KOLHAPUR INSTITUTE OF TECHNOLOGY'S COLLEGE OF ENGINEERING

(EMPOWEREDAUTONOMOUS)

GOKUL SHIRGAON, KOLHAPUR

Department of Computer Science and Business Systems

Curriculum Structure and Syllabus for

Second Year B.Tech in

Computer Science and Business Systems

(As Per NEP, 2020)

	Teaching and Evaluation Scheme for Second Year Semester-III										
Sr. No.	Course Code	Cluster	Course	Teaching Scheme				Evaluation Scheme			
				L	T	P	Credits	Components	Max	Min Marks for Pass	
1	UCBPC0301	PC	Formal Language and Automata Theory	3	1	0	4	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
2	UCBPC0302	PC	Object Oriented Programming using JAVA	3	0	0	3	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
3	UCBBS0303	BS	Computational Statistics	3	0	0	3	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
4	UCBPC0304	PC	Software Engineering	3	0	0	3	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
5	UCBEM0305	EM	Fundamentals of Management	2	0	0	2	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
6	UCBVE0306	VEC	Indian Constitution	2	0	0	2	ISE	50	20	
7	UCBPC0331	PC	Object Oriented Programming using JAVA Laboratory	0	0	2	1	ISE	25	10	
								ESE (POE)	50	20	
8	UCBBS0332	BS	Computational Statistics Lab	0	0	2	1	ISE	25	10	
								ESE (POE)	25	10	
9	UCBPC0333	PC	Software Engineering Laboratory	0	0	2	1	ISE	25	10	
10	UCBIL0371	OJT	Mini-Project-I	0	0	2	1	ISE	25	10	
11	UCBMM0341	MM	Fundamentals of Marketing (MM-I)	2	0	0	2	ESE	100	40	
			Total	**18**	**1**	**08**	**23**	**Total**	**850**		
	Total Contact Hours =27, Total Credits =23										

Title of the Course: Formal Language & Automata Theory	**L**	**T**	**P**	**Credit**
Course Code: UCBPC0301	**3**	**1**	**-**	**4**

Course Pre-Requisite: Fundamentals of Discrete Mathematics

Course Description: This course to study about different formal languages and how these languages are validated with respect their machines. Also, this course provides the knowledge of designing abstract process model for any discrete system through state transition diagrams.

Course Learning Objectives:

1. To understand the different formal languages and its acceptors
2. To apply the knowledge of automata Theory, grammars & Regular Expressions for solving language problems
3. To design different language acceptors to validate the respective formal languages
4. To understand decidable and undecidable languages with time complexity

Course Outcomes:

CO	**After the completion of the course the student should beable to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Understand the different formal languages and its acceptors	II	Understand
CO2	Apply the knowledge of automata Theory, grammars & Regular Expressions for solving language problems.	III	Apply
CO3	Design different language acceptors to validate the respective formal languages.	III	Apply
CO4	Understand decidable and undecidable languages with time complexity.	II	Understand

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3												1	
CO2	2	2	1										1	
CO3	2	2	1										2	
CO4	2	3		1									1	

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Introduction to Finitinte Automata	**8 Hours**

Alphabet, languages and grammars, productions and derivation, 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.

Unit 2: Kleen's Theorem	**7 Hours**

Kleene's theorem, pumping lemma for regular languages, Myhill-Nerode theorem and its uses, minimization of finite automata. Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms.

Unit 3: Context-free languages, Pushdown Automata and Context Sensitive Languages	**8 Hours**

Context-free grammars (CFG) and languages (CFL), Chomsky and Greibachnormal 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.

Unit 4: Turing machines	**8 Hours**
The basic model for Turing machines (TM), Turing recognizable(recursively enumerable) and Turing-decidable (recursive) languages, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs	
Unit 5: Undecidability	**6 Hours**
Church-Turing thesis, universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice s theorem, undecidable problems about languages.	
Unit 6: Basic Introduction to Complexity	**8 Hours**
Introductory ideas on Time complexity of deterministic and nondeterministic Turing machines, P and NP, NP- completeness, Cook's Theorem, other NP -Complete problems.	
Textbooks:	
1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman (2008). Introduction to Automata Theory, Languages, and Computation Pearson. 2. John C. Martin (2002). Introduction to Languages and Theory of Computation. McGraw-Hill Higher Education.	
References:	
1. Harry R. Lewis and Christos H. Papadimitriou (1981). Elements of the Theory of Computation. Springer. 2. Michel Sipser (2013). Introduction to Theory of Computation. Cengage Learning.	

Title of the Course: Object Oriented Programming using Java	**L**	**T**	**P**	**Credit**
Course Code: UCBPC0302	**2**	-	**-**	**2**

Course Pre-Requisite: Knowledge of basic Programming languages like C, C++

Course Description: The course aims to provide knowledge about Object Oriented Programming Design and concepts by using JAVA language.

Course Learning Objectives:

1. To expose the students to concepts of Object-Oriented Paradigm.
2. To demonstrate the fundamental and object-oriented concepts of Java
3. To showcase the knowledge of exception handling,file handling and collections in design of applications.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Interpret the fundamentals of OOPs concepts in Java programming	II	Understand
CO2	Apply knowledge of various OOPs concepts for designing the applications.	III	Apply
CO3	Apply knowledge of exception handling,file handling and collections in design of applications.	III	Apply

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	-	1	-	-	3	-	-	-	-	-	-	-	2	1
CO2	-	2	2	-	3	-	-	-	-	-	-	1	3	2
CO3	-	2	2	-	3	-	-	-	-	-	-	1	3	2

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Introduction to Object Oriented Design	**7 Hours**

Introduction to procedural & object-oriented programming, Limitations of procedural programming, need of object-oriented programming, Fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism.

Unit 2: Basics of Core Java Programming	**8 Hours**

The Java Programming Environment- JVM, JIT Compiler, Byte Code Concept, A Simple Java Program, Source File Declaration Rules, Comments, Data Types, Variables, Operators, Strings, Input and Output, Control Flow, Big Numbers, Arrays, Jagged Array.

Unit 3: Object Oriented Design in Java	**8 Hours**

Objects and Classes: Object Oriented Programming Concepts, Declaring Classes, Declaring Member Variables, Defining Methods, Constructor, Passing Information to a Method or a Constructor, Creating and using objects, Controlling Access to Class Members, Static Fields and Methods, this keyword, Object Cloning, Class Design.

Unit 4: Interface, Inheritance and Packaging	**7 Hours**

Implementing an Interface, Using an Interface as a Type, Evolving Interfaces, Default Methods. Inheritance: Definition, Super classes, and Subclasses, Overriding and Hiding Methods,

Polymorphism, Inheritance Hierarchies, Super keyword, Final Classes and Methods, Abstract Classes and Methods, casting, Design Hints for Inheritance, Nested classes & Inner Classes, finalization, and garbage collection. Packages: Class importing, creating a Package, naming a Package, Using Package Members, Managing Source and Class Files. Developing and deploying (executable) Jar File

Unit 5: Exception and I/O Streams	**8 Hours**

Exception: Definition, Dealing with Errors, The Classification of Exceptions, Declaring Checked Exceptions, Throw an Exception, Creating Exception Classes, Catching Exceptions, Catching Multiple Exceptions, Re-throwing, and Chaining Exceptions, finally clause, Advantages of Exceptions, Tips for Using Exceptions. I/O Streams: Byte Stream – Input Stream, Output Stream, DataInputStream, DataOutputStream, FileInputStream, FileOutputStream, Character Streams, BufferedStream, Scanner, File, RandomAccesFile

Unit 6: Collection and Database Programming	**7 Hours**

Collections: Collection Interfaces, Concrete Collections- List, Queue, Set, Map, the Collections Framework.

Textbooks:

1. Core Java Volume I Fundamentals. Edition 8. by Cay Horstmann and Gary Cornell. Pearson

2. Core Java- Volume II Advanced Features by Cay Horstmann and Gary Cornell. Edition 8. Pearson

References:

1.The Java Tutorial: A Short Course on the Basics by Raymond Gallardo, Scott Hommel, Sowmya Kannan. Edition 6. Addison-Wesley Professional

2.JAVA-The Complete Reference by Herbert Schildt. Edition 9. Oracle Press, Mcgraw Hill

3.JAVA™ How to program by Deitel Paul, Deitel Harvey. Edition 10. PHI Learning.

Title of the Course: Computational Statistics	**L**	**T**	**P**	**Credit**
Course Code: UCBBS0303	**3**	-	-	**3**

Course Pre-Requisite: Basic statistics, Probability Distributions.

