Bharati Vidyapeeth

(Deemed to be University)

Pune, India

Faculty of Engineering and Technology

Programme: B.Tech (Computer Science and Business Systems)

COURSE STRUCTURE AND SYLLABUS

(Choice Based Credit System)

B.Tech (Computer Science and Business Systems)

2018 Course

Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune
B.Tech- Computer Science & Business Systems (Semester- III and IV)
Revised New Syllabus

Semester- III		Teaching Scheme				Examination Scheme-Marks							Credit		
						End Semester Examination	Continuous Assessment			TW & Practical	TW & Oral	Total	Theory	Term Work	Total
ID	Course	Lecture	Tutorial	Practical	Contact Hours per week		Unit Test	Attendance	Assignments						
2.1	Formal Language and Automata Theory	4	1	2	6	60	20	10	10	-	50	150	5	1	6
2.2	Computer Organization & Architecture	4	0	2	5	60	20	10	10	-	50	150	4	1	5
2.3	Object Oriented Programming	3	0	2	5	60	20	10	10	50	-	150	3	1	4
2.4	Computational Statistics	3	0	2	5	60	20	10	10	50	-	150	3	1	4
2.5	Software Engineering	4	1	2	6	60	20	10	10	50	-	150	5	1	6
2.6	Indian Constitution (Non Credit)												0	0	0
	Total	18	2	10	27	300	100	50	50	150	100	750	20	5	25

Semester- IV		Teaching Scheme				Examination Scheme-Marks							Credit		
						End Semester Examination	Continuous Assessment			TW & Practical	TW & Oral	Total	Theory	Term Work	Total
ID	Course	Lecture	Tutorial	Practical	Contact Hours per week		Unit Test	Attendance	Assignments						
2.8	Operating Systems	4	0	2	5	60	20	10	10	50	-	150	4	1	5
2.9	Database Management Systems	4	0	2	5	60	20	10	10	50	-	150	4	1	5
2.10	Software Design with UML	4	0	2	5	60	20	10	10	50	-	150	4	1	5
2.11	Introduction to Innovation, IP Management & Entrepreneurship	4	0	0	3	60	20	10	10	-	-	100	3	0	3
2.12	Business Communication & Value Science – III	2	0	4	6	50	0	0	0	-	50	100	2	2	4
2.13	Operations Research	2	0	2	4	60	20	10	10	-	50	150	2	1	3
2.14	Essence of Indian Traditional Knowledge(Non Credit)									-	-	0			
	Total	20	0	12	28	350	100	50	50	150	100	800	19	6	25

BHARATI VIDYAPEETH (Deemed to be University)
COLLEGE OF ENGINEERING, PUNE-43
B. Tech. (Computer Science & Business Systems)

Vision of the Department

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

Mission of the Department

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

Programme Educational Objectives

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

Programme Outcomes

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.

FORMAL LANGUAGE & AUTOMATA THEORY

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Lectures: 3Hrs./Week	Semester Examination: 60 marks	Theory: 4 Credits
Tutorials: 1Hr./Week	Continuous Assessment: 40 marks	
Lab: Nil	Term Work: Nil	

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) To construct finite state machines to solve problems in computing.
2) To write mathematical expressions for the formal languages.
3) To understand context free and context sensitive languages.
4) To construct Turing Machine for formal languages.
5) To express the understanding of the decidability and undecidability problems.
6) To understand NP Hard and complete problems.

Topics to Be Covered:

UNIT – I **[6 Hours]**

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

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, *Kleene'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, 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.

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]**

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

Reference Books:

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 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

Syllabus of Semester III

Computer Organization & Architecture

TEACHING SCHEME:	**EXAMINATION SCHEME:**	**CREDITS ALLOTTED:**
Theory: 04 Hours / Week	End Semester Examination: 60 Marks	**Theory: 4 Credits**
Practical: 02 Hours / Week	Continuous Assessment: 40 Marks	
	Term Work and Oral: 50 Marks	**TW and Oral :1 Credit**

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) Explain the architecture and functions of Central Processing Unit.
2) Solve fixed point and floating-point arithmetic problems using algorithms
3) List the design approaches and functional requirements for implementing control unit.
4) Analyze the characteristics of memory system.
5) Describe the I/O organization and interconnections.
6) Infer parallel processing and multiprocessor configuration.

Topics to Be Covered:

UNIT – I **[8 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 [8 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 [8 Hours]

CPU control unit design: Hardwired and micro-programmed design approaches, design of a simple hypothetical CPU.

UNIT – IV [8 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 [4 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 [4 Hours]

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

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

Home 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

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

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

B.Tech (Computer Science & Business Systems)

Semester – III

OBJECT ORIENTED PROGRAMMING

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Lectures: 3Hrs./Week	Semester Examination: 60 marks	Theory: 3 Credits
Practical: 2 Hrs/Week	Continuous Assessment: 40 marks	
	Term Work and Practical: 50 marks	Term Work and Practical: 1 Credit

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 able to:

1) Understand basic concepts of Procedural programming and, the overview of C programming language
2) Understand some basic difference 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.

