Bharati Vidyapeeth

(Deemed to be University)

Pune, India

College of Engineering, Pune

B.Tech. Computer Science and Business Systems (2023 Course)

Program Curriculum

As Per NEP Guidelines

VISION OF UNIVERSITY:

Social Transformation through Dynamic Education

MISSION OF UNIVERSITY:

- To make available quality education in different areas of knowledge to the students as per their choice and inclination.
- To offer education to the students in a conducive ambiance created by enriched infrastructure! and academic facilities in its campuses.
- To bring education within the reach of rural, tribal and girl students by providing them substantive fee concessions and subsidized hostel and mess facilities.
- To make available quality education to the students of rural, tribal and other deprived sections of the population

VISION OF THE INSTITUTE:

To be World Class Institute for Social Transformation Through Dynamic Education.

MISSION OF THE INSTITUTE:

- To provide quality technical education with advanced equipment, qualified faculty members, infrastructure to meet needs of profession and society.
- To provide an environment conducive to innovation, creativity, research and entrepreneurial leadership.
- To practice and promote professional ethics, transparency and accountability for social community, economic and environmental conditions.

VISION OF THE DEPARTMENT

To syndicate industry and institute to impart high quality knowledge through scholarity, research and creative endeavour

MISSION OF THE DEPARTMENT

- To impart contemporary technology conforming to a dynamic curriculum.
- To engage in professional development and scholarly endeavour through knowledge of common business principles.
- To promote the awareness of business discipline and ethical responsibility through industry alliance

Program Educational Objectives (PEOs)

1. Prevail technical competency to concord the industry engrossment.
2. Assimilate business management skills.
3. Instigate business level innovation with societal consideration.

Program Specific Outcomes (PSOs)

Students of B. Tech (CSBS) will be

PSO1: Able to apply pragmatic, innovative and critical thinking approach for solving complex business problems.

PSO2: Able to choose effective business communication techniques in professional

Institute/organization.

PSO3: Able to use financial domain understanding to formulate technological strategy.

PSO4: Skilled in contemporary courses from emerging domains such as artificial intelligence, Machine learning and data science.

Program Outcomes (POs)

The students of B.Tech (Computer Science & Business Systems) will be able to

a. Demonstrate logical and programming skills through comprehensive programming foundation.

b. Apply knowledge of mathematics, computer engineering and basic science to comprehend and solve real world problems.

c. Develop software applications and processes for complex problems to provide efficient solutions by assessing its environmental, social and ethical constraints.

d. Investigate and solve complex computing problems with alternate solutions.

e. Use functional skills of modern IT tools and techniques for engineering activities.

f. Understand the social and cultural impact of computing on society.

g. Provide optimized computational solutions that apprehend the societal and environmental aspects.

h. Exhibit the professional, ethical and legal responsibilities related to industry.

i. Perform as an individual and efficient team player to accomplish a goal.

j. Present professional concepts through effective communication skills and documentation.

k. Demonstrate management skills for developing time-bound projects within the available budget and resources.

l. Develop the ability of lifelong learning for new IT practices.

A. DEFINITION OF CREDITS:

1 Hour Lecture (L) per week	1 credit
1 Hour Tutorial (T) per week	1 credit
1 Hour Practical (P) per week	0.5 credits
2 Hours Practical (Lab)/week	1 credit

B. COURSE CODE AND DEFINITION

Course Code	Definitions
L	Lecture
T	Tutorial
P	Practical
TW	Term Work
O	Oral
SEE	Semester End Examination
MJ	Major (Core) Courses
MI	Minor Courses
GE	General Elective Courses
OE	Open Elective Courses
SE	Skill Enhancement Courses
AE	Ability Enhancement Courses
VE	Vocational Enhancement Courses
VS	Vocational Skill Courses
MAC	Mandatory Credit Course
VA	Value Added Courses
CC	Co-curricular Courses
EC	Extra-Curricular Courses
ID	Inter-disciplinary Courses

MD	Multidisciplinary Courses
RP	Research I Project Courses
PC	Practical Courses
BS	Basic Science
ES	Engineering Science
AC	Audit Course
EC	Extracurricular Activities
BM	Basic Mathematics
BP	Basic Physics
BC	Basic Chemistry
UH	Universal Human Values
ET	Electrical Technology

BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)
COLLEGE OF ENGINEERING, PUNE
B. Tech. (CSBS): Semester – III (NEP 2020 COURSE)

Sr. No.	Category	Subject Code	Subject	Teaching Scheme			Examination Scheme-Marks						Credits			
				L	P	T	ESE	IA	TW	OR	PR	Total	L	P	T	Total
1.	MJ	MJ1106301	Formal Language and Automata Theory	3	-	1	60	40	25	-	-	125	3	-	1	4
2.	ID	ID1106302	Computer Organization and Architecture	3	2	-	60	40	25	-	-	125	3	1	-	4
3.	MJ	MJ1106303	Software Engineering	3	2	-	60	40	25	25	-	150	3	1	-	4
4.	MJ	MJ1106304	Computational Statistics	3	-	-	60	40	-	-	-	100	3	-	-	3
5.	MJ	MJ1106305	Object Oriented Programming	3	2	-	60	40	25	-	25	150	3	1	-	4
6.	SE	SE1106306	Skill Based Course III (Data Science with Python)	-	2	-	-	-	25	-	25	50	-	1	-	1
			Total	**15**	**8**	**1**	**300**	**200**	**125**	**25**	**50**	**700**	**15**	**4**	**1**	**20**
7.	AC	AC1106307	Indian Knowledge System	2	-	-	-	100	-	-	-	-	-	-	-	2
8.	VA	VA1106308	Value Added Course- I	2	-	-	-	100	-	-	-	-	-	-	-	2

BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)
COLLEGE OF ENGINEERING, PUNE
B. Tech. (CSBS): Semester – IV (NEP 2020 COURSE)

Sr. No.	Category	Subject Code	Subject	Teaching Scheme			Examination Scheme-Marks						Credits			
				L	P	T	ESE	IA	TW	OR	PR	Total	L	P	T	Total
1.	MJ	MJ1106401	Operating Systems	3	2	-	60	40	25	25	-	150	3	1	-	4
2.	MJ	MJ1106402	Database Management Systems	3	2	-	60	40	25	-	25	150	3	1	-	4
3.	ID	ID1106403	Introduction to Innovative IP Management & Entrepreneurship	2	-	-	60	40	-	-	-	100	2	-	-	2
4.	MJ	MJ1106404	Design Thinking	3	2	-	60	40	25	-	-	125	3	1	-	4
5.	MJ	MJ1106405	Operation Research	3	2	-	60	40	25	-	-	125	3	1	-	4
6.	SE	SE1106406	Skill Based Course IV (Software Design with UML)	-	2	1	-	-	25	-	25	50	-	1	1	2
			Total	**14**	**10**	**1**	**300**	**200**	**125**	**25**	**50**	**700**	**14**	**5**	**1**	**20**
7.	AE	AE1106407	MOOC - I	-	-	-	-	-	-	-	-	-	-	-	-	2
8.	EC	EC1106408	Social Activity	-	-	-	-	-	-	-	-	-	-	-	-	2

B. TECH (Computer Science & Business Systems)

SEMESTER – III

COURSE SYLLABUS

FORMAL LANGUAGE & AUTOMATA THEORY

TEACHING SCHEME	EXAMINATION SCHEME		CREDITS ALLOTTED	
Lectures :3Hrs./Week	**Semester Examination**	**:60 marks**	**Theory**	**:3 Credits**
Tutorials :1Hr./Week	**Internal Assessment**	**:40 marks**	**Tutorial**	**:1 Credit**
	Term Work	**:25 Marks**	**Total**	**:4 Credits**
	Total	**:125 marks**		

Course Pre-Requisites:

The students should have basic Knowledge Set algebra, elementary formal logic, constructing proofs, recurrence relations, Discrete Structures and Data structures and problem solving.