Course Description: In this course student will develop understanding of current and modern computational statistical approaches and their applications to a variety of real datasets and to apply principles of data science.

Course Learning Objectives:

1. Understanding of different Computational Techniques in Statistics.
2. To gain the knowledge of how to analyze multivariate data.
3. Application of different Statistical data analysis Techniques.
4. To provide an idea of using data aggregation techniques.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Make use of Multivariate Normal Distribution for solving problems.	III	Apply
CO2	Analyze the data using different Statistical data analyzing techniques.	IV	Analyze
CO3	Apply the concept Clustering.	III	Apply
CO4	Summarize the data using data aggregation technique.	II	Understand

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2	2												1
CO2	3	2		3									2	2
CO3	3	2											1	2
CO4	2	2											2	2

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Multivariate Normal Distribution	**7 Hours**

1.1 Multivariate Statistical Analysis

1.2 Estimation of parameters

1.3 Distribution of Linear combination of multivariate normal variates

1.4 Independence of sub vector

1.5 Conditional distribution

Unit 2: Discriminant Analysis	**8 Hours**

1.1 Statistical background

1.2 Mathematical model

1.2 Linear discriminant function analysis

1.3 Estimating linear discriminant functions.

1.4 Properties.

Unit 3: Principal Component Analysis	**8 Hours**
1.1 Principal components 1.2 Algorithm for conducting Principal Component Analysis. 1.3 Maths behind PCA 1.4 Deciding on how many principal components to retain. 1.5 Dimensionality Reduction	
Unit 4: Factor Analysis	**8 Hours**
1.1 Factor analysis model 1.2 Covariance structure for orthogonal factor model 1.3 Factor extraction method 1.4 Types of factors Analysis 1.5 Factor Rotation-Transformation of factor analysis solutions, Factor scores.	
Unit 5: Clustering	**8 Hours**
1.1 Introduction 1.2 Types of clustering. 1.3Bby partitioning methods, hierarchical clustering, K-Means Clustering 1.4 Correlations and distances 1.5 Profiling and Interpreting Clusters	
Unit 6: Data Aggregation, Group Operations	**6 Hours**
1.1 Introduction 1.2 Data Aggregators 1.3 Group by Mechanics 1.4 Data Aggregation 1.5 Groupwise Operations and Transformations 1.6 Types of data aggregation 1.7 Applications of data aggregation 1.8 Pivot Tables and Cross Tabulations.	

Textbooks:

1. T.W. Anderson (2009). An Introduction to Multivariate Statistical Analysis. Wiley.

2. J.D. Jobson (1992). Applied Multivariate Data Analysis Volume II: Categorical and Multivariate Methods. Springer.

References:

1. D.A. Belsey, E. Kuh and R.E. Welsch (2004). Regression Diagnostics, Identifying Influential Data and Sources of Collinearety. Wiley-Interscience.

2. J. Neter, W. Wasserman and M.H. Kutner (2003). Applied Linear Regression Models. McGraw-Hill Education.

3. A.S. Mulaik. (2009). The Foundations of Factor Analysis. Chapman and Hall/CRC.

4. D.C. Montgomery and E.A. Peck (2006). Introduction to Linear Regression Analysis. Wiley India Pvt Ltd.

5. M.R. Anderberg (1973). Cluster Analysis for Applications. Academic Press Inc.

6. D.F. Morrison (1990). Multivariate Statistical Analysis. McGraw-Hill Education.

Title of the Course: Software Engineering	**L**	**T**	**P**	**Credit**
Course Code: UCBPC0304	**3**	-	-	**3**

Course Pre-Requisite: Knowledge of Software Engineering Process, Programming Languages

Course Description: This course provides fundamentals of Software Development Life Cycle (SDLC), principles of software engineering practices and introductory concepts of software project management.

Course Learning Objectives:

1. To provide the idea of decomposing the given problem into Analysis, Designing, Implementation, Testing and Maintenance phases.
2. To provide an idea of using various process models in the software industry according to given circumstances.
3. To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Understand the various aspects of software engineering starting from requirements gathering till the Beta testing	II	Understand
CO2	Analyze the gathered requirements and be able to apply a suitable life cycle model for software development.	IV	Analyze
CO3	Design reliable software using appropriate design patterns and standards.	III	Apply
CO4	Implement and test the developed software by the modern software project management tools and techniques.	IV	Analyze

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	3	3									2	2	2
CO2	2	2			2							2	2	2
CO3	2	3	3		2							2	2	2
CO4	2	2	3		3							2	2	2

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Introduction	**7 Hours**

Introduction: Programming in the small vs. programming in the large; software project failures and importance of software quality and timely availability; of software engineering towards successful execution of large software projects; emergence of software engineering as a discipline, Software Engineering Historical Development from Jackson Structured Programming to Agile Development.

Unit 2: Software Project Management	**8 Hours**

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 3: Agile Software Engineering and Software Quality Management and Reliability	**8 Hours**

Agile Software Engineering: Concepts of Agile Methods, Extreme Programming; Agile Process Model - Scrum, Feature; Scenarios and Stories Software quality; Garvin's quality dimensions, McCall's quality factor, ISO 9126 quality factor; Software Quality Dilemma; Introduction to Capability Maturity Models (CMM and CMMI); Introduction to software reliability, reliability models and estimation.

Unit 4: Software Requirements Analysis, Design and Construction	**8 Hours**
Introduction to Software Requirements Specifications (SRS) and requirement elicitation techniques; techniques for requirement modelling – 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 5: Software Design	**6 Hours**
Design Concepts, Function Oriented Design: Structure Charts, Structured Design Methodology, An Example. Object Oriented Design: OO Concepts, Unified Modelling Language (UML), A Design Methodology, Examples. Detailed Design, Verification, Metrics Case Study Study the design of Library Software / or any project statement	
Unit 6: Software Testing	**8 Hours**
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; Unit Testing, Integration Testing, System Testing and Acceptance Testing.	
Textbooks:	
1. Sommerville, I (2016). Software Engineering. 10th Edition, Pearson Education Limited. 2. Pressman, Roger S., Maxim, Bruce R. (2015). SOFTWARE ENGINEERING; A Practitioner's Approach (8th). Mcgraw-Hill.	
References:	
1. Ivar Jacobson, Harold Bud Lawson, Pan-Wei Ng, Paul E. McMahon and Michael Goedicke (2019). The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons. Morgan & Claypool Publishers 2. Carlo Ghezzi, Jazayeri Mehdi and Mandrioli Dino (2002). Fundamentals of Software Engineering. Pearson. 3. Michael Jackson (1995). Software Requirements and Specification: A Lexicon of Practice, Principles and Prejudices. Addison-Wesley. 4. Ivar Jacobson, Grady Booch and James Rumbaugh (2002). The Unified Development Process. Pearson Education. 5. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides (1994). Design Patterns: Elements of Object-Oriented Reusable Software. Addison-Wesley Professional.	

6. Norman E Fenton and Shari Lawrence Pfleeger (1996). Software Metrics: A Rigorous and Practical Approach. Cengage Learning.

7. Bertrand Meyer (1997). Object-Oriented Software Construction. Prentice Hall.

8. Ivar Jacobson (1992). Object Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley.

9. Bertrand Meyer (2013). Touch of Class: Learning to Program Well with Objects and Contracts. Springer.

10. Martin Fowler (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley.

Title of the Course: Fundamentals of Management	**L**	**T**	**P**	**Credit**
Course Code: UCBEM0305	**2**	-	-	**2**

Course Pre-Requisite: Self-management, overview of business organizations and their operations and challenges faced by them.

Course Description: The course helps the students to know the necessary skills and functions required to be an efficient manager in a contemporary business environment.