Topics to Be Covered:

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

Text Books:

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

Reference Books:

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

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

B.Tech in Computer Science & Business Systems

Final Syllabus, Semester – III

Computational Statistics

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Theory: 03 Hours / Week	End Semester Examination: 60 Marks	**Theory: 3 Credits**
Practical: 04 Hours / Week	Continuous Assessment: 40 Marks	**TW and Practical :1 Credit**
	Term Work and Practical: 50 Marks	

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

Course Objectives:

The aim of this course is to give graduate students a solid foundation of computational statistics, basics of analysis and Python programming. The course objective is to exercise students for data set handling, data wrangling, data visualization etc. using Python.

Course Outcomes:

- Understand basics of normal distribution and linear regression model.
- Apply knowledge of multivariate regression and discriminant analysis.
- Outline component analysis and factor analysis.
- Design various clusters
- Understand and demonstrate fundaments of Python programming.
- Demonstrate visualization in Python

Topics to Be Covered:

UNIT – I [8 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 – II [5 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 III [7 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 – IV **[5 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.

UNIT – V **[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 – VI **[6 Hours]**

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

Visualization in Python: Matplotlib package, Plotting Graphs, Controlling Graph, Adding Text, More Graph Types, Getting and setting values, Patches.

Term Work

1. Introduction to python programming (String operation, Mathematical operation, loops, branching).
2. Implementation of classes and constructor in Python.
3. Implementation of basic data structures in Python.
4. File Handling in the Python.
5. Introduction to data set handling in Python.
6. Implement various pre-defined libraries in Python like Panda, NumPy, Cbor (Drawing of statistical graph).
7. Implementation Multivariate Normal Distribution.
8. Implementation Multiple Linear Regression Model
9. Implementation Multivariate Regression
10. Implementation Discriminant Analysis
11. Implementation clustering and segmentation
12. Implementation of data wrangling, data aggregation, group operations and time series operations.
13. Data Visualization in Python.

Text Books:

1. *An Introduction to Multivariate Statistical Analysis,* T.W. Anderson.
2. *Applied Multivariate Data Analysis, Vol I & II,* J.D. Jobson.
3. *Beginning Python: From Novice to Professional,* Magnus Lie Hetland. *Edition, 2005.*

Reference Books:

1. *The Foundations of Factor Analysis*, A.S. Mulaik.
2. *Introduction to Linear Regression Analysis*, D.C. Montgomery and E.A. Peck.
3. *Python for Data Analysis,* Wes Mc Kinney.
4. *Programming Python,* Mark Lutz.
5. *Python 3 for Absolute Beginners,* Tim Hall and J-P Stacey.

SOFTWARE ENGINEERING

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Lectures: 4Hrs./Week	Semester Examination: 60 marks	Theory: 5 Credits
Tutorials: 1Hr./Week	Continuous Assessment: 40 marks	Term Work and Practical credit: 1
Lab: 2Hrs./Week	Term Work and Practical: 50	

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

Topics to Be Covered:

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.

Home Assignments:

1. UML diagrams
2. Data Flow Diagrams
3. Testing
4. Software project covering various software development methodology techniques will be implemented.

Text Books:

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

Reference Books:

3. *Fundamentals of Software Engineering,* Carlo Ghezzi, Jazayeri Mehdi, Mandrioli Dino

4. *Software Requirements and Specification: A Lexicon of Practice, Principles and Prejudices*, MichaelJackson
5. *The Unified Development Process*, Ivar Jacobson, Grady Booch, James Rumbaugh
6. *Design Patterns: Elements of Object-Oriented Reusable Software*, Erich Gamma, Richard Helm,Ralph Johnson, John Vlissides
7. *Software Metrics: A Rigorous and Practical Approach*, Norman E Fenton, Shari Lawrence Pfleeger

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

OPERATING SYSTEM

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Lectures: 4Hrs./Week	Semester Examination: 60 marks	Theory: 4 Credits
Tutorials: 0Hr./Week	Continuous Assessment: 40 marks	Term Work and Practical credit: 1
Lab: 2Hrs./Week	Term Work and Practical: 50	

Course Pre Requisites:

Prerequisites for this course include thorough knowledge in some high-level programming language of C or C++ and UNIX and Linux 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 6^{th} unit.

Course Objective:

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 algorithms, deadlock detection algorithms and agreement protocols

5. To know the management aspects of memory management and virtual memory

6. To learn programmatically file management techniques

Course Outcomes:

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

Topics to Be Covered:

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.