Course Objective:

1) To understand problem classification and problem solving by machines.
2) To understand the basics of automata theory and its operations.
3) To study computing machines by describing, classifying, and comparing different types of computational. models.
4) Encourage students to study theory of computability and complexity.
5) To understand the P and NP class problems and its classification.
6) To understand the fundamentals of problem decidability and reducibility.

Course Outcomes:

1) Understand the Graphs and Trees related concepts.
2) Design the Finite Automata Machines and its operations.
3) Understand Context Free Grammar and Context Free languages.
4) Construct Turing Machine for formal languages.
5) Identify and Design the equivalence of languages described by Pushdown Automata
6) Identify NP Hard and NP- Complete problems.

UNIT – I **6 Hours**

Introduction: Alphabet, Strings and languages, Graphs, Directed Graphs, Trees, FSM.

UNIT – II **6 Hours**

Regular Languages and Finite Automata: 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, Keene's theorem, pumping lemma for regular languages, Myhill-Nerode theorem and its uses, minimization of finite automata.

UNIT – III **6 Hours**

Context-Free Languages and Pushdown Automata: Productions and Derivation, Context-free grammars (CFG) and languages (CFL), Chomsky hierarchy of languages, Chomsky Normal Forms and Greibach normal forms, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.
Context-Sensitive Languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

UNIT – IV **6 Hours**

Turing Machines: 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.

UNIT – V **6 Hours**

Push Down Automata (PDA): Introduction, Pushdown Automata (PDA), Transition Diagrams, Functions and Tables, Deterministic Push- down Automata (DPDA) - definition, Nondeterministic Pushdown Automata (NPDA), Equivalence of context-free grammars and PDA, properties of context-free languages.

UNIT – VI **6 Hours**

Basic Introduction to Complexity: Introductory ideas on Time complexity of deterministic and nondeterministic Turing machines, P and NP, NP-completeness, Cook's Theorem, and other NP-Complete problems.

Undecidability: Church-Turing thesis, Universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.

List of Assignments:

YACC, the parser-generating tool (Chapter 5 of Introduction to Automata Theory, Languages, and Computation (John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman.)

List of Project Based Learning Topics:

1. Design a FA for Vending Machine
2. Explain Pigeon hole Principle
3. Implement Push Down Automata
4. Implement Regular Expression
5. Implement lexical Analyzer
6. Implement Turing Machine for Mathematical Expression
7. Design an application to search a string from given text using FA
8. Implement a FSM for residing mod 3.
9. Provide solutions for Missionaries and Cannibals problems.

Textbooks:

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

Reference Books:

1. Elements of the Theory of Computation, Harry R. Lewis and Christos H. 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.
5. Computers and Intractability: A Guide to the Theory of NP Completeness, M. R. Garey and D. S. Johnson.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

COMPUTER ORGANIZATION & ARCHITECTURE

TEACHING SCHEME	**EXAMINATION SCHEME**	**CREDITS ALLOTTED**
Theory :3 Hours / Week	**End Semester Examination :60 Marks**	**Theory :3 Credits**
Lab : 2 Hours / Week	**Internal Assessment :40 Marks**	**Practical :1 Credit**
	Term work :25 Marks	
	Total :125 Marks	**Total :4 Credits**

Course Pre-Requisites:

The students should have basic Knowledge Digital electronics and computer system.

Course Objective:

To understand the design of the various functional units of computer system.

Course Outcomes:

After completion of this course students will be able to

1. Identify various components of a computer and their interconnection.
2. Explain the functions & organization of various blocks of CPU.
3. Understand CPU instruction characteristics, enhancement features of CPU.
4. Describe an assortment of memory types (with their characteristics) used in computer systems and basic principle of interfacing input and output devices.
5. Describe the I/O organization and interconnections.
6. Infer parallel processing and multiprocessor configuration.

UNIT – I **6 Hours**

Revision of basics in Boolean logic and Combinational/Sequential Circuits.

Functional Blocks of a Computer: CPU, memory, input-output subsystems, control unit.

Introduction to x86 Architecture

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.

UNIT – II **6 Hours**

Data Representation: Signed number representation, fixed and floating-point representations, character representation.

Computer Arithmetic: 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.

UNIT – III **6 Hours**

CPU Control Unit Design: Hardwired and micro-programmed design approaches, design of a simple hypothetical CPU.

UNIT – IV **6 Hours**

Memory System Design: Semiconductor memory technologies, memory organization.

Memory Organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.

UNIT – V **6 Hours**

Peripheral Devices and Their Characteristics: 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 – SCSI, USB

UNIT – VI **6 Hours**

Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards.

Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency.

List of Assignments:

Assignments covering the following topics should be given

1. Booth's algorithm for multiplication
2. Restoring and non-restoring division
3. Fixed point and floating-point representation
4. Programmer's model of 80386
5. Hardwired and micro-programmed design approaches.
6. Characteristics of Memory system
7. Cache organization and address mapping
8. Virtual memory and replacement algorithms
9. Calculating throughput and speed in pipelining
10. Multiprocessor architecture

List of Laboratory Experiments

1. Boolean Logic Gates Implementation: Design and simulate basic logic gates (AND, OR, NOT, etc.) using software like Logisim or Verilog.
2. Write assembly language program for simple tasks like arithmetic operations, control flow using x86 instructions.
3. Fixed and Floating-Point Representations: Study the representation of numbers in fixed-point and floating-point formats. Perform arithmetic operations on fixed-point and floating-point numbers.
4. Hardwired Control Unit Design: Design a hardwired control unit for a simple CPU architecture. Implement control signals for various instructions.
5. Semiconductor Memory Technologies: Study different semiconductor memory technologies (DRAM, SRAM, Flash, etc.) and their characteristics.
6. Cache Memory Simulation: Simulate cache memory with varying cache sizes, block sizes, mapping functions, and replacement algorithms. Analyse cache hit/miss rates and access time.
7. I/O Device Interface Implementation: Implement interfaces for different I/O devices (e.g., keyboard, mouse, display) using programming languages like C or Python.
8. Introduction to Parallel Processors: Study the architecture and organization of parallel processors. Analyze the benefits of parallelism in terms of performance and scalability.

List of Project Based Learning Topics:

1. Automatic night lamp with morning alarm
2. Traffic light with sensor + 7segment
3. Multi pattern running lights.
4. Washing machine
5. Simple Lock Using Keypad and 7 segment
6. Electronic quiz table
7. Electronic Digital Clock
8. Temperature controller
9. Plant Irrigation System
10. Car Parking Management
11. Customer counter for supermarket
12. Electronic queue management system in food stall
13. Safety box
14. Shop lot automatic door with 7segment display
15. Bank queue management system
16. Water level controller
17. Automatic home system
18. Commuter system
19. Automatic room light control
20. Elevator control system

Textbooks:

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

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

SOFTWARE ENGINEERING

TEACHING SCHEME	**EXAMINATION SCHEME**	**CREDITS ALLOTTED**
Lectures :3 Hrs./Week	**Semester Examination :60 Marks**	**Theory :3 Credits**
Lab :2Hrs./Week	**Internal Assessment :40 Marks**	
	Term work :25 Marks	**Practical :1 Credit**
	Oral :25 Marks	
	Total :150 Marks	**Total :4 Credits**

Course Pre-Requisites:

The students should have sound knowledge of data structures, programming experience and an extensive hands-on experience of using software.