Course Learning Objectives:

1. Understand current management concepts, models and managerial processes and their integration into business activities.
2. Differentiate between various aspects of managing organizations.
3. Explain the concepts and significance of management in business.
4. Analyze the role of management in society and in our lives.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO 1	Understand current management concepts, models and managerial processes and their integration into business activities.	II	Understand
CO 2	Differentiate between various psychological and ethical aspects of managing organizations.	II	Understand
CO 3	Explain the concepts and significance of suitable leadership and management in business.	II	Understand
CO 4	Analyze the role of management in society and in our lives and in businesses	IV	Analyze

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1			2											
CO2		1												
CO3														2
CO4	1						2							

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Management Theories	**6 Hours**
Concept and Foundations of Management, Evolution of Management Thoughts [Pre-Scientific Management Era (before 1880), Classical management Era (1880-1930), Neo-classical Management Era (1930-1950), Modern Management era (1950-on word). Contribution of Management Thinkers: Taylor, Fayol, Elton Mayo etc. Latest management theory by Jack Welch.	
Unit 2: Functions of Management	**4 Hours**
Functions of Management- Planning, Organizing, Staffing, Directing, Controlling	
Unit 3: Organization Behavior	**5 Hours**
Introduction, Personality, Perception, Learning and Reinforcement, Motivation, Group Dynamics, Power & Influence, Work Stress and Stress Management, Decision Making, Problems in Decision Making, Decision Making, Organizational Culture, Managing Cultural Diversity.	
Unit 4: Organizational Design	**5 Hours**
Classical, Neoclassical and Contingency approaches to organizational design; Organizational theory and design, Organizational structure (Simple Structure, Functional Structure, Divisional Structure, Matrix Structure). Its examples.	

Unit 5: Managerial Ethics	**3 Hours**
Ethics and Business, Business and Social Responsibility, International Standards, Corporate Governance, Corporate Citizenship, Corporate Social Responsibility.	
Unit 6: Leadership	**7 Hours**
Leadership: Concept, Nature, Importance, Attributes of a leader, developing leaders across the organization, Leadership Grid.	

Home Assignment:

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.

1. Topic: Corporate social responsibility (CSR) and HRM implications: What does it mean to be socially responsible within an increasingly financially driven market economy?

2. Topic: Leaders are Born, Not Made! The debate

Textbooks:

1. Richard L. Daft (2016). Understanding the Theory and Design of Organizations. Cengage Learning India Private Limited.
2. Stephen P. Robbins, David A. Decenzo (2016). Fundamentals of Management. Pearson Education.

References:

1. Stephen P. Robbins, Timothy A. Judge, Neharika Vohra (2018). Organizational Behaviour. Pearson Education.
2. Harold Koontz, O'Donnell and Heinz Weihrich (2012). Essentials of Management. Tata McGraw Hill.

Title of the Course: Object Oriented Programming using Java Laboratory **Course Code:** UCBPC0331	**L**	**T**	**P**	**Credit**
	-	**-**	**2**	1

Course Pre-Requisite: Knowledge of basic Programming languages like C, C++

Course Description: This course is designed to develop Java programming expertise. Upon completion, students should be able to write programs in Java. Emphasis is on Class design, Implementation, File Handling, Exception Handling and Collection Framework.

Course Learning Objectives:

1. To expose the students to concepts of Object-Oriented Paradigm
2. To make students implement program solutions for given problems using OOPs concepts.
3. To make students use exception handling and collections in the application development.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Interpret the fundamentals of OOPs concepts in Java programming	II	Understand
CO2	Build Java program to implement OOP concepts	III	Apply
CO3	Develop Java applications by applying OOPs concepts and Java Features.	III	Apply

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	-	2	2	-	3	-	-	1	-	-	-	2	2	1
CO2	-	2	3	-	3	-	-	1	-	-	-	2	3	2
CO3	-	2	3	-	3	-	-	1	-	-	-	2	3	2

Assessments:

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50%, and 50% weights respectively.

Assessment	Marks
ISE	25
ESE	50

ISE are based on practical performance/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc. ESE: Assessment is based on oral examination

Course Contents:

Experiment No. 1: Creating classes and initializing objects.	**02 Hrs.**

Aim and Objectives: Applying Classes and Object concepts**.**
Theoretical Background: classes and objects
Experimentation: Develop a Java Program to implement class and create its objects.
Results and Discussions: Objects are created, data members of class are initialized, members functions can be called.
Conclusion: You can call non-static members of class using objects, constructor get called when you create object of class

Experiment No. 2: Classes and Objects	**02 Hrs.**

Aim and Objectives: Applying Classes and Object concepts.
Outcomes: Student will be able to use class, its members and create objects.
Theoretical Background: classes and objects
Experimentation: Develop a Java Program to implement class and create its objects.
Results and Discussions: Objects are created, data members of class are initialized, members functions can be called.
Conclusion: You can call static members of class using class name, static data member is common between all objects**.**

Experiment No.3: Implementing multiple Inheritance in Java.	**02 Hrs.**

Aim and Objectives: Multiple Inheritance in Java.

Outcomes: Students will be able to create multiple inheritance with the help of Interfaces.

Theoretical Background: Java do not support multiple inheritance with classes, if programmer need to multiple inheritance structure she needs to use interfaces.

Experimentation: Developing Java program with interface inheritance. Results and Discussions: Interface methods are overridden in class, calling overridden functions using objects is done.

Conclusion: The Class which inherit interface, must have to provide override methods declared in interface. The default access to members of interface is public

Experiment No. 4: Inheritance in Java	**02 Hrs.**

Aim and Objectives: Inheritance using classes.

Outcomes: Students will be able to create hierarchical inheritance with the help of classes.

Theoretical Background: Child class should extend base class to acquire base class properties.

Experimentation: Create Separate Engine, Tyre, and Door Class. Create a Car class using these classes. And show functionality of each component in the car.

Results and Discussions: When class extends from another class. The extending class is child and the class which is extended is base. The child class object can call public and protected members of base class. Child class can call its own members despite of its access specifier. When you create object of child class its constructor get called. The child class constructor calls the base class constructor.

Conclusion: Child class acquire the properties of base class.

Experiment No. 5: Creating Packages.	**02 Hrs.**

Aim and Objectives: Creating JAR(Java Archive)-package.
Outcomes: Student will be able to create deployable packages of Java application.
Theoretical Background: Application need multiple classes. You cannot have all classes in single file. So, you have to club together the required classes of your application. This can be done with packages in Java. The archive of these packages is called as JAR. Jar file can be created as executable and non-executable jar files.
Experimentation: -Develop a mathematical package for Statistical operations like Mean, Median, Average, Standard deviation. Create a sub package in the math package -convert. In "convert" package provide classes to convert decimal to octal, binary, hex and vice-versa. Develop application program to use this package, and build executable jar file of it.
Results and Discussions: The default access of class is package. You can create executable version of JAR file.

Conclusion: The applications developed using Java, which contains multiple ".class" files and hierarchical directory structure should be deployed using JAR

Experiment No. 6: Understanding Constructors	**02 Hrs.**

Aim and Objectives: Understanding and using parameterized constructors in Java.
Outcomes: Students will be able to use multiple constructor in single class.
Theoretical Background: Java provides zero parameter constructor, program do not need to write it. This is the reason, the zero-parameter constructor is called as default constructor. Programmer can write its own constructor which is zero parameter or parameterized constructor. When programmer choose to write constructor in class, java do not provide default constructor. In this case the programmer has to write zero parameter constructor on its own.
Experimentation: Develop a class Expr to create and evaluate given expression. Constructor accepts the expression as String. For example, Expr("x^2") or Expr("sin(x)+3*x"). If the parameter in the constructor call does not represent a legal expression, then the constructor throws an IllegalArgumentException. The message in the exception describes the error. Provide eval(double num) and eval(int num) method to evaluate given expression and return evaluated answer.

For example, if Expr represents the expression 3*x+1, then func.value(5) is 3*5+1, or16.Finally,getDefinition() returns the definition of the expression. This is just the string that was used in the constructor that created the expression object.

Results and Discussions: Calling specific constructor from available ones Passing arguments to object. Use of "this" keyword to identify current object class members.

Conclusion: Java class can have overload of constructors; this keyword identifies current object under execution

Experiment No. 7: Exception Handling in Java.	**02 Hrs.**

Aim and Objectives: Handling exception in java program.

Outcomes: Students will be able to deal/tackle runtime error in java program.

Theoretical Background: Java have two error reporting mechanisms. One is compile time and one is runtime. The compile time error deals with syntax of Java programming language and you cannot get byte code of java program unless your program is error free. The run time error is called as exceptions. They represent semantic issues in program.

Experimentation: Write a class to represent Roman numerals. The class should have two constructors. One constructs a Roman numeral from a string such as "XVII" or "MCMXCV". It should throw a NumberFormatException if the string is not a legal Roman numeral. The other constructor constructs a Roman numeral from an int. It should throw a NumberFormatException if the int is outside the range 1 to 3999.In addition, the class should have two instance methods. The method toString() returns the string that represents the Roman numeral. The method toInt() returns the value of the Roman numeral as an int

Results and Discussions: You can handle exceptions using try, catch blocks, The mission critical code which should execute in the both scenarios- i.e whether exception occurred, or application executed normally- should be written in finally block.