Home Assignments:

1. To implement scheduling algorithms
2. To implement resource allocation graph
3. To implement Banker's Algorithm
4. To implement the shell programming in UNIX OS

Text Books:

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: 4 Hrs./Week	Semester Examination: 60 marks	Theory: 4 Credits
Tutorials: NIL	Continuous Assessment: 40 marks	Term Work and Practical : 1 credit
Lab: 2 Hrs./Week	Term Work and Practical: 50	

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 influence of data in the effective development of software.

Course Prerequisites:
Students should have knowledge of
1) Basic understanding of data and data structure
2) Basic understanding of programming language

Course Outcomes:

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

Topics to Be Covered:

UNIT – I [6 Hours]

Introduction: Introduction to Database. Hierarchical, Network and Relational Models.

Database system architecture: Data Abstraction, Data Independence, Data DefinitionLanguage (DDL), Data Manipulation Language (DML).

Data models: Entity-relationship model, network model, relational and object orienteddata 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, Queryequivalence, 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, Webdatabases, Distributed databases, Data warehousing and data mining.

Assignments:

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

Text Books:

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

Software Design with UML

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Lectures: 4Hrs./Week	Semester Examination: 60 marks	Theory: 4 Credits
Lab: 2Hrs./Week	Continuous Assessment: 40 marks	Term Work and Practical credit: 1
	Term Work and Practical: 50	

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 behavior and business processes using .

Course Outcomes:

7. Apply Unified Modeling Language (UML) for representation of an object-oriented system using different modeling views
8. Analyze requirements to represent logical design that is recognized by various object relationships.
9. Identify interaction among structural elements to translate analysis model into design model.
10. Model dependencies among packages and package able element ownership
11. Model dynamic behavior of the system and message flow from one object to other.
12. Envision the topology of the physical components of a system where the software components are utilized

Topics to Be Covered:

UNIT – I [6 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 [6 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 [6 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 [6 Hours]

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

UNIT – V [6 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 [6 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.

Home Assignments:

5. Study of UML notations
6. Class diagram
7. Interaction diagrams
8. Activity diagram
9. State diagram
10. Software project covering various software development methodology techniques will be implemented.

Text Books:

1. Object-Oriented Software Engineering: using UML, Patterns, and Java. Bernd Bruegge and Allen H. Dutoit.

Reference Books:

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

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: 3Hrs./Week	Semester Examination: 60 marks	Theory: 5 Credits
Tutorials: 0Hr./Week	Continuous Assessment: 40 marks	Term Work and Practical credit: 1
Lab: 0Hrs./Week	Term Work and Practical: 00	

Course Pre Requisites:

Good knowledge of Fundamentals of Management (Covered in Year 2, Semester 1)

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:

As a part of this course, students will:

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

Topics to Be Covered:

UNIT – I [6 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 [6 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 [6 Hours]

Entrepreneurship:

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

UNIT – IV [6 Hours]

Entrepreneurship- Financial Planning:

- Financial Projections and Valuation
- Stages of financing
- Debt, Venture Capital and other forms of Financing

UNIT – V [6 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 [6 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.

Home Assignments:

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.

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.

Topic 1- Is innovation manageable or just a random gambling activity?

Topic 2- Innovation: Co-operating across networks vs. 'go-it-alone' approach.

Topic 3- Major Court battles regarding violation of patents between corporate companies.

Text Books:

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

Business Communication and Value Science-III

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Lectures: 2Hrs./Week	Semester Examination: 50 marks	Theory: Credits 02
Tutorials: 0 Hr./Week	Continuous Assessment: No	Term Work Credit: 02
Lab: 4 Hrs./Week	Term Work and Oral: 50	

Course Pre Requisites:

Good knowledge of Business Communication and Value Science (Covered Semester 1 and 2) Basic Knowledge of English (verbal and written) Completion of all units from Semesters 1 and 2

Course Objective:

Develop technical writing skills; introduce students to Self-analysis techniques like SWOT & TOWS and develop the sense of Pluralism & cultural spaces, Cross-cultural communication, Science of Nation building**.**

Course Outcomes:

Upon completion of the course, students shall have ability to

Apply & analyze the basic principles of SWOT & life positions.

Understand, analyze & leverage the power of motivation in real life.

Identify & respect pluralism in cultural spaces.

Understand and apply the concepts of Global, glocal and translocational

Analyze cross cultural communication

Apply the science of Nation building, the diverse culture of India

Identify the common mistakes made in cross-cultural communication, tools of technical writing, recognize the roles and relations of different genders.

Understand Artificial intelligence &recognize its impact in daily life

Topics to Be Covered:

UNIT – I **[6 Hours]**

SWOT and Life Positions:

Summarize the basic principles of SWOT and Life Positions; apply SWOT in real life scenarios. TOWS analysis, research on TOWS and find out how you can turn your threat into opportunity

UNIT – II **[6 Hours]**

SWOT and TOWS:

Research through SWOT and TOWS on what are the strengths they have identified to survive in the VUCA World, Motivation: its role and application in real life.

