; Cost of capital – equity, debt, and WACC; Leverage – Operating, Financial, and Combined. | **04Hr** |
| **Unit-4** | **Working Capital and Financing Decisions:**<br>Concept and classification of working capital, Determinants of working capital, Management of cash, receivables, and inventory; Sources of financing – long-term and short-term; Dividend decision – types of dividends and factors affecting dividend policy. | **04Hr** |

**References**

**Text Books :**

1. **I.M. Pandey**, *Financial Management*, Vikas Publishing House, Latest Edition

**Reference Books :**

1. **Prasanna Chandra**, *Financial Management: Theory and Practice*, Tata McGraw-Hill
2. **Eugene F. Brigham & Joel F. Houston**, *Fundamentals of Financial Management*, Cengage Learning
3. **Khan and Jain**, *Financial Management*, Tata McGraw-Hill

# 25UG- VEC1- CB3072- Ethics and Values in Business and Technology

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | 2 Hrs/Week | **ISA** | **:** | 25 Marks |
| **Credits** | **:** | 2 | | | |
| **Tutorials** | **:** | -- | **ESE** | **:** | -- |

| **Course Objective:** The objective of this course is to | |
|---|---|
| **1** | To understand the importance of ethics and human values in business and professional contexts. |
| **2** | To explore ethical challenges in emerging technologies and corporate practices. |
| **3** | To develop decision-making skills to address moral and professional dilemmas. |

| **Course Outcomes:** | | |
|---|---|---|
| **COs** | **At the end of successful completion of the course, the students will be able to** | **Bloom's Taxonomy** |
| **CO1** | Explain key ethical principles and human values relevant to business and professional environments. | Understand |
| **CO2** | Identify ethical issues in business operations and technological applications including AI, data privacy, and cybersecurity. | Understand |
| **CO3** | Apply ethical frameworks and decision-making models to analyze real-world cases and propose responsible solutions. | Apply |

| **Course Description :** | |
|---|---|
| This course aims to sensitize students to the ethical and value-based challenges encountered in the domains of business and technology. It introduces frameworks for ethical decision-making, explores professional responsibilities, and highlights the role of ethics in technology development, data privacy, and corporate governance. | |
| **Prerequisites :** | • Societal and professional responsibilities<br>• Current trends in business and technology |

| Section – I | | |
|---|---|---|
| **Unit-1** | **Introduction to Ethics and Human Values**<br>Definition, nature, and importance of ethics and values; types of values – personal, cultural, organizational; moral dilemmas and ethical decision-making; ethics vs law; value system in professional life. | **04Hr** |
| **Unit-2** | **Business Ethics**<br>Principles of business ethics; ethics in corporate governance; ethical issues in marketing, finance, and HR; workplace ethics – discrimination, harassment, transparency, whistleblowing; CSR (Corporate Social Responsibility) and sustainability. | **04Hr** |
| **Unit-3** | **Technology and Ethics**<br>Technology as a double-edged sword; ethical concerns in AI, data privacy, cyber security, and automation; intellectual property rights (IPR); ethical software development; professional codes of ethics (ACM, IEEE). | **04Hr** |
| **Unit-4** | **Case Studies and Emerging Issues** Case studies on ethical dilemmas in business and tech (e.g., Facebook-Cambridge Analytical, Theranos, Uber, facial recognition); environmental ethics in technology; ethics of surveillance and social media; digital well-being and responsible innovation. | **04Hr** |

**References**

**Text Books :**

1. Business Ethics: Concepts and Cases, Manuel G. Velasquez, Pearson Education.

**Reference Books :**

2. Ethics in Information Technology, George W. Reynolds, Cengage Learning.
3. Professional Ethics, R. Subramanian, Oxford University Press.
4. Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing, Herman T. Tavani, Wiley.