Course Objective:

The course introduces key aspects of software engineering processes for the development of a complex software system.

Course Outcomes:

1. Learn importance of software engineering process and its principles
2. Understand the software development life cycle with appropriate models
3. Understand software quality concepts
4. Document user requirements using suitable techniques
5. Analyze the software design from and Object-Oriented perspective.
6. Apply appropriate testing techniques on a software

UNIT – I **6 Hours**

Introduction: 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.

UNIT – II **6 Hours**

Software Project Management: 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.

UNIT – III **6 Hours**

Software Quality and Reliability: 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.

UNIT – IV **6 Hours**

Software Requirements Analysis, Design and Construction: 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.

UNIT – V **6 Hours**

Object Oriented Analysis, Design and Construction: 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.

UNIT – VI **6 Hours**

Software Testing: Introduction to faults and failures; 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; concepts of inspection.

List of Assignments:

Teaching faculty will design home assignment on following topics.

1. Software development Models
2. Software Requirement Specification
3. Data Flow Diagrams
4. Testing
5. Object Oriented Analysis, Design and Construction
6. Software project covering various software development methodology techniques will be implemented.

List of Laboratory Exercises:

1. Develop Flow-Charts for (any open-ended problem statement) to understand basic problem-solving technique using suitable tool.
2. Perform domain analysis for given problem.
3. Develop requirements specification document as per IEEE format for a given problem
4. Develop DFD model (level-0, level-1 DFD and Data dictionary) of the project under consideration.
5. Perform Structured design for the developed DFD model.
6. Calculate Cyclomatic complexity for given code snippet.
7. Identify the usage of regression testing.
8. Identify the different types of performance testing

List of Project Based Learning Topics:

1. Fingerprint voting system
2. Weather forecasting system
3. Android local train ticketing system
4. Railway tracking and arrival time prediction system
5. Android Patient Tracker
6. Opinion mining for social networking platforms
7. Automated payroll system with GPS tracking and image capture

8. Data leakage detection system
9. Credit card fraud detection
10. AI shopping system
11. Camera motion sensor system
12. Bug tracker
13. e-Learning platform
14. Smart health prediction system
15. Software piracy protection system

Text Books:

1. Software Engineering, Ian Sommerville
2. Object Oriented Software Engineering: A Use Case Driven Approach --Ivar Jacobson

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, MichaelJackson
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
5. Software Metrics: A Rigorous and Practical Approach, Norman E Fenton, Shari Lawrence Pfleeger
6. Software Engineering: Theory and Practice, Shari Lawrence Pfleeger and Joanne M. Atlee
7. Object-Oriented Software Construction, Bertrand Meyer
8. Object Oriented Software Engineering: A Use Case Driven Approach --Ivar Jacobson
9. Touch of Class: Learning to Program Well with Objects and Contracts --Bertrand Meyer
10. UML Distilled: A Brief Guide to the Standard Object Modeling Language --Martin Fowler

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

COMPUTATIONAL STATISTICS

TEACHING SCHEME	EXAMINATION SCHEME		CREDITS ALLOTTED
Theory :3 Hours / Week	**End Semester Examination**	**:60 Marks**	**Theory :3 Credits**
Lab :0 Hours / Week	**Internal Assessment**	**:40 Marks**	
	Total	**: 100 Marks**	**Total :3 Credits**

Course Pre-requisites: The Students should have knowledge of basics of statistics.

Course Objectives:

The aim of this course is to give graduate students knowledge of statistical concepts like factor analysis, regression analysis and Python programming. The course objective is to exercise students for data set handling, data wrangling, data visualization etc. using Python.

Course Outcomes:

1. Learn methods of Data Wrangling.
2. Understand importance of different Group Operations and Time series using real time datasets.
3. Apply knowledge of Normal Distribution and Design Simple and Multiple Linear Regression Models.
4. Create Multivariate Regression Model and Understand Discriminant Analysis
5. Understand and Demonstrate Dimension Reduction Techniques.
6. Demonstrate the use of Clustering on real time datasets.

UNIT – I **6 Hours**

Python Concepts, Data Structures, Classes: Interpreter, Program Execution, Statements, Expressions, Flow Controls, Functions, Numeric Types, Sequences and Class Definition, Constructors, Text & Binary Files - Reading and Writing.
Data Wrangling: Combining and Merging Datasets, Reshaping and Pivoting, Data Transformation, String Manipulation, Regular Expressions

UNIT – II **6 Hours**

Data Aggregation, Group Operations, Time Series: GroupBy Mechanics, Data Aggregation, Groupwise Operations and Transformations, Pivot Tables and Cross Tabulations, Time Series Basics, Data Ranges, Frequencies and Shifting.

UNIT – III **6 Hours**

Multivariate Normal Distribution: Multivariate Normal Distribution Functions, Conditional Distribution and its relation to regression model, Estimation of parameters.
Multiple Linear Regression Model: Standard multiple regression models with emphasis on detection of collinearity, outliers, non-normality and autocorrelation, Validation of model assumptions.

UNIT – IV **6 Hours**

Multivariate Regression: Assumptions of Multivariate Regression Models, Parameter estimation, Multivariate Analysis of variance and covariance.
Discriminant Analysis: Statistical background, linear discriminant function analysis, Estimating linear discriminant functions and their properties.

UNIT - V **6 Hours**

Principal Component Analysis: Principal components, Algorithm for conducting principal component analysis, deciding on how many principal components to retain, H-plot.
Factor Analysis: Factor analysis model, extracting common factors, determining number of factors, Transformation of factor analysis solutions, Factor scores.

UNIT – VI **6 Hours**

Clustering and Segmentation Analysis: Introduction, Types of clustering, Correlations and distances, clustering by partitioning methods, hierarchical clustering, overlapping clustering, K-Means Clustering-Profiling and Interpreting Clusters.

List of Assignments:
Respective subject teacher shall design any six assignments on above units.

Textbooks:

1. An Introduction to Multivariate Statistical Analysis, T.W. Anderson.
2. Applied Multivariate Data Analysis, Vol I & II, J.D. Jobson.
3. Statistical Tests for Multivariate Analysis, H. Kris.
4. Programming Python, Mark Lutz.
5. Python 3 for Absolute Beginners, Tim Hall and J-P Stacey.
6. Beginning Python: From Novice to Professional, Magnus Lie Hetland. Edition, 2005.

List of Project Based Learning Topics:

1. Design and development of Student management system using object-oriented approach and file structure.
2. Development of student performance analysis system (Use of file, OO Python and regression model, Graphical dashboard).
3. Development of multivariate predictive model for rain forecasting (use rainfall data for last 50 years).
4. Development of multivariate predictive model for gold rate. (Use daily gold rate data for last 10 years).
5. Development of multivariate predictive model for patrol rate. (Use daily patrol rate data for last 10 years).
6. Comparative analysis of predictions of single multivariate predictive model against multiple linear predictive models.
7. Comparative analysis of dimensionality reduction performance using principle component analysis (PCA) and linear discriminant analysis (LDA).
8. Comparative analysis of classification performance of principle component analysis (PCA) and linear discriminant analysis (LDA) techniques.
9. Study of effectiveness of analysis of variance (ANOVA) and analysis of covariance (ANCOVA) for predictive analysis.
10. Comparing operating differences of various clustering Techniques.
11. Comparative analysis of performance for parameter (variable/factors) selection using principal component analysis (PCA) and factor analysis (FA) for multivariate analysis.