Conclusion: Runtime errors can be handled with try, catch, and finally block

Experiment No. 8: File Handling	**02 Hrs.**

Aim and Objectives: Performing file handling using Java Program.

Outcomes: Student will be able to use I/O streaming classes in Java for file handling.

Theoretical Background: Java provide Input and Output streaming classes. They can be used to deal with input output devices. Here students will learn how to write file operation using Java program.

Experimentation: Take file name as input to your program, if file is existing the open and display contents of the file. After displaying contents of file ask user – do you want to add the data at the end of file. If a user gives yes as response, then accept data from user and append it to file. If file in not existing, then create a fresh new file and store user data into it. User should type exit on new line to stop the program.

Results and Discussions: Java can deal with file using byte streams and character streams. It has variety of classes to deal with file operation.

Conclusion: Java application can perform file handling.

Experiment No. 9: Buffered Streams in Java.	**02 Hrs.**

Aim and Objectives: Using wrapper classes to reduce disk operations in file handling.

Outcomes: Student will be able to develop efficient file handling programs.

Theoretical Background: Buffered Reader/Writer, DataInputStream/ Output Stream, Print Writer are wrapper classes, which improves efficiency of Java program. Student should be able to use buffering of data to avoid disk I/O operations for every read and write of character/word to and from file.

Experimentation: Take Student information such as name, age, weight, height, city, phone from user and store it in the file using DataOutputStream and FileOutputStream and Retrieve data using DataInputStream and FileInputStream and display the result.

Results and Discussions: Using buffering for file handling is efficient than byte or character data.

Conclusion: Students can use buffers to deal with file handling to avoid unnecessary disk access.

Experiment No. 10: Collection Framework in Java	**02 Hrs.**

Aim and Objectives: Using Java Collection Framework Create Objects.

Outcomes: Student will be able to implement List, Map, Queue interface and sub classes

Theoretical Background: List, Map and Queue are java collection Interfaces. Implementation classes like ArrayList, HashMap used to store the object and data.

Experimentation: create ArrayList to store department names as CSE, IT, MECH, CIVIL, ENV, PROD, BIO, ENTC.

Include Below ArrayList Methods:

1. add 2. remove 3. size

4. display all elements using system.out.println

5. add an Iterator to ArrayList to retrieve elements and display

6. clear ()

7. display all elements using system.out.println

Results and Discussions: ArrayList, HashMap provides inbuilt methods to store and travel object data.

Conclusion: Students can use ArrayList, HashMap for store and access object data more efficiently

Textbooks:

1. Core Java Volume I Fundamentals (Unit 1 to Unit 4) by Cay Horstmann and Gary Cornell Edition 8. Pearson
2. Core Java- Volume II Advanced Features (Unit 5 and Unit 6) by Cay Horstmann and Gary Cornell. Edition 8. Pearson

References:

1. 1.The Java Tutorial: A Short Course on the Basics by Raymond Gallardo, Scott Hommel, Sowmya Kannan. Edition 6. Addison-Wesley Professional
2. JAVA-The Complete Reference by Herbert Schildt. Edition 9. Oracle Press, Mcgraw Hill
3. JAVA™ HOW TO PROGRAM by Deitel Paul, Deitel Harvey. Edition 10. PHI Learning
4. Thinking in Java by Bruce Eckel. Edition 4. Prentice Hall.

Title of the Course: Computational Statistics Laboratory **Course Code:** UCBBS0332	**L**	**T**	**P**	**Credit**
	-	**-**	**2**	**1**

Course Pre-Requisite: Multivariate calculus, Statistical Methods, and Data Analysis.

Course Description: The course is designed to provide application of Python language on real life data set to the students.

Course Learning Objectives:

1. Understand Python programming language for analysis of data.
2. Perform operations on real life data sets using Python Programming language.
3. Construction of different types of graphs for analysis of data.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Demonstrate the fundamentals of Python Program for data analysis	II	Understand
CO2	Design the graphs for the given data sets	III	Apply
CO3	Analyze the results by applying various statistical methods by using Python.	IV	Analyze

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	2												1
CO2	3	2											2	3
CO3	3	2											2	3

Assessments:

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50%, and 50% weights respectively.

Assessment	Marks
ISE	25
ESE	25

ISE are based on practical performance/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc. ESE: Assessment is based on oral examination

Course Contents:

Experiment No. 1: Write Simple Python Program containing python statements, expressions, control flow, functions, and its execution.	**02 Hrs.**
Experiment No. 2: Write python program using function and numeric types	**02 Hrs.**
Experiment No.3: Visualization in Python using matplotlib	**02 Hrs.**
Experiment No. 4: Plotting graph using matplotlib package	**02 Hrs.**
Experiment No. 5: Test the performance by using Simple Regression	**02 Hrs.**
Experiment No. 6: Test the performance by using Multiple Regression	**02 Hrs.**
Experiment No. 7: Performing Discriminant Analysis	**02 Hrs.**
Experiment No. 8: Performing Principal Component Analysis	**02 Hrs.**
Experiment No. 9: Performing Factor Analysis	**02 Hrs.**
Experiment No. 10: Performing Cluster Analysis	**02 Hrs.**
Experiment No. 11: Performing Data aggregation group operations	**02 Hrs.**

Textbooks:

1. T.W. Anderson (2009). An Introduction to Multivariate Statistical Analysis. Wiley.
2. J.D. Jobson (1992). Applied Multivariate Data Analysis Volume II: Categorical and Multivariate Methods. Springer.

References:

3. D.A. Belsey, E. Kuh and R.E. Welsch (2004). Regression Diagnostics, Identifying Influential Data and Sources of Collinearety. Wiley-Interscience.
4. J. Neter, W. Wasserman and M.H. Kutner (2003). Applied Linear Regression Models. McGraw-Hill Education.
5. A.S. Mulaik. (2009). The Foundations of Factor Analysis. Chapman and Hall/CRC.
6. D.C. Montgomery and E.A. Peck (2006). Introduction to Linear Regression Analysis. Wiley India Pvt Ltd.
7. M.R. Anderberg (1973). Cluster Analysis for Applications. Academic Press Inc.
8. D.F. Morrison (1990). "Multivariate Statistical Analysis". McGraw-Hill Education.

Title of the Course: Software Engineering Laboratory **Course Code:** UCBPC0333	**L**	**T**	**P**	**Credit**
	-	**-**	**2**	1

Course Pre-Requisite: Knowledge of Software Engineering Process, Programming Languages

Course Description: This course to apply the knowledge of software engineering process and design the solution for real time problems using design tools and technologies.

Course Learning Objectives:

1. To provide the idea of decomposing the given problem into Analysis, Designing, Implementation, Testing and Maintenance phases.
2. To provide an idea of using various process models in the software industry according to given circumstances.
3. To gain the knowledge of how Analysis, Design, Implementation, Testing and Maintenance processes are conducted in a software project.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Analyze and identify requirements for real time problems	II	Understand
CO2	Design and implement various software design models using modern engineering tools	IV	Analyze
CO3	Provide appropriate solutions for the real time problems using software engineering methodology	III	Apply
CO4	Design test cases for various real time problems	IV	Analyze

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1		3		3		2					2	2	3	3
CO2		2	3		3	2	1	2			2	2	2	2
CO3		2	3		3	2	1	2			2	2	2	2
CO4			2		3	2					2	2	2	2

Assessments:

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50%, and 50% weights respectively.

Assessment	Marks
ISE	25
ESE	-

ISE are based on practical performed/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc. ESE: Assessment is based on oral examination

Course Contents:

Assignment No 1: Prepare the problem statement for the given applications.	**02 Hrs.**
Assignment No 2: Develop Software Requirement Specification (SRS) for the applications considered in the assignment no. 1	**02 Hrs.**
Assignment No 3: Design the data flow diagram for the application.	**02 Hrs.**
Assignment No 4: Design the class diagrams for the application.	**02 Hrs.**
Assignment No 5: Design the Use-case diagrams for the application.	**02 Hrs.**
Assignment No 6: Design the activity diagrams for the application.	**02 Hrs.**
Assignment No 7: Design the interaction diagrams for the application.	**02 Hrs.**
Assignment No 8: Implementation of UML in C++	**02 Hrs.**
Assignment No 9: Writing test cases for the functions in the application and create test report.	**02 Hrs.**
Assignment No 10: Study the configuration management tools(GitHub) and project management tools(jira) used in software development life cycle.	**02 Hrs.**

Instructions for Students:

1. Individual student must select project application from the given list.
2. All the assignment solutions need to prepare in given format and the same should submit in Moodle.
3. For designing UML diagrams students must use modern open-source tools such as draw.io, UModel etc

List of Project Applications:

1. Railway Reservation System
2. Airline Management System
3. Hospital Management System
4. Examination Management System
5. KIT Bus Service System

Textbooks:

1. Ian Sommerville (2006). Software Engineering. Pearson Education.
2. Pressman, Roger S., Maxim, Bruce R. (2015). SOFTWARE ENGINEERING, A Practitioner's Approach (8th). Mcgraw-Hill.