# 25UG-CEP- CB3073- Community Engagement Activity (CEA)/Field Project

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | -- | **ISA** | **:** | 25 Marks |
| **Credits** | **:** | 2 | | | |
| **Tutorials** | **:** | -- | **ESE** | **:** | -- |

| **Course Objective:** The objective of this course is to | |
|---|---|
| **1** | Understand community challenges and societal issues. |
| **2** | Apply computational thinking and business knowledge to real-world scenarios. |
| **3** | Build empathy, leadership, and project management skills. |
| **4** | Promote sustainable development through technology and innovation. |

| **Course Outcomes:** | | |
|---|---|---|
| **COs** | **At the end of successful completion of the course, the students will be able to** | **Bloom's Taxonomy** |
| **CO1** | Identify and analyze real-world community problems through field research. | Analyze |
| **CO2** | Design appropriate solutions using computer science and business principles to address community needs. | Create |
| **CO3** | Collaborate effectively with diverse stakeholders and team members in executing the field project. | Apply |
| **CO4** | Evaluate the impact of the implemented solution and reflect on ethical, social, and professional responsibilities. | Evaluate |

**Course Description :**

This course emphasizes experiential learning through real-world community engagement or field projects. Students identify social issues, design technology-driven or business-oriented solutions, and collaborate with stakeholders to implement and assess impact, fostering civic responsibility, innovation, and interdisciplinary application of CSBS knowledge.

| **Prerequisites :** | • Computer Science fundamentals<br>• Exposure to Business Communication and Ethics |
|---|---|

| **Section – I** | | |
|---|---|---|
| **Unit-1** | The Community Engagement Activity (CEA) or Field Project is a practical component of the CSBS curriculum aimed at bridging classroom learning with real-world societal challenges. It encourages students to identify, analyze, and address local community issues through active participation, fieldwork, and innovative problem-solving. The course provides exposure to areas such as digital literacy, environmental sustainability, healthcare awareness, financial inclusion, and social entrepreneurship. Through collaborative efforts, students apply their technical and business knowledge to design and implement impactful solutions, fostering empathy, leadership, and a sense of civic responsibility. The activity culminates in a detailed project report and presentation, showcasing the students' engagement, outcomes achieved, and learning reflections. | |

# 25UG-PCC- CB302P - Computer Organization and Architecture Lab

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | -- | **Practical's** | **:** | 2 Hrs/Week |
| **Credits** | **:** | 1 | **ISA** | **:** | 25 Marks |
| **Tutorials** | **:** | -- | **ESE** | **:** | -- |

**Course Description :**

The Computer Organization and Architecture Lab provides hands-on experience with the fundamental concepts of computer systems. Through practical experiments, students explore instruction execution, data representation, arithmetic operations, memory organization, control unit design, and I/O mechanisms. The lab reinforces theoretical knowledge by enabling students to simulate and analyze the internal functioning of a computer, enhancing their understanding of system architecture and performance.

**Experiment List**

| Expt. No. | Experiment Title | Bloom's Taxonomy |
|---|---|---|
| **1** | Simulate Instruction Execution Cycle | Understand |
| **2** | Number System Conversion (Binary, Decimal, Hex) | Apply |
| **3** | Implementation of Addressing Modes | Apply |
| **4** | Design and Simulation of Ripple Carry Adder | Apply |
| **5** | Booth's Multiplication Algorithm Simulation | Analyze |
| **6** | Floating Point Representation and IEEE 754 Format | Understand |
| **7** | Simulation of CPU Control Unit (Hardwired or Micro programmed) | Create |
| **8** | Demonstration of Interrupt-Driven I/O and DMA Mechanisms | Understand |
| **9** | Pipeline Simulation and Hazard Detection | Analyze |
| **10** | Cache Memory Mapping and Replacement Policies Simulation | Apply |

# 25UG-PCC- CB305P - Object Oriented Programming Lab

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | : | 2* Hrs/Week | **Practical's** | : | 2 Hrs/Week |
| **Credits** | : | 1 | **ISA** | : | 25 Marks |
| **Tutorials** | : | -- | **POE** | : | 50 Marks |

| **Course Description :** | |
|---|---|
| This course introduces the principles of procedural and object-oriented programming using C and C++. It emphasizes core programming constructs, memory management, functions, and object-oriented features such as encapsulation, inheritance, and polymorphism. Students gain practical skills through structured programming tasks and application development using standard libraries, templates, and error-handling mechanisms. | |
| **Prerequisites :** | • Computers and algorithms<br>• Logic development and flowcharting skills |

**Experiment List**

| Expt. No. | Experiment Title | Bloom's Taxonomy |
|---|---|---|
| 1 | Write a program in C++ to demonstrate use of data types, operators, and control structures. | Understand |
| 2 | Write a C++ program to implement classes and objects with data hiding and encapsulation. | Apply |
| 3 | Implement function overloading and operator overloading in C++. | Apply |
| 4 | Design a C++ program using constructors, destructors, and dynamic memory allocation. | Create |
| 5 | Write a program to implement single and multiple inheritances in C++. | Apply |
| 6 | Demonstrate runtime polymorphism using virtual functions and base class pointers. | Analyze |
| 7 | Implement class and function templates in C++ with specialization. | Apply |
| 8 | Develop a file handling program in C++ using stream and exception handling. | Create |
| 9 | Write a C++ program to use inline functions, default arguments, and reference variables. | Apply |
| 10 | Implement Macro and Inline Function to Compare Their Behavior in C++ | Apply |