Reference Books:

1. Regression Diagnostics , Identifying Influential Data and Sources of Collinearety, D.A. Belsey, E. Kuh and R.E. Welsch
2. Applied Linear Regression Models, J. Neter, W. Wasserman and M.H. Kutner.
3. The Foundations of Factor Analysis, A.S. Mulaik.
4. Introduction to Linear Regression Analysis, D.C. Montgomery and E.A. Peck.
5. Cluster Analysis for Applications, M.R. Anderberg.
6. Multivariate Statistical Analysis, D.F. Morrison.
7. Python for Data Analysis, Wes Mc Kinney.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT –IV

OBJECT ORIENTED PROGRAMMING

TEACHING SCHEME	**EXAMINATION SCHEME**		**CREDITS ALLOTTED**
Theory :3 Hours / Week	**End Semester Examination**	**:60 Marks**	**Theory :3 Credits**
Lab :2 Hours / Week	**Internal Assessment**	**:40 Marks**	**Practical: 1 Credit**
	Termwork	**:25 Marks**	
	Practical	**:25 Marks**	**Total :4 Credits**
	Total	**: 150 Marks**	

Course Pre-Requisites:

The students should have basic Knowledge of “C” programming language.

Course Objective:

The course introduces fundamental concepts of Object-oriented programming.

Course Outcomes:

At the end of this course students will be able to:

1. Understand basic concepts of Procedural programming and the overview of C programming language.
2. Understand some basic differences between C and C++.
3. Understand basic concepts of Object-Oriented Programming, classes and objects in OOP.
4. Apply the concept of Access Specifier, friend function, constructor, destructor and Error Handling using C++ programs.
5. Implement the concept of polymorphism, virtual functions and inheritance using C++.
6. Develop OOP applications using Templates and file Handling.

UNIT – I

6 Hours

Procedural Programming, An Overview of C: 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

UNIT – II

6 Hours

Some Difference between C and C++: 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.

UNIT – III **6 Hours**

The Fundamentals of Object-Oriented Programming: Necessity for OOP, Data Hiding, Data Abstraction, Encapsulation, Procedural Abstraction, Class and Object.

UNIT – IV **6 Hours**

More Extensions to C in C++ to Provide OOP Facilities: 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)

UNIT – V **6 Hours**

Essentials of Object-Oriented Programming: 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.

UNIT – VI **6 Hours**

Generic Programming: Template concept, class template, function template, template specialization
Input and Output: Streams, Files, Library functions, formatted output
Object Oriented Design and Modelling: UML concept, use case for requirement capturing, Class diagram, Activity diagram and Sequence Diagram for design, Corresponding C++ code from design

List of Assignments:

1. Define Procedural Oriented Programming. Explain basic concepts of procedural oriented programming.
2. Differentiate between C and C++ in detail with suitable example.
3. Explain basic concepts of Object-Oriented Programming in detail with suitable example.
4. Write short note on:
5. Scope Resolution Operator ii)Access Specifiers
6. Explain Virtual Function and Function Overloading in detail with Example.
7. Explain Concepts of Object-Oriented Design and Modelling.

List of Laboratory Exercises:

1. Parameter passing: passing parameter by value vs by reference, passing array as constant pointer
2. Function overloading: writing string operations like strcat and strncat, strcpy and strncpy as overloaded functions.
3. Dynamically allocating space for a pointer depending on input and doing this repeatedly, depending on different inputs and finally de-allocating the pointer.
4. Define class complex with all possible operations: constructor, destructor, copy constructor, assignment operator with the data members stored as pointer to integers.
5. Define class vector of integers with all possible operations like constructor, destructor, copy constructor and assignment operators
6. Define class matrix of integers with all possible operations like constructor, destructor, copy constructor and assignment operators
7. Define class matrix of integers using vector, with all possible operations like constructor, destructor, copy constructor and assignment operators
8. Define class stack, queue, linked-list, array, set using some data-type (int) with data members kept as private and functions kept in both protected and public sections.
9. Define class complex with all possible operators: constructor, destructor, copy constructor, assignment operator and operators >, <, >=, <=, ==, ++ (pre and post), +, +=, ( ), with the data members stored as pointer to integers.
10. Define class vector of integers with all possible operations like constructor, destructor, copy constructor and assignment operators>, <, >=, <=, ==, ++ (pre and post), +, +=, ( )
11. Define class matrix of integers with all possible operations like constructor, destructor, copy constructor and assignment operators>, <, >=, <=, ==, ++ (pre and post), +, +=, ( ).
12. Define class matrix of integers using vector, with all possible operations like constructor, destructor, copy constructor and assignment operators>, <, >=, <=, ==, ++ (pre and post), +, +=, ( ).
13. Define stack and queue inherited from array class, with standard functions and operators
14. Define a class called ‘array’ with data type passed as template type with constructor, destructor, copy constructor and assignment operators and index operator.
15. Define template functions for compare and use it in the algorithms like bubble sort, insertion sort, merge sort.
16. Formatted input-output examples
17. Input manipulators

18. Overriding operators <<, >>
19. Define class model for complex number, student class, book class and show it using UML diagram as
a. well as concrete class.
20. Show behavioural modelling through sequence diagram and activity diagram for workflow in a typical log-in, log-out situation.

List of Project Based Learning Topics:

1. Employee Management System.
2. Trading Software.
3. Billing System.
4. Intuitive Gadgets.
5. Traffic Management System
6. Security Systems.
7. Car Rental System.
8. Login and Registration System.
9. Bookshop inventory system.
10. Student Report Management System.
11. Calendar application.

Text Books:

1. The C++ Programming Language, BjarneStroustrup.
2. C++ and Object-Oriented Programming Paradigm, Debasish Jana

Reference Books:

1. Programming – Principles and Practice Using C++, BjarneStroustrup.
2. The Design and Evolution of C++, BjarneStroustrup.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

SKILL LAB – III (DATA SCIENCE WITH PYTHON)

TEACHING SCHEME	EXAMINATION SCHEME		CREDITS ALLOTTED	
Theory :0 Hours / Week	**Termwork**	**:25Marks**	**Practical**	**:1 Credits**
Lab :2 Hours / Week	**Practical**	**:25 Marks**		
	Total	**: 50 Marks**	**Total**	**:1 Credits**

Course Objective:

To develop analytical skills among the students using data analysis methods and Python.

Prerequisite:

The students should have knowledge of basics of statistics.

Course Outcomes: On completion of the course, students will have the ability to:

1. Understand and Demonstrate fundamentals of Python programming.
2. Demonstrate visualization in Python using different Packages and Libraries.
3. Understand and Visualize various distributions using real time datasets.
4. Design models for simple and multiple linear regression models.
5. Develop a model for PCA to understand the effects of dimension reduction.
6. Design various clusters using real time datasets.

Unit I : 04 Hours

Python Basics: Python Programming Environment, Statements in Python, Data Structures, Expressions, Flow Controls, Functions, Numeric Types, Sequences and Class Definition, Constructors, Text & Binary Files- Reading and Writing.

Unit II : 04 Hours

Packages and Libraries used for Plotting in Visualization: Various Packages and Libraries used for plotting, Plotting Graphs, Controlling Graph, Adding Text, Different types of Graphs, Getting and Setting values in Graph.