References:

1. Carlo Ghezzi, Jazayeri Mehdi, Mandrioli Dino (2003). Fundamentals of Software Engineering. Prentice Hill
2. Ivar Jacobson, Grady Booch, James. Rumbaugh (1999). The Unified Development Process. Pearson Education.
3. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (1995). Design Patterns: Elements of Object-Oriented Reusable Software. Addison-Wesley.

Title of the Course: Value Education Course Laboratory **Course Code:** UCBVE0334	**L**	**T**	**P**	**Credit**
	-	**-**	**2**	1

Course Pre-Requisite: human values and culture

Course Description: This course gives insights about human values and culture. Also, this course helps to develop social responsibilities.

Course Learning Objectives:

1. Understand the meaning of values and culture
2. Develop as socially responsible citizens
3. Identify and correlate the relationship between values and human rights

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Understand the meaning of values and culture	II	Understand
CO2	Develop as socially responsible citizens	III	Apply
CO3	Identify and correlate the relationship between values and human rights.	III	Apply

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	-							1				2		
CO2	-							1				2		
CO3	-							1				2		

Assessments:

Teacher Assessment:

One component of In Semester Evaluation (ISE) 100% weights.

Assessment	Marks
ISE	25

ISE are based on practical performance/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc.

Course Contents:

Experiment No. 1:	**02 Hrs.**
Discussion on philosophy	
Experiment No. 2:	**02 Hrs.**
Discussion on spiritual personalities	
Experiment No.3:	**02 Hrs.**
Discussion of Ethical Values	
Experiment No. 4:	**02 Hrs.**
Discussion on Leadership qualities	
Experiment No. 5:	**02 Hrs.**
Discussion on religious values and harmony	
Experiment No. 6:	**02 Hrs.**
Discussion on life Moralities	
Experiment No. 7:	**02 Hrs.**
Discussion on power of mind	
Experiment No. 8:	**02 Hrs.**
Discussion on Human Rights	
Experiment No. 9:	**02 Hrs.**
Discussion on Universal Values.	
Experiment No. 10:	**02 Hrs.**
Discussion on benefits of yoga.	

Textbooks:
1. Das, M.S. & Gupta, V.K. : Publications, Social Values among Young adults: A changing Scenario New Delhi M D Publications, 1995 2. Jash, P. Glimpses of Hindu Cults and Culture, Sundeep Prakashan, Delhi, 1997 NCERT Education in Values, New Delhi, 1992
References:
1. **https://onlinecourses.nptel.ac.in/noc23_hs89/preview** 2. **https://nptel.ac.in/courses/109104068**

KOLHAPUR INSTITUTE OF TECHNOLOGY'S COLLEGE OF ENGINEERING KOLHAPUR (EMPOWERED AUTONOMOUS)

Department of Computer Science and Business Systems

	Teaching and Evaluation Scheme for Second Year Semester-IV										
Sr. No.	Course Code	Cluster	Course	Teaching Scheme				Evaluation Scheme			
				L	T	P	Credits	Components	Max	Min Marks for Pass	
1	UCBPC0401	PC	Operating Systems	3	0	0	3	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
2	UCBPC0402	PC	Database Management System	3	0	0	3	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
3	UCBPC0403	PC	Computer Organization & Architecture	3	0	0	3	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
4	UCBEM0404	EM	Operations Research	3	0	0	3	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
5	UCBAE0405	AE	Business Communication & Value Science –III	2	0	0	2	ISE-I	10		40
								ISE-II	10		
								MSE	30		
								ESE	50	20	
6	UCBPC0431	PC	Operating Systems Laboratory	0	0	2	1	ISE	25	10	
								ESE(POE)	50	20	
7	UCBPC0432	PC	Database Management System Laboratory	0	0	2	1	ISE	25	10	
								ESE(POE)	50	20	
8	UCBPC0433	PC	Computer Organization & Architecture Laboratory	0	0	2	1	ISE	25	10	
9	UCBAE0434	AE	Business Communication & Value Science –III Laboratory	0	0	2	1	ISE	25	10	
10	UCBCC0435	CC	Co-Curricular Course-II	0	0	2	1	ISE	25	10	
11	UCBIL0471	OJT	Mini Project-II	0	0	2	1	ISE	25	10	
12	UCBMM0441	MM	Financial and Cost Accounting (MM-II)	3	0	0	3	ESE	100	40	
			Total	**17**	**0**	**12**	**23**	**Total**	**850**		
	Total Contact Hours=29, Total Credits=23										

Title of the Course: Operating System	**L**	**T**	**P**	**Credit**
Course Code: UCBPC0401	**3**	**-**	**-**	**3**

Course Pre-Requisite: Fundamentals of Electronics and Computer, Data Structure and Algorithms, Computer Organization and Architecture

Course Description: This is one of the core course of Computer Science & Engineering Programme. In this course you will become familiar with the core concepts of OS - how OS work, how a processes & threads are created, inter-process communication & synchronisation, the various scheduling algorithms, memory management & memory allocation strategies, etc.

Course Learning Objectives:

1. To summarize the basic concept of operating system and process management.
2. To apply the concept of resource management and Process Scheduling
3. To describe concepts of Process Synchronization and deadlock
4. To explain concept of Memory Management and virtual memory

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Summarize basic concept of operating system and process management	I	Remember
CO2	Apply the concept of resource management and Process Scheduling	III	Apply
CO3	Describe concepts of Process Synchronization and deadlock	II	Analyse
CO4	Explain concept of Memory Management and virtual memory	II	Understand

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	1	1									1			
CO2	2	2			1				1	2				
CO3	2	2		1	1							2		1
CO4	2	3		2						1		1	1	2

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Introduction to Operating System	**8 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. 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 .Self-Study: OS Booting System

Unit 2: Process Scheduling	**7 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 3: Process Synchronization	**8 Hours**

Over view of Process Synchronization, Inter-process Communication:Concurrent processes, 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:

Unit 4: Deadlocks	**7 Hours**
Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Deadlock allocation graphs Banker's algorithm, Deadlock detection and Recovery.	
Unit 5: Memory Management	**7 Hours**
Basic concept, Logical and Physical address maps, Memory allocation: Contiguous Memory allocation – Fixed and variable partition– Internal and External fragmentation and Compaction. Concept Of Copy on Wite.	
Unit 6: Virtual memory	**8 Hours**
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).	

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2009). Operating System Principles. Wiley India.
2. Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne (2010). Operating System Concepts Essentials. John Wiley & Sons Inc.
3. Dhananjay Dhamdhere (2017). Operating Systems A Concept Based Approach. Mc-Graw Hill.

References:

1. William Stallings (2012). Operating Systems: Internals and Design Principles. Pearson Education
2. Charles Patrick Crowley (2017). Operating System: A Design-oriented Approach. McGraw Hill Education.
3. Milan Milenkovic (2001). Operating Systems –Concepts and design. TMGH.
4. Andrew S. Tanenbaum (2015). Modern Operating Systems. Pearson Education International.

Swayam Courses:

1.https://onlinecourses.nptel.ac.in/noc20_cs04/preview

Title of the Course: Database Management Systems	**L**	**T**	**P**	**Credit**
Course Code: UCBPC0402	**3**	**-**	**-**	**3**

Course Pre-Requisite: Set Theory, Operating Systems, and Data Structures.

Course Description: This Course is designed to understand the internals of Database System, with elaboration from Database Design, Using Relational Database (using SQL) and the transaction concepts

Course Learning Objectives:

1. To understand the fundamentals of database management systems programming.
2. To design an appropriate database for a given problem.
3. To write SQL queries to design & manage the database.
4. To illustrate Transactions, Concurrency and Recovery apply to database system.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Describe the fundamentals of database management systems programming.	II	Understand
CO2	Design appropriate databases for a given problem.	III	Apply
CO3	Write SQL queries to design & manage the database	III	Apply
CO4	Illustrate Transactions, Concurrency and Recovery apply to database system	II	Understand

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	1		2											
CO2	1	3	2	3									2	2
CO3	1	1	1	3	2								2	2
CO4	1		1											

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Introduction to Database	**8 Hours**

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

Database System Architecture: Data Abstraction, Data Independence, Data definition language (DDL), Data Manipulation Language (DML).