**References**

**Text Books :**

1. *The C++ Programming Language*, Bjarne Stroustrup, Addison Wesley.
2. *C++ and Object-Oriented Programming Paradigm*, Debasish Jana, PHI Learning Pvt. Ltd.

# 25UG-PCC- CB303P - Computational Statistics Lab

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | -- | **Practical's** | **:** | 2 Hrs./Week |
| **Credits** | **:** | 1 | **ISA** | **:** | 25 Marks |
| **Tutorials** | **:** | -- | **POE** | **:** | 50Marks |

**Course Description :**

This laboratory course introduces students to Python programming fundamentals, data structures, file handling, and visualization using Matplotlib. It also covers the implementation of multivariate data analysis techniques including regression, clustering, and dimensionality reduction using real-world datasets. The lab focuses on practical application of statistical and machine learning methods for data-driven problem solving.

**Experiments List**

| Expt. No. | Experiment Title | Bloom's Taxonomy |
|---|---|---|
| **1** | Write Python programs to demonstrate flow control, loops, and functions. | Apply |
| **2** | Implement classes, objects, and constructors in Python. | Apply |
| **3** | Perform file handling operations (text and binary files) in Python. | Apply |
| **4** | Visualize data using line plots, bar charts, and scatter plots with Matplotlib. | Apply |
| **5** | Customize graphs by adding text, titles, legends, and changing axes with Matplotlib. | Analyze |
| **6** | Perform simple linear and multiple regression analysis on real-world datasets. | *Apply* |
| **7** | Implement multivariate regression using scikit-learn on a real dataset. | *Apply* |
| **8** | Conduct cluster analysis using K-Means and Hierarchical Clustering algorithms. | *Analyze* |
| **9** | Apply Principal Component Analysis (PCA) for dimensionality reduction and visualization. | Analyze |
| **10** | Perform Linear Discriminant Analysis (LDA) and interpret class separation. | *Evaluate* |

# 25UG-PCC- CB304P –Software Engineering Lab

| Teaching Scheme | | | Evaluation Scheme | | |
|---|---|---|---|---|---|
| **Lectures** | **:** | -- | **Practical's** | **:** | 2 Hrs./Week |
| **Credits** | **:** | 1 | **ISA** | **:** | 25 Marks |
| **Tutorials** | **:** | -- | **POE** | **:** | -- |

| **Course Description :** |
|---|
| This lab course provides hands-on experience in software engineering practices such as requirement analysis, system modeling using UML diagrams, and software testing. Students will use Agile tools like Jira or Azile for project tracking, estimation, and implementation of a chosen system |

**Experiments List**

| Expt. No. | Experiment Title | Bloom's Taxonomy |
|---|---|---|
| **1** | Write down the problem statement for a suggested system of relevance project. | Understand |
| **2** | Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system. | Apply |
| **3** | To perform the function oriented diagram: Data Flow Diagram (DFD) and Structured chart. | Apply |
| **4** | To perform the user's view analysis for the suggested system: Use case diagram. | Apply |
| **5** | To draw the structural view diagram for the system: Class diagram, object diagram. | Apply |
| **6** | To draw the behavioral view diagram : State-chart diagram, Activity diagram | Apply |
| **7** | To perform the behavioral view diagram for the suggested system : Sequence diagram, Collaboration diagram | Apply |
| **8** | To perform the implementation view diagram: Component diagram for the system. | Apply |
| **9** | To perform the environmental view diagram: Deployment diagram for the system. | Apply |
| **10** | To perform various testing using the testing tool unit testing, integration testing for a sample code of the suggested system. | Analyze |
| **11** | To Perform Estimation of effort using FP Estimation for chosen system. | Analyze |
| **12** | To Prepare time line chart/Gantt Chart/PERT Chart for selected software | Create |

Member Secretary Board of Studies | Chairman Board of Studies | Academic Dean TKIET,Wrananagar | Dean S.E.T.M | Principal TKIET,Wrananagar