Unit III: 04 Hours

Distributions: Type of data, Bernoulli Distribution, Uniform Distribution, Binomial Distribution, Normal Distribution, Poisson Distribution, Exponential Distribution. Visualization of Distribution.

Unit IV: 04 Hours

Linear Regression: Linear Regression Model, Model Assumptions, Validation of model.

Multiple Linear Regression: Multiple Linear Regression Model, Model Assumptions, Validation of model.

Unit V: 04 Hours

Principal Component Analysis: Principal Components, Algorithm for conducting Principal Component Analysis, Factors to decide retention of Principal Components.

Unit VI: **04 Hours**

Clustering: Introduction to Clustering, Types of Clustering, Correlations and Distance Measures, Hierarchical Clustering and Non-Hierarchical Clustering, Profiling and Interpreting Clusters.

Textbooks

1. Tim Hall and J-P Stacey "Python 3 for Absolute Beginners", A press.
2. Wes Mc Kinney. "Python for Data Analysis",O'Reilly Media, Inc.

Reference Books

1. Jake Vander Plas, "Python Data Science Handbook: Essential Tools for Working with Data"O'Reilly Media
2. David Spiegel halter "The Art of Statistics"Pelican

List of Laboratory Exercises

1. Introduction to python programming (String operation, Mathematical operation, loops, branching)
2. Write a program to perform basic operations using Python Functions.
3. Write a program to perform to read, write and modify text file data using OO Python.
4. Implement various pre-defined libraries in Python like Panda, NumPy, Cbor (Drawing of statistical graph)
5. Exercise different functionalities of Matplotlib package.
6. Write a program to measure central tendency and dispersion of given data.
7. Write a program to visualize different types of distributions.
8. Write a program to develop linear and multiple regression models using real time datasets
9. Implementation of clustering using real time datasets.

B. TECH (Computer Science & Business Systems)

SEMESTER – IV

COURSE SYLLABUS

OPERATING SYSTEM

TEACHING SCHEME		**EXAMINATION SCHEME**		**CREDITS ALLOTTED**	
Lectures	**:3Hrs./Week**	**Semester Examination**	**:60 Marks**	**Theory**	**:3 Credits**
Lab	**:2Hrs./Week**	**Internal Assessment**	**:40 Marks**		
		Term Work	**:25 Marks**	**Practical**	**:1 Credit**
		Oral	**:25 Marks**		
		Total	**:150 Marks**	**Total**	**:4 Credits**

Course Pre-Requisites:

Prerequisites for this course include thorough knowledge in some high-level programming language as C or C++ and UNIX / Linux operating system environment. As programs are to be implemented by writing C code during the course and will cover the details of C and its close relationship to UNIX and Linux in the case study in 6th unit.

Course Objectives:

1. To learn the basic concepts of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication.
3. To learn the methods of process scheduling.
4. To gain knowledge on Mutual exclusion, deadlock detection algorithms.
5. To know the concept of memory management and virtual memory.
6. To learn programmatically file management techniques.

Course Outcomes:

1. To learn and apply the basic concept of operating system.
2. To infer the concept of process and process state transition and concept of thread and multithreading.
3. Understand the importance of scheduling and types of scheduling algorithms.
4. To gain the knowledge of inter process communication strategies, concept of deadlock along with its avoidance.
5. To analyse the memory management techniques, paging, and segmentation.
6. To understand the file management and disk management techniques.

UNIT – I **6 Hours**

Introduction: Concept of Operating Systems (OS), Generations of OS, Types of OS, OS Services, Interrupt handling and System Calls, Basic architectural concepts of an OS, Concept of Virtual Machine, Resource Manager view, process view and hierarchical view of an OS.

UNIT – II **6 Hours**

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching.
Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.

UNIT – III **6 Hours**

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.
Scheduling Algorithms: Pre-emptive and non-pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF.

UNIT – IV **6 Hours**

Inter-process Communication: Concurrent processes, precedence graphs, Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Semaphores, Strict Alternation, Peterson’s Solution, The Producer / Consumer Problem, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader’s & Writer Problem, Dinning Philosopher Problem, Barber’s shop problem.
Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker’s algorithm, Deadlock detection and Recovery.
Concurrent Programming: Critical region, conditional critical region, monitors, concurrent languages, communicating sequential process (CSP); Deadlocks - prevention, avoidance, detection and recovery.

UNIT – V **6 Hours**

Memory Management: Basic concept, Logical and Physical address maps, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction.
Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page allocation, Partitioning, Paging, Page fault, Working Set, Segmentation, Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

UNIT – VI **6 Hours**

I/O Hardware: I/O devices, Device controllers, Direct Memory Access, Principles of I/O. File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed),Free-space management (bit vector, linked list, grouping), directory implementation(linear list, hash table), efficiency and performance.
Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.
Case Study: UNIX OS file system, shell, filters, shell programming, programming with the standard I/O, UNIX system calls.

List of Assignments:

1. To learn evolution and structure of operating system.
2. To understand the concept of Real Time scheduling.
3. To analyse the problem of process synchronization.
4. To implement the shell programming in UNIX OS.

List of Laboratory Exercises:

1. Unix commands (files directory, data manipulation, network communication etc), shell programming and vi editor
2. C program implementation of the following:
    a. Scheduling Algorithms
    b. Shared memory
    c. Thread and Multi Thread
    d. Inter Process Communication
    e. Deadlock Avoidance and Deadlock Detection
    f. Semaphore
    g. Memory Management
    h. Indexing and Hashing

List of Project Based Learning Topics:

1. Virtual traffic management system using threads with semaphore to control traffic.
2. Virtual memory management system.
3. File system handling.
4. A Client -Server application, use of IPC.
5. A simple web browser.
6. Device driver for some device.
7. Design of mail system project.
8. Design of RTOS for embedded system.
9. Mini project on Linux Shell.
10. Railway reservation system using scheduling.

Textbooks:

1. Operating System Concepts Essentials. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne.

Reference Books:

1. Operating Systems: Internals and Design Principles. William Stallings.
2. Operating System: A Design-oriented Approach. Charles Patrick Crowley.
3. Operating Systems: A Modern Perspective. Gary J. Nutt.
4. Design of the Unix Operating Systems. Maurice J. Bach.
5. Understanding the Linux Kernel, Daniel Pierre Bovet, *.Marco Cesati*

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT – III
Unit Test -2	UNIT – IV, UNIT – V, UNIT – VI

DATABASE MANAGEMENT SYSTEMS

TEACHING SCHEME	EXAMINATION SCHEME	CREDITS ALLOTTED
Lectures :3 Hrs./Week	**Semester Examination :60 marks**	**Theory :3 Credits**
Lab :2 Hrs./Week	**Internal Assessment :40 marks**	
	Term work :25 Marks	**Practical :1 credit**
	Practical :25 Marks	
	Total :150 Marks	**Total :4 Credits**

Course Prerequisites:

Students should have knowledge of

1. Basic understanding of data and data structure
2. Basic understanding of programming language

Course Objectives:

1. Identify various techniques to communicate with database.
2. Relate relevant data for effective processing of data.
3. Construct a database to maintain data adroitly.
4. Study various queries and tools to deal with the data.
5. Understand the relation between data set and respective means to access it.
6. Understand the influence of data in the effective development of software.

Course Outcomes:

After successful completion of this course students will be able to:

1. Model an application’s data requirements using conceptual modeling tools.
2. Demonstrate concepts of relational algebra and queries
3. Demonstrate concepts of relational database design
4. Interpret the query processing and optimization activities in database.
5. Interpret the transaction activities in database.
6. Recognize the emerging database applications and security concerns.