Unit 2: Data models & Relational query languages	**7 Hours**

Data models: Entity-relationship model, network model, relational and object-oriented data models, integrity constraints, and data manipulation operations.

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 3: Introduction to NoSQL Database and Relational database design	**7 Hours**

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

Unit 4: Query Optimization & Storage strategies	**8 Hours**

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

Storage strategies: Indices, B-trees, Hashing.

Unit 5: Transaction processing:	**8 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 6: Database Security	**7 Hours**
Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.	

Textbooks:

1. Abraham Silberschatz, S. Sudarshan, and Henry F. Korth (2019). Database System Concepts. McGraw-Hill.
2. E. F. Codd (1990). The relational model for database management. Addison-Wesley Longman Publishing Co.

References:

1. Alfred V. Aho and J. D. Ullman (1988). Principles of Database and Knowledge – Base Systems. Computer Science Press
2. R. Elmasri and S. Navathe (2011). Fundamentals of Database Systems. Pearson Education, Inc-Addison-Wesley
3. Serge Abiteboul, Richard Hull and Victor Vianu (1994). Foundations of Databases. Pearson Education, Inc- Addison-Wesley

Title of the Course: Computer Organization & Architecture	**L**	**T**	**P**	**Credit**
Course Code: UCBPC0403	**3**	**-**	**-**	**3**

Course Pre-Requisite: Basic functional units of a computer system

Course Description: The course helps the students to gather a conceptual overview of memory, registers, arithmetic unit, control unit, and input/output components. It also helps to examine computer design trade-offs. This course will be also helpful for exams like GATE.

Course Learning Objectives:

1. To learn the fundamentals of Combinational and Sequential Circuits.
2. To learn the CUP architecture & Data Representation.
3. To learn the CPU and Control unit Design.
4. To learn the fundamentals peripheral device and Memory organization.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Understand the Combinational and Sequential Circuits.	II	Understand
CO2	Explain the CUP architecture & digital representation of data in a computer system.	II	Understand
CO3	Analyze the performance aspects of the control unit.	III	Apply
CO4	Demonstrate the concepts of organization	III	Apply

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	1	2											1	
CO2	2	2												
CO3	1	1	1	1										2
CO4	2	1		1									1	2

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Introduction	7 **Hours**

Revision of basics in Boolean logic and Combinational/Sequential Circuits. Functional blocks of a computer: CPU, memory, input-output subsystems, control unit.

Unit 2: Instruction set architecture of a CPU	**7 Hours**

Registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Outlining instruction sets of some common CPUs.

Unit 3: Data representation	**8 Hours**

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 4: Control unit and memory design	**8 Hours**

Introduction to x86 architecture. CPU control unit design: Hardwired and micro-programmed design approaches, design of a simple hypothetical CPU. Memory system design: Semiconductor memory technologies, memory organization.

Unit 5: Peripheral devices and their characteristics	**8 Hours**
Input-output subsystems, I/O device interface, I/O transfers – program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes – role of interrupts in process state transitions, I/O device interfaces – SCII, USB. Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards.	
Unit 6: Memory organization	7 **Hours**
Parallel Processors: Introduction to parallel processors, Concurrent access to memory and cache coherency. Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies.	
Textbooks:	
1. Mano M. M. (1993). Computer System Architecture. Prentice Hall of India, New Delhi. **2.** V. Carl Hamacher and Safwat G. Zaky (2011). Computer Organization and Embedded Systems. McGraw Hill Higher Education.	
References:	
1. John P. Hayes (2017). Computer Architecture and Organization. McGraw Hill Education. 2. William Stallings (2016). Computer Organization and Architecture: Designing for Performance. Pearson Education India. 3. Vincent P. Heuring and Harry F. Jordan (1996). Computer System Design and Architecture. Pearson Education India.	

Title of the Course: Operations Research	**L**	**T**	**P**	**Credit**
Course Code: UCBEM0404	**3**	**-**	**-**	**3**

Course Pre-Requisite: Fundamentals of Mathematics and Basic functional areas of Engineering.

Course Description: Operations Research is the study of Optimization techniques. OR has wide scope of applications in various fields as it helps solve various Industry problems. The various Optimization techniques like LPP, Assignment, Sequencing, Queuing, Inventory Control will be helpful to obtain optimal solutions to various problems.

Course Learning Objectives:

1. To provide strong foundation for understanding the fundamental principles and laws of Operation Research
2. To understand Linear programming, network Analysis and queuing, transportation models.
3. To understand and solve the problems regarding management and application of various models.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Demonstrate essential operations research modelling tools to solve the real-world business problems.	II	Understanding
CO2	Make use of Course knowledge to solve LPP, Transportation, and Queuing and Game theory, Network Management problems.	III	Applying
CO3	Analyze the concept of LPP, game theory models, queuing theory models for solving the engineering problems.	IV	Analyzing
CO4	Choose the appropriate models in operation research for a given practical problem.	V	Evaluating

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	2									1			
CO2	3	2									1			
CO3	3	2									1			
CO4	3	2									1			

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Linear Programming Problems	**8 Hours**

Revision of basics in Boolean logic and Combinational/Sequential Circuits. Functional blocks of a computer: CPU, memory, input-output subsystems, control unit.

Unit 2: Inventory Control	8 **Hours**

Objectives of Inventory Control, Concept of inventory costs, Classification of Inventory Models, Concept of EOQ, EOQ Models.

Unit 3: Network Models	**7 Hours**

Project definition, Concept and construction of network diagram, CPM and PERT, Determination of critical paths, Estimation of Project time and its variance in PERT using statistical principles, Concept of project crashing.

Unit 4: Transportation and Assignment Model	**8 Hours**

TP - Formulation, Balanced & unbalanced situations, Solution methods – NWCR, least cost and VAM, test for optimality (MODI method).

AP - Examples, Definitions – decision variables, constraints, formulation, Balanced & unbalanced situations, Solution method – Hungarian, test for optimality.

Unit 5: Queuing Theory	**8 Hours**
Definitions – queue (waiting line), waiting costs, characteristics (arrival, queue, service discipline) of queuing system, queue types (channel vs. phase). Classification of Queuing models, Kendall's notation for representing Queuing models, M/M/1 Model.	
Unit 6: Game Theory	6 **Hours**
Introduction, Terms used in Game Theory, Game with pure strategies, Game with mixed strategies, Dominance theory, Algebraic method, Graphical solution of 2xn and mx2 games, linear programming approach for game theory.	

Textbooks:

1. S.D.Sharma (2012). Operations Research. KEDAR NATH.
2. Kanti Swarup, Man Mohan and P.K.Gupta (2019). Operations Research. Sultan Chand & Sons, New Delhi
3. Manohar Mahajan (2016). Operations Research. Dhanpat Rai & Sons
4. H. A.Taha (2018). Operations Research- An Introduction. Pearson.
5. D. S. Hira, P. K. Gupta (2015). Operations Research. S Chand & Co. Ltd New Delhi

References:

1. Wagner H. M.(1980). Principles of Operations Research. Prantice- Hall of India, New Delhi.
2. F.S. Hiller and G.J. Lieberman (2005). Introduction to Operations Research. McGraw Hill.
3. A. Ravi Ravindran (2007). Operations Research and Management Science-Hand Book. CRC Press Inc.
4. Wiest & Levy (1970). Management Guide to PERT/CPM. Prentice Hall.

Title of the Course: Business Communication and Value Science-III	**L**	**T**	**P**	**Credit**
Course Code: UCBAE0405	**2**	**-**	**-**	**2**

Course Pre-Requisite: Basic Knowledge of English (verbal and written)

Completion of all units from Semester 1 and 2 courses: BCVS-I and BCVS II

Course Description: This course will help students to understand oneself, be motivated, understand and apply various aspects of cross-cultural communication, learn technical writing skills, and be a societal technocrat.