UNIT – III **[6 Hours]**

Pluralism in cultural spaces:

Identify pluralism in cultural spaces, Respect pluralism in cultural spaces, Differentiate between the different cultures of India,

UNIT – IV **[6 Hours]**

Cross cultural communication

Define the terms global, glocal and translocational, Differentiate between global, glocal and translocational culture, implications of cross-cultural communication, common mistakes made in cross-cultural communication, roles and relations of different genders

UNIT – V **[6 Hours]**

Nation Building:

Role of science in nation building, tools and best practices of technical writing, technical writing in real-life scenarios

UNIT – VI **[6 Hours]**

Roles of technical writing in science and technology:

AI (artificial intelligence), the importance of AI, Designing College in the year 2090 with help of technical writing and technology, role of technical writing in science and technology, IOT

Text Books:

1. **Swot Analysis: A Guide to Swot for Business Studies Students by Alan Sarsby**
2. **The SWOT Analysis: Using Your Strength to Overcome Weaknesses, Using Opportunities to Overcome Threats by Lawrence G. Fine**
3. **Cross-Cultural and Intercultural Communication by William B. Gudykunst**

Operations Research

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Lectures: 2Hrs./Week	Semester Examination: 60 marks	Theory: 2 Credits
Tutorials: 0Hr./Week	Continuous Assessment: 40 marks	Term Work and Practical credit: 1
Lab: 2Hrs./Week	Term Work and Practical: 50	

Course Pre Requisites:

Good knowledge of mathematics (Covered in Year 1,2)

Course Objective:

Course Outcomes:

As a part of this course, students will:

- Understand OR problem and associated models.
- Understand Linear Algebra.
- Use transportation and assignment problems.
- Use PERT for modeling.
- Use Inventory Control System.
- Apply queuing theory and modulation techniques.

Topics to Be Covered:

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.

Text Books:

3. *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

Essence of Indian Traditional Knowledge

TEACHING SCHEME:	EXAMINATION SCHEME:	CREDITS ALLOTTED:
Lectures: 2Hrs./Week	Semester Examination: 50 marks	Theory: Non-Credits

Course Objective: Give exposure to the repositories of our indigenous knowledge and wisdom which have evolved over centuries, and they still continue to serve social and cultural functions.

Course Outcomes: Student will be able to

1. Understand basic principles, thought process, reasoning and inference of Indian Traditional Knowledge Systems.
2. Recognize wisdom of Sanskrit literature and its importance in modern society with rapid technological advancements.
3. Be familiar with scientific worldview and basic principles of Yoga and holistic health care system
4. Understand that sustainability is at the core of Indian Traditional Knowledge Systems connecting society and nature.

Topics to Be Covered:

UNIT – I [6 Hours]

Basic Structure of Indian Knowledge System: The historical impact of social change on traditional knowledge systems. Indigenous Knowledge (IK), Traditional Knowledge (TK) Vs western knowledge traditional knowledge vis-à-vis formal knowledge. Significance of TK Protection, value of TK in global economy, Role of Government to harness TK. The Scheduled Tribes and Other Traditional Forest Dwellers (Recognition of Forest Rights) Act, 2006, Plant Varieties Protection and Farmer's Rights Act, 2001 (PPVFR Act).

.

UNIT – II [6 Hours]

Modern Science and Indian Knowledge System; Mathematics in India, Early Historical Period, The Classical Period, The Classical Period, post-Āryabhaṭa, Features of Indian Mathematics. Early Chemical Techniques, Atomism in Vaiśeṣika, Chemistry in Early Literature, Indian Philosophy Sāṁkhya, Yoga, Vaiśeṣika, Nyāya, Mīmāṁsā, Vedānta, Sāṁkhya.

UNIT – III [6 Hours]

Yoga and Holistic Health care: Ayurveda for Life, Health and Well-being ,Definition of Ayurveda, The Principles of Ayurvedic Healing, Treating diseases to restore health, Astanga Ayurveda.

Reference Books:

1. V. Sivaramakrishnan (Ed.), Cultural Heritage of India-course material, Bharatiya Vidya Bhavan, Mumbai. 5th Edition, 2014.

2. Swami Jitatmanand, Holistic Science and Vedant, Bharatiya Vidya Bhavan.
3. "Knowledge Traditions and Practices of India" Kapil Kapoor, Michel Danino
4. RN Jha, Science of Consciousness Psychotherapy and Yoga Practices, Vidyanidhi Prakasham, Delhi, 2016

E-Resources:

1. https://www.youtube.com/watch?v=LZP1StpYEPM
2. http://nptel.ac.in/courses/121106003/