UNIT – I **6 Hours**

Introduction: Introduction to Database. Hierarchical, Network and Relational Models. Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).Data models: Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

UNIT – II **6 Hours**

Relational Query Languages: Relational algebra, Tuple and domain relational calculus,SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL,ORACLE, DB2, SQL server.

UNIT – III **6 Hours**

Relational Database Design: Domain and data dependency, Armstrong's axioms, Functional Dependencies, Normal forms, Dependency preservation, Lossless design.

UNIT – IV **6 Hours**

Query Processing and Optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

Storage strategies: Indices, B-trees, Hashing.

UNIT – V **6 Hours**

Transaction Processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp-based schedulers, multi-version and optimistic Concurrency Control schemes, Database recovery.

UNIT – VI **6 Hours**

Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.
Advanced topics: Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

List of Assignments:

Respective subject teacher shall design any six assignments on above units.

List of Laboratory Exercises:

Assignments & tutorials covering the relational database design and operations in SQL and PL/SQL

List of Project Based Learning Topics:

1. Make a project to maintain employee data using files and dynamic object/structure. The project should be able to read, write, modify, add and search records. Also demonstrate the effect of performing change in employer data definition after few records have been added.
2. Make an extended ER diagram for insurance management system. Transform this into relation design and implement these relations with appropriate domain and integrity constraints.
3. Employ various data control restrictions on databases, relations and attributes of relations.
4. Create a phonebook which enables user to save contacts with additional information and provides various retrieval mechanisms. Provisions should be made to view data in multiple ways.
5. Design and develop a library management system. The relations in the system should be normalized up to BCNF
7. Design and develop a inventory management system and create multiple views on the relations so that users not authorised to edit the relations should be able to views the data.
8. Implement of audit trails and backup on relations.
9. Create a student result calculation system. However when updating final results after calculation should be only of students who paid complete fees, such that transaction of each row is executed separately. Hint- use explicit cursor
10. Develop a student data management system using hash files.
11. Installation of a NoSQL database and implementing a simple student database to compare with SQL database.

Textbooks:

1. Database System Concepts. Abraham Silberschatz, Henry F. Korth and S. Sudarshan.

Reference Books:

1. Principles of Database and Knowledge – Base Systems, Vol 1 by J. D. Ullman.
2. Fundamentals of Database Systems. R. Elmasri and S. Navathe.
3. Foundations of Databases. Serge Abiteboul, Richard Hull, VictorVianu.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT – III
Unit Test -2	UNIT – IV, UNIT – V, UNIT – VI

INTRODUCTION TO INNOVATION, IP MANAGEMENT & ENTREPRENEURSHIP

TEACHING SCHEME	**EXAMINATION SCHEME**	**CREDITS ALLOTTED**
Lectures : 2 Hrs./Week	**Semester Examination :60 marks**	**Theory :2 Credits**
	Internal Assessment :40 marks	
	Total :100 Marks	**Total :2 Credits**

Course Pre-Requisites:

Good knowledge of Fundamentals of Management.

Course Objective:

The major emphasis of the course will be on creating a learning system through which management students can enhance their innovation and creative thinking skills, acquaint themselves with the special challenges of starting new ventures and use IPR as an effective tool to protect their innovations and intangible assets from exploitation.

Course Outcomes:

1. Learn to be familiar with creative and innovative thinking styles.
2. Learn opportunity reorganization and entrepreneurship skills.
3. Learn to investigate, understand and internalize the process of founding a startup.
4. Understand financial aspects of Entrepreneurship.
5. Learn to manage various types of IPR to protect competitive advantage.
6. Understand the types of IP.

UNIT – I **4 Hours**

Innovation: What and Why?
Innovation as a core business process, Sources of innovation, Knowledge push vs. need pull innovations.
Class Discussion- Is innovation manageable or just a random gambling activity?

UNIT – II **4 Hours**

Building an Innovative Organization
Creating new products and services, exploiting open innovation and collaboration, use of innovation for starting a new venture
Class Discussion- Innovation: Co-operating across networks vs. 'go-it-alone' approach.

UNIT – III **4 Hours**

Entrepreneurship:
Opportunity recognition and entry strategies, Entrepreneurship as a Style of Management, Maintaining Competitive Advantage- Use of IPR to protect Innovation.

UNIT – IV **4 Hours**

Entrepreneurship- Financial Planning: Financial Projections and Valuation. Stages of financing, Debt, Venture Capital, and other forms of Financing

UNIT – V **4 Hours**

Intellectual Property Rights (IPR): Introduction and the economics behind development of IPR: Business Perspective, IPR in India – Genesis and Development, International Context, Concept of IP Management, Use in marketing.

UNIT – VI **4 Hours**

Types of Intellectual Property: Patent- Procedure, Licensing and Assignment, Infringement and Penalty, Trademark- Use in marketing, example of trademarks- Domain name, Geographical Indications- What is GI, Why protect them? Copyright- What is copyright, Industrial Designs- What is design? How to protect?
Class Discussion- Major Court battles regarding violation of patents between corporate companies.

List of Assignments:

1. Case study materials book will be given to students. Students are required to meet in groups before coming to class and prepare on the case for the day. Instructor may ask the student groups to present their analysis and findings to the class.
2. Further, the topic for class discussion will be mentioned beforehand and students should be ready to discuss these topics (in groups) in class. Students are required to meet in groups before coming to class and prepare on the topic. Few topics are mentioned below as examples. Instructor can add or change any topic as per requirement.
3. Topic 1- Is innovation manageable or just a random gambling activity?
4. Topic 2- Innovation: Co-operating across networks vs. 'go-it-alone' approach.
5. Topic 3- Major Court battles regarding violation of patents between corporate companies.

List of Project Based Learning Topics:

Design case studies for based on any of the following technologies.

1. Artificial intelligence
2. Machine Learning
3. Cloud Computing
4. IOT
5. HCI
6. Brain Computer Interface
7. Web Designing
8. Blockchain

Textbooks:

1. Joe Tidd, John Bessant. Managing Innovation: Integrating Technological, Market and Organizational Change
2. Case Study Materials: To be distributed for class discussion

Syllabus for Unit Test:	**Unit**
Unit Test -1	UNIT – I, UNIT – II, UNIT – III
Unit Test -2	UNIT – IV, UNIT – V, UNIT – VI

DESIGN THINKING

TEACHING SCHEME	**EXAMINATION SCHEME**	**CREDITS ALLOTTED**
Lectures :3 Hrs./Week	**Semester Examination :60 marks**	**Theory :3 Credits**
Lab :2 Hrs./Week	**Internal Assessment :40 marks**	**Practical/Oral :1 Credit**
	Term Work :25 Marks	
	Total :125 Marks	**Total :4 Credits**

Course Pre-Requisites:

Students should be well versed Completion of all units from Semesters 1, 2, 3 and 4

Course Objective:

1. Recognize the importance of DT
2. Explain the phases in the DT process
3. List the steps required to complete each phase in DT process
4. Apply each phase in the DT process
5. Use doodling and storytelling in presenting ideas and prototypes
6. Create value proposition statements as part of their presentations
7. Recognize how DT can help in functional work
8. Recognize how Agile and DT complement each other to deliver customersatisfaction

Course Outcomes:

After successful completion of this course students will be able to:

1. Understand and Implement the Phases in the DT process
2. Analyze the steps required to conduct an immersion activity
3. Evaluate the personas to create problem statements in the define phase of DT
4. Apply the steps in the ideate phase of DT
5. Implement a prototype to create a value proposition statement
6. Develop and Test a prototype through a DT process