Course Learning Objectives:

1. To understand the basic principles of SWOT & life positions.
2. To identify & respect pluralism in cultural spaces.
3. To understand, apply & analyze the tools of technical writing.
4. To understand Artificial intelligence & recognize its impact in daily life

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Understand the basic principles of SWOT & life positions.	II	Understand
CO2	Identify & respect pluralism in cultural spaces	III	Apply
CO3	Understand, apply & analyze the tools of technical writing	IV	Analyze
CO4	Understand Artificial intelligence & recognize its impact in daily life	II	Understand

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1									1	2		2		2
CO2								1	2	3		1		1
CO3						2		2	1	3		1		3
CO4								1	1	3		1		3

Assessments:

Teacher Assessment:

Two components of In Semester Evaluation (ISE), One Mid Semester Examination (MSE) and one End Semester Examination (ESE) having 20%, 30% and 50% weights respectively.

Assessment	Marks
ISE-1	10
MSE	30
ISE-2	10
ESE	50

ISE-1 and ISE-2 are based on assignment/declared test/quiz/seminar/Group Discussions etc.

MSE: Assessment is based on 50% of course content (Normally first three modules)

ESE: Assessment is based on 100% course content with 60-70% weightage for course content (normally last three modules) covered after MSE.

Course Contents:

Unit 1: Principles of SWOT and life positions	**5 Hours**
Understanding SWOT, Applications of SWOT, SWOT Vs. TOWS	
Unit 2: Motivation in real life	**4 Hours**
Maslow's theory - Recognize how motivation helps real life - Leverage motivation in real-life scenarios, stories, videos	
Unit 3: Pluralism in cultural spaces	**6 Hours**
Awareness and respect for pluralism in cultural spaces Concepts and differences between global, glocal and translocational culture Cross-cultural communication (Verbal and non-verbal communication): Implications, common mistakes made in cross-cultural communication, Gender awareness	
Unit 4: Role of science in nation building	**5 Hours**
Introduction to role of science in nation building- discussion through Augmented Reality, role of science post- independence	

Unit 5: Introduction to technical writing: Technical report writing	**6 Hours**
Importance of reports, objectives, characteristics Types/categories, formats (Manuscript, memo, letter) Structure/elements of manuscript reports Report writing and presentation in teams with the help of ICT tools	
Unit 6: Recognize the importance of AI	**4 Hours**
AI in Everyday Life Communicating with machines Ted talk videos	

Web References:

5. Examples of Technical Writing for Students: https://freelancewriting.lovetoknow.com/kinds-technical-writing
6. Skills of a Good Technical Writer https://clickhelp.com/clickhelp-technical-writing-blog/11-skills-of-a-good-technical-writer/
7. Benefits and challenges of cultural diversity in the workplace: https://www.hult.edu/blog/benefits-challenges-cultural-diversity-workplace/

Online Resources:

1. https://youtu.be/CsaTslhSDI
2. https://m.youtube.com/watch?feature=youtu.be&v=IIKvV8_T95M
3. https://m.youtube.com/watch?feature=youtu.be&v=e80BbX05D7Y
4. https://m.youtube.com/watch?v=dT_D68RJ5T8&feature=youtu.be

Title of the Course: Operating Systems Laboratory (Unix) **Course Code:** UCBPC0431	**L**	**T**	**P**	**Credit**
	-	**-**	**2**	1

Course Pre-Requisite: Fundamentals of Electronics and Computer.

Course Description: This is one of the core course of Computer Science & Engineering Programme. In this course students will become familiar with the core concepts of OS - how OS work, how a processes & threads are created, inter-process communication & synchronization, the various scheduling algorithms, memory management & memory allocation strategies, etc. This course will be also helpful for exams like GATE.

Course Learning Objectives:

1. To summarize the features of different operating systems and implement basic concepts of OS.
2. To compare algorithms for CPU scheduling, Disk Scheduling and page replacement policies.
3. To demonstrate and implement Concepts on File and I/O system

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Summarize the features of different operating systems and Impliment basic consecpts of OS	III	Apply
CO2	Compare algorithms for CPU scheduling, Disk Scheduling and page replacement policies.	IV	Analyse
CO3	Demonstrate and impliment Concepts on File amd I/O system	III	Apply

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	1		2				1	1	1		2	1	2
CO2	2	2		2				1	1			1		1
CO3	1	2	2		1				1	1		1		3
CO4	1	1		2				1	1	1		2	1	2

Assessments:

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50%, and 50% weights respectively.

Assessment	Marks
ISE	25
ESE	50

ISE are based on practical performed/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc. ESE: Assessment is based on oral examination

Course Contents:

Experiment 1: Installation and configuration of any one of the operating system (UBUNTU, CENT-OS Linux etc). Execute System Calls - fork(), exit(), exec(), wait(), waitpid(), getpid(), getppid().	**02 Hrs.**
Experiment 2:Program to create process using Fork().	**02 Hrs.**
Experiment 3:Evaluate performance of scheduling algorithms. a) FCFS b) SJF c) RR d) PRIORITY in terms of turnaround time, response time.	**02 Hrs.**
Experiment 4 : Program of Multithreading . POSIX threads. pthread_create(), pthread_join(), pthread_exit(), pthread_self()	**02 Hrs.**
Experiment 5 :Program For classical problem of Inter-Process Synchronization(IPS).	**02 Hrs.**
Experiment 6 :Write a Program For Inter Process Communication using Pipe.	**02 Hrs.**
Experiment 7:Write a Program to simulate page Replacement algorithm. a) FIFO b) LRU c) Optimal	**02 Hrs.**
Experiment 8: Program to simulate disk Scheduling Algorithm.	**02 Hrs.**
Experiment 9: Program related to shell Programming.	**02 Hrs.**
Experiment 10: Program for file allocation methods- a) continuous b) Linked c) Indexed.	**02 Hrs.**

Instructions for Students:

1. Continuous assessment shall be based on experiments performed, submission of results of program in the form of report/journal, timely completion, attendance, understanding, efficient codes, punctuality and neatness.
2. Practical/Oral examination shall be based on the practical's performed in the lab.
3. Lab assessment of 25 marks shall be based on continuous assessment and performance in Practical/Oral examination.

Textbooks:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2009). Operating System Principles. Wiley India.
2. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne (2010). Operating System Concepts Essentials. John Wiley & Sons Inc.
3. Dhananjay Dhamdhere (2017). Operating Systems A Concept Based Approach. Mc-Graw Hill

References:

1. William Stallings (2012). Operating Systems: Internals and Design Principles. Pearson Education
2. Charles Patrick Crowley (2017). Operating System: A Design-oriented Approach. McGraw Hill Education
3. Milan Milenkovic (2001). Operating Systems –Concepts and design. TMGH.
4. Andrew S. Tanenbaum (2015). Modern Operating Systems. Pearson Education International.
5. Maurice J. Bach. (2015). Design of the Unix Operating Systems. Pearson Education India.

Title of the Course: Database Management Systems Laboratory **Course Code:** UCBPC0432	**L**	**T**	**P**	**Credit**
	-	**-**	**2**	**1**

Course Pre-Requisite: Basics of ER notations, Normalization and Data Structures.

Course Description: This course to apply the basic knowledge of database design strategies to solve real world database applications. Also implementing the relational database models using DDL and solve the functional queries using various DML commands.

Course Learning Objectives:

1. To Analyze and design ER model for a application.
2. To Create relational database model for a real-world application.
3. To solve data manipulation queries in the application by writing SQL queries.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Analyze and design ER model for a application	VI	Create
CO2	Create relational database model for a real-world application	VI	Create
CO3	Solve data manipulation queries in the application by writing SQL queries.	VI	Create

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1		3		3		2					2	2	3	3
CO2		2	3		3	2	1	2			2	2	2	2
CO3		2	3		3	2	1	2			2	2	2	2
CO4			2		3	2					2	2	2	2

Assessments:

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50%, and 50% weights respectively.

Assessment	Marks
ISE	25
ESE	50

ISE are based on practical performed/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc. ESE: Assessment is based on oral examination

Course Contents:

Assignment No 1: Design ER Model and apply the normalization for a given application database.	**02 Hrs.**
Assignment No 2: Create relational database using DDL commands and mention the necessary key constraints for required attributes.	**02 Hrs.**
Assignment No 3: Write SQL queries to fetch required data using basic DML commands.	**02 Hrs.**
Assignment No 4: Write SQL queries to fetch required data using set operations.	**02 Hrs.**
Assignment No 5: Write SQL queries to fetch required data using join operations.	**02 Hrs.**
Assignment No 6: Write SQL queries to fetch required data using aggregate functions.	**02 Hrs.**
Assignment No 7: Write SQL queries to fetch required data using String operations.	**02 Hrs.**
Assignment No 8: Write SQL queries to fetch required data using nested queries	**02 Hrs.**
Assignment No 9: Create triggers and stored functions to perform data modification across the dependent tables.	**02 Hrs.**
Assignment No 10: Write SQL queries for creating views and perform the modification operations on view tables.	**02 Hrs.**
Assignment No 11: Create program interface with database and perform SQL operations over the database.	**02 Hrs.**

Instructions for Students:

1. Each practical batch will be assigned as particular database application.
2. All the assignment should perform on the assigned database application.

List of Applications:

1. Railway Reservation System
2. Airline Management System
3. Hospital Management System
4. Examination Management System
5. KIT Bus Service System

Textbooks:

1. Abraham Silberschatz, S. Sudarshan, and Henry F. Korth (2019). Database System Concepts. McGraw-Hill.
2. E. F. Codd (1990). The relational model for database management. Addison-Wesley Longman Publishing Co.

References:

1. Alfred V. Aho and J. D. Ullman (1988). Principles of Database and Knowledge – Base Systems. Computer Science Press
2. R. Elmasri and S. Navathe (2011). Fundamentals of Database Systems. Pearson Education, Inc-Addison-Wesley
3. Serge Abiteboul, Richard Hull and Victor Vianu (1994). Foundations of Databases. Pearson Education, Inc- Addison-Wesley.

Title of the Course: Computer Organization & Architecture Lab **Course Code:** UCBPC0433	**L**	**T**	**P**	**Credit**
	-	**-**	**2**	**1**

Course Pre-Requisite: Fundamentals of Electronics and Computer, Fundamentals of Computer Programming

Course Description: This is one of the core course of Computer Science & Engineering Programme. In this course you will become familiar with the core concepts of OS - how OS work, how a processes & threads are created, inter-process

Course Learning Objectives:

1. To learn combinational & Sequential Circuit.
2. To learn Assembly language programs for various operations.
3. To learn interfacing of different peripherals.

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Design Combinational & Sequential Circuit.	III	Apply
CO2	Develop a Machine language program to perform different tasks.	III	Apply
CO3	Develop Machine language program to interface peripherals to microprocessor	III	Apply

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	1	2											1	
CO2	2	2			2									
CO3	1	1	2	1	3								1	2

Assessments:

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50%, and 50% weights respectively.

Assessment	Marks
ISE	25

ISE are based on practical performed/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc. ESE: Assessment is based on oral examination

Course Contents:

Experiment No 1: Implementation of Encoder and Decoder (Simulation).	**02 Hrs.**
Experiment No 2: Implementation of Counter/Linear Feedback Shift Register (Simulation).	**02 Hrs.**
Experiment No 3: Write a Machine language program to perform addition, subtraction, Multiplication and Division	**02 Hrs.**
Experiment No 4: Write a Machine language program to sort data in an ascending order.	**02 Hrs.**
Experiment No 5: Write a Machine language program to sort data in a descending order.	**02 Hrs.**
Experiment No 6: Write a Machine language program to Count odd and even integers from a series of memory locations.	**02 Hrs.**
Experiment No 7: Write a Machine language program to find smallest number in a block.	**02 Hrs.**
Experiment No 8: Write a Machine language program to find largest number in a block.	**02 Hrs.**
Experiment No 9: Write a Machine language program to transfer data one memory location to another memory location.	**02 Hrs.**
Experiment No 10: Write a Machine language program to keypad Interfacing	**02 Hrs.**
Experiment No 11: Write a Machine language program to LCD Interfacing	**02 Hrs.**

Textbooks:

1. Mano M. M. (1993). Computer System Architecture. Prentice Hall of India, New Delhi.
2. A. David Patterson and John L. Hennessy (2009). Computer Organization and Design: The Hardware/Software Interface.Morgan Kaufmann.
3. V. Carl Hamacher and Safwat G. Zaky (2011). Computer Organization and Embedded Systems. McGraw Hill Higher Education.

References:

1. John P. Hayes (2017). Computer Architecture and Organization. McGraw Hill Education.
2. William Stallings (2016). Computer Organization and Architecture: Designing for Performance. Pearson Education India.
3. Vincent P. Heuring and Harry F. Jordan (1996). Computer System Design and Architecture. Pearson Education India.

Title of the Course: Business Communication and Value Science-III Lab **Course Code:** UCBAE0434	**L**	**T**	**P**	**Credit**
	-	**-**	**2**	1

Course Pre-Requisite: Business Communication and Value Science-III.

Course Description: This course will help students to understand oneself, be motivated, understand and apply various aspects of cross-cultural communication, learn technical writing skills, and be a societal technocrat

Course Learning Objectives:

1. To apply & analyze the basic principles of SWOT & life positions.
2. To apply the concepts of Global, glocal and translocational.
3. To analyze cross cultural communication.
4. To apply the science of Nation building

Course Outcomes:

CO	**After the completion of the course the student should be able to**	**Bloom's Cognitive**	
		Level	**Descriptor**
CO1	Apply & analyze the basic principles of SWOT & life positions.	III	Apply
CO2	Apply the concepts of Global, glocal and translocational	III	Apply
CO3	Analyze cross cultural communication	IV	Analyze
CO4	Apply the science of Nation building	III	Apply
CO5	Analyze the social issues around and provide solutions	IV	Analyze

CO-PO Mapping:

CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1									1	2		2		2
CO2								1	2	3		1		1
CO3						2		2	1	3		1		3
CO4								1	1	3		1		3
CO5						1			2	2		1		2

Assessments:

Teacher Assessment:

One component of In Semester Evaluation (ISE) and one End Semester Examination (ESE) having 50%, and 50% weights respectively.

Assessment	Marks
ISE	25
ESE	--

ISE are based on practical performed/ Quiz/ Mini-Project assigned/ Presentation/ Group Discussion/ Internal oral etc. ESE: Assessment is based on oral examination

Course Contents:

Experiment 1: Applying SWOT in real life scenarios: Create your SWOT, SWOT Vs. TOWS-The Balancing Act- Ted talk on biomimicry. Turning your threat into opportunity-two people mutually identifying opportunities from each other's threats.	**02 Hrs.**
Experiment 2: Presentation: on the strengths identified during SWOT to survive in the VUCA World. Group presentations of 10 mins each.	**02 Hrs.**
Experiment 3: Leverage motivation in real-life scenarios: recognize how motivation helps in real life. Scenario based activity on identifying and leveraging motivation. Presenting findings and approaches as groups. They need to explain the idea of motivation with the help of examples.	**02 Hrs.**
Experiment 4 : Learn and Exchange: Group activity in which participants need to learn the following four greetings of a state (different from their own) and exchange it with another group: (Indicative only): “Good morning/afternoon/evening/night”, "please", "thank you", "you're welcome" and "I'm sorry"	**02 Hrs.**
Experiment 5 : Debate: on Global, glocal, tanslocational impacts (topics to be decided by the faculty or suggested by the students) Culture shock: Group activity to perform skits based on situations	**02 Hrs.**

Experiment 6: Gender awareness campaign: Groups to present the detailed plan of Gender awareness campaigns with four different themes: College, Workplace, Family, Friends	**02 Hrs.**
Experiment 7: Group presentations: on eminent scientists and mathematicians of ancient India.	**02 Hrs.**
Experiment 8: "Voice of the Future" Activity How will a voice assistant evolve in 25 years from now? Each group will present a skit. Design your college in the year 2090 AI in Everyday Life Discussion in groups on given topics and then cross sharing of discussion points amongst the groups. Debate- Will machines control us in future?	**02 Hrs. (+4 hours home activity)**
Experiment 9: Visit to rural area/ underprivileged parts of city to address some of the local issues; if relevant, suggest a practical technology solution to the issues	**02 Hrs.**
Experiment 10: Group presentations on the above issues and solutions	**02 Hrs.**

Web References:

1. Examples of Technical Writing for Students: https://freelance-writing.lovetoknow.com/kinds-technical-writing
2. Skills of a Good Technical Writer https://clickhelp.com/clickhelp-technical-writing-blog/11-skills-of-a-good-technical-writer/
3. Benefits and challenges of cultural diversity in the workplace: https://www.hult.edu/blog/benefits-challenges-cultural-diversity-workplace/

Online References:

1. https://youtu.be/CsaTslhSDI
2. https://m.youtube.com/watch?feature=youtu.be&v=IIKvV8_T95M
3. https://m.youtube.com/watch?feature=youtu.be&v=e80BbX05D7Y
4. https://m.youtube.com/watch?v=dT_D68RJ5T8&feature=youtu.be