UNIT – I **6 Hours**

Introduction: Recognize the importance of Design Thinking why is Design Thinkingimportant for business? Why is Design Thinking important for you? , Identify the steps in the DT process What is DT? Empathize (search for rich stories and find some love), Define (user need and insights – their POV), Ideate (ideas, ideas, ideas), Prototype (build to learn), Test (show, don’t tell)

UNIT – II **6 Hours**

Empathy Phase: Recognize the steps in the empathize phase of DT, What is empathy? Ask What? How? Why?, Different types to developing Empathy towards People Identify the steps required to conduct an immersion activity, How to empathize?, Intro to Immersion Activity, Conduct an immersion activity and fill up the DT question template, Immersion activity

UNIT – III **6 Hours**

Define Phase: Creating personas: Recognize the steps to create personas in the definephase of DT, What is a persona and how do I create one? Four Different Perspectives on Personas 1)Goal-directed Personas 2)Role-Based Personas 3) Engaging Personas 4) Fictional Personas, 10 steps to Creating Your Engaging Personas and Scenarios Recognize the steps to create problem statements in the define phase of DT, Problem statements, Defining problem statements, Define the problem statements in the define phase of DT

UNIT – IV **6 Hours**

Ideate Phase: How to Ideate?, Recognize the steps in the ideate phase of DT, Apply the steps in the ideate phase of DT, Ideation games: Game 1: Six Thinking Hats, Game 2: Million-dollar idea, Ideate to find solutions, Characteristics Required for Successful Ideation, Recognize how doodling can help to express ideas, Recognize the importance storytelling in presenting ideas and prototypes, What is Storytelling in DT?

UNIT – V **6 Hours**

Prototype Phase: Recognize the importance of the prototype phase in DT, Prototype your idea, Create a prototype: Types of Prototyping 1)Low-Fidelity Prototyping 2) High-Fidelity Prototyping , Guidelines for Prototyping, Recognize the importance of service value proposition, Create a value proposition statement

UNIT – VI **6 Hours**

Testing Phase: Testing in Design Thinking, Test the Prototype, Role of DT in your work, discuss How DT can help me to become a better coder?, Agile and DT complement each other to deliver customer satisfaction, Share your Satori.

Text Books:

Design Thinking by Mr.Sambhrant Srivastava, Mr.Vijay Kumar, Dr.Pramod Kumar Srivastava.

Reference Books:

Hooked by Nir Eyal. The Art of Creative Thinking by Rod JudkinsStart-Up nation by Dan Senor and Saul singerStart with Why by Simon Sinek

List of Laboratory Exercises:

1. Foundations of DT, Course outline, Group roles
2. Intro and Problem Definition of project (phase 1 – Hear)
3. Idea Generation of project (phase 2 - Create)
4. Concept Development and Design project (phase 3 - Create)
5. Project Teams meet to discuss and continue to prepare their work
6. Testing and Project Documentation on project (phase 4 - Deliver)
7. Final group project presentation

List of Project-Based Learning:

1. Case study on Airbnb
2. Case study on Pill pack
3. Case study on Clean Team
4. Case study on IBM
5. Case study on Stanford hospital
6. Case study on Uber eats
7. Case study on Bank of America

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

OPERATION RESEARCH

TEACHING SCHEME	**EXAMINATION SCHEME**	**CREDITS ALLOTTED**
Lectures :3 Hrs./Week	**Semester Examination :60 marks**	**Theory :3 Credits**
Lab :2 Hrs./Week	**Internal Assessment :40 marks**	**Practical :1 Credit**
	Term Work :25 Marks	
	Total :125 Marks	**Total :4 Credits**

Course Pre-Requisites:

Good knowledge of mathematics.

Course Objective:

The students will be able to understand various models in operations research used in industries to solve problems.

Course Outcomes:

As a part of this course, students will:

1. Understand OR problem and associated models.
2. Understand Linear Algebra.
3. Use transportation and assignment problems.
4. Use PERT for modeling.
5. Use Inventory Control System.
6. Apply queuing theory and modulation techniques.

UNIT – I **6 Hours**

Introduction to OR:
Origin of OR and its definition. Concept of optimizing performance measure, Types of OR problems, Deterministic vs. Stochastic optimization, Phases of OR problem approach – problem formulation, building mathematical model, deriving solutions, validating model, controlling and implementing solution.

UNIT – II **6 Hours**

Linear Programming:
Linear programming – Examples from industrial cases, formulation & definitions, Matrix form. Implicit assumptions of LPP. Some basic concepts and results of linear algebra – Vectors, Matrices, Linear Independence/Dependence of vectors, Rank, Basis, System of linear eqns., Hyperplane, Convex set, Convex polyhedron, Extreme points, Basic feasible solutions.
Geometric method: 2-variable case, Special cases – infeasibility, unboundedness, redundancy &degeneracy, Sensitivity analysis.
Simplex Algorithm – slack, surplus & artificial variables, computational details, big-M method, identification and resolution of special cases through simplex iterations.
Duality – formulation, results, fundamental theorem of duality, dual-simplex and primal-dual algorithms.

UNIT – III **6 Hours**

Transportation and Assignment Problems:
TP - Examples, Definitions – decision variables, supply & demand constraints, formulation, Balanced & unbalanced situations, Solution methods – NWCR, minimum cost and VAM, test for optimality (MODI method), degeneracy and its resolution.
AP - Examples, Definitions – decision variables, constraints, formulation, Balanced &unbalanced situations, Solution method – Hungarian, test for optimality (MODI method), degeneracy & its resolution.

UNIT – IV **6 Hours**

PERT – CPM:
Project definition, Project scheduling techniques – Gantt chart, PERT & CPM, Determination of critical paths, Estimation of Project time and its variance in PERT using statistical principles, Concept of project crashing/time-cost trade-off.

UNIT – V **6 Hours**

Inventory Control:
Functions of inventory and its disadvantages, ABC analysis, Concept of inventory costs, Basics of inventory policy (order, lead time, types), Fixed order-quantity models – EOQ, POQ & Quantity discount models. EOQ models for discrete units, sensitivity analysis and Robustness, Special cases of EOQ models for safety stock with known/unknown stock out situations, models under prescribed policy, Probabilistic situations.

UNIT – VI **6 Hours**

Queuing Theory:
Definitions – queue (waiting line), waiting costs, characteristics (arrival, queue, service discipline) of queuing system, queue types (channel vs. phase). Kendall's notation, Little's law, steady state behavior, Poisson's Process & queue, Models with examples - M/M/1 and its performance measures; M/M/m and its performance measures; brief description about some special models.

Simulation Methodology:
Definition and steps of simulation, random number, random number generator, Discrete Event System Simulation – clock, event list, Application in Scheduling, Queuing systems and Inventory systems.

List of Assignments:

Respective subject teacher shall design any six assignments on above units.

List of Laboratory Exercises:

1. Formulation of linear programming problems.
2. Solution of linear programming problem using graphical method with:
   i. Multiple constraints
   ii. Unbounded solution
   iii. Infeasible solution
   iv. Alternative or multiple solution
3. Enumeration of all basic solutions for linear programming problem.
4. Solution of linear programming problem with simplex method.
5. Problem solving using Big M method.
6. Problem solving using two phase method.
7. Solution on primal problem as well as dual problem.
8. Solution based on dual simplex method.
9. Verification of weak duality, strong duality and complementary slackness property.
10. Solution of transportation problem.
11. Solution of assignment problem.
12. Solution of integer programming problem using Branch and Bound method.
13. Solution of integer programming problem using Gomory's cutting plane method.
14. Simulation: Random number generation.
15. Monte Carlo method.
16. Performance measures for M/M/1 queuing model.
17. ABC analysis.
18. Inventory model.

List of Project Based Learning Topics:

1. Students must work on one of the projects listed below (but not limited to) during the semester
2. Find the companies that used OR as a tool to sort a problem successfully and unsuccessfully. Compare them and analyze as to why certain strategies worked and others failed.
3. Visit any industry and choose one of their products. Develop a LPP for maximizing profits on the sale of that product considering the various constraints on it. Solve the LPP and make suggestions of the same for the company.
4. Develop a software that helps in making timetable for the department by making and solving an LPP.
5. Visit a small departmental store/hotel, collect data, and make an LPP for optimum use of space. Solve the LPP and make relevant suggestions
6. Write a research paper on how LPP helps companies to solve problems referencing latest papers.
7. Write a research paper on how assignment tools help companies to solve problems referencing latest papers.
8. Write a research paper on how transportation tools help companies to solve problems referencing latest papers.
9. Visit a small-scale industry. Collect data and make WBS and a network diagram. Solve it by CPS and PERT methods ad make relevant suggestions
10. Write a research paper on how network analysis tools help companies to solve problems referencing latest papers.
11. Write a research paper on how queuing models help companies to solve problems referencing latest papers.
12. Go to a nearby petrol pump, bank, departmental store, hotel. Record the arrival and service rates for multiple day. Analyze the data and make relevant suggestions
13. Write a research paper on how inventory models help companies to solve problems referencing latest papers.
14. Go to a nearby petrol pump, departmental store, hotel. Record inventory levels and inventory practices for multiple day. Analyze the data and make relevant suggestions.

Textbooks:

1. Operations Research: An Introduction. H.A. Taha.

Reference Books:

1. Linear Programming. K.G. Murthy.
2. Linear Programming. G. Hadley.
3. Principles of OR with Application to Managerial Decisions. H.M. Wagner.
4. Introduction to Operations Research. F.S. Hiller and G.J. Lieberman.
5. Elements of Queuing Theory. Thomas L. Saaty.
6. Operations Research and Management Science, Handbook: Edited By A. Ravi Ravindran.
7. Management Guide to PERT/CPM. Wiest& Levy.
8. Modern Inventory Management. J.W. Prichard and R.H. Eagle.

Syllabus for Unit Test:	**Unit**
Unit Test -1	UNIT – I, UNIT – II, UNIT - III
Unit Test -2	UNIT – IV, UNIT – V, UNIT - VI

SKILL LAB – IV (SOFTWARE DESIGN WITH UML)

TEACHING SCHEME	**EXAMINATION SCHEME**		**CREDITS ALLOTTED**
Lectures :0 Hrs./Week	**Term work**	**: 25 Marks**	**Theory :0 Credits**
Lab :2 Hrs./Week	**Practical**	**: 25 Mark**	**Practical :1 Credit**
Tutorial: 1 Hrs./Week			**Tutorial :1 Credit**
	Total	**:50 Marks**	**Total :2 Credits**

Course Pre-Requisites:

The students should have sound knowledge software engineering and programming experience using data structures.

Course Objective:

To model software solutions, application structures, system behaviour and business processes using UML.

Course Outcomes:

1. Apply Unified Modelling Language (UML) for representation of an object-oriented system using different modelling views.
2. Analyse requirements to represent logical design that is recognized by various object relationships.
3. Identify interaction among structural elements to translate analysis model into design model.
4. Model dependencies among packages and package able element ownership.
5. Model dynamic behavior of the system and message flow from one object to other.
6. Envision the topology of the physical components of a system where the software components are utilized.

UNIT – I **4 Hours**

Introduction to on Object Oriented Technologies and the UML Method: Software development process: The Waterfall Model vs. The Spiral Model; The Software Crisis, description of the real world using the Objects Model; Classes, inheritance and multiple configurations; Quality software characteristics; Description of the Object-Oriented Analysis process vs. the Structure Analysis Model. UML Language: Standards; Elements of the language; General description of various models; The process of Object-Oriented software development; Description of Design Patterns; Technological Description of Distributed Systems.

UNIT – II **4 Hours**

Requirements Analysis Using Case Modeling AND The Logical View Design: Analysis of system requirements; Actor definitions; Writing a case goal; Use Case Diagrams; Use Case Relationships. **The Static** Structure Diagrams: The Class Diagram Model; Attributes descriptions; Operations descriptions; Connections descriptions in the Static Model; Association, Generalization, Aggregation, Dependency, Interfacing, Multiplicity.

UNIT – III **4 Hours**

Transfer from Analysis to Design in the Characterization Stage: Interaction Diagrams: Description of goal; Defining UML Method, Operation, Object Interface, Class; Sequence Diagram; Finding objects from Flow of Events; Describing the process of finding objects using a Sequence Diagram; Describing the process of finding objects using a Collaboration Diagram.

UNIT – IV **4 Hours**

Package Diagram Model: Description of the model; White box, black box; Connections between packagers; Interfaces; Create Package Diagram; Drill Down.

UNIT – V **4 Hours**

Dynamic Model: State Diagram / Activity Diagram: Description of the State Diagram; Events Handling; Description of the Activity Diagram; Exercise in State Machines.

UNIT – VI **4 Hours**

Component Diagram Model: Physical Aspect; Logical Aspect; Connections and Dependencies; User face; Initial DB design in a UML environment. Deployment Model: Processors; Connections; Components; Tasks; Threads; Signals and Events.

List of Assignments:

Teaching faculty will take assignment on following topic for internal assessment.

1. Study of UML notations
2. Class diagram
3. Interaction diagrams
4. Activity diagram
5. State diagram
6. Software project covering various software development methodology techniques will be implemented.

List of Laboratory Exercises:

1. For Object Oriented Modelling, choose a hypothetical system of significant complexity (on your project topic) and write an SRS.
2. Draw one or more Use Case diagrams for capturing and representing requirements of the system. Use case diagrams must include various scenarios as per template.
3. Draw basic class diagrams to identify and describe key concepts like classes, types in your system and their relationships.
4. Draw sequence diagrams with advanced notation for your system to show objects and their message exchanges.
5. Draw activity diagrams to display either business flows or activity flow.
6. Draw component diagrams assuming that you will build your system reusing existing components along with a few new ones.
7. Draw deployment diagrams to model the runtime architecture of your system.
8. Implement Singleton Pattern, Abstract Factory Pattern and Singleton Pattern using Java.

List of Project Based Learning Topics:

1. Implementation level UML class diagram to illustrate usage of Android Camera API
   Deployment diagram for Android application deployment.
2. Online shopping UML diagrams
3. Ticket vending machine UML diagrams
4. Bank ATM UML diagrams.
5. Hospital management UML diagrams
6. Airport check-in and security screening Use case modelling and Requirement analysis
7. e-Library online public access UML
8. Coffee vending machine UML diagrams.
9. Online order Processing UML diagrams.

Textbooks:

1. Object-Oriented Software Engineering: using UML, Patterns, and Java. Bernd Bruegge and Allen H. Dutoit.
2. The Unified Modelling Language User Guide. Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

Reference Books:

1. Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlassises.

Syllabus for Unit Test:

Unit Test -1	UNIT – I, UNIT – II, UNIT – III
Unit Test -2	UNIT – IV, UNIT – V, UNIT – VI