FACULTY OF ENGINEERING

Scheme of Instruction & Examination

And

Syllabi

B.E. III and IV Semester

Of

Four Year Degree Programme

In

CSE (AI&ML)

(With effect from the academic year 2020-2021)

As approved in the faculty meeting held on 11.8.2021

Issued by

Dean, Faculty of Engineering
Osmania University, Hyderabad
2021

SCHEME OF INSTRUCTION & EXAMINATION
B.E. - III SEMESTER
CSE (AI&ML)

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P / D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
Theory Courses										
1	BS207MT	Mathematics – III (Probability & Statistics)	3	-	-	3	30	70	3	3
2	HS105CSM	Finance and Accounting	3	-	-	3	30	70	3	3
3	PC301CSM	Data Structures and Algorithms	3	-	-	3	30	70	3	3
4	ES302EC	Digital Electronics	3	-	-	3	30	70	3	3
5	ES303CSM	Python Programming	3	-	-	3	30	70	3	3
6	PC304CSM	Automata Languages and Computation	3	-	-	3	30	70	3	3
7	*MC306HS	Gender Sensitization	3	-	-	3	30	70	3	0
Practical/ Laboratory Courses										
8	PC 351 CSM	Data Structures and Algorithms Lab	-	-	2	2	25	50	3	1
9	PC 352 CSM	Python Programming Lab	-	-	2	2	25	50	3	1
			21	**-**	**4**	**25**	**260**	**590**	**-**	**20**

PC: Professional Course **PE**: Professional Elective,
HS: Humanities and social Science **MC**: Mandatory Course
L: Lecture T: Tutorial **P**: Practical **D**: Drawing
CIE: Continuous Internal Evaluation, **SEE**: Semester End Examination (Univ. Exam)

Note:

1. Each contact hour is a Clock Hour
2. The practical class can be of two and half hour (clock hours) duration as per the requirement of a particular laboratory.

B.E III SEMESTER Syllabus

Computer Science and Engineering (AI & ML)

Course Code	Course Title						Core/Elective
BS207MT	**Mathematics – III (Probability & Statistics)**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- To introduce the solution methodologies for second order Partial Differential Equations with applications in engineering
- To provide an overview of probability and statistics to engineers

Course Outcomes

After completing this course, the student will be able to:

1. Solve field problems in engineering involving PDEs.
2. They can also formulate and solve problems involving random variables and apply statistical methods for analyzing experimental data.

UNIT-I: Introduction of Probability, Conditional probability, Theorem of Total probability, Baye’s Theorem and its applications, Random variables, Types of random variables, Probability mass function and Probability density function, Mathematical expectations.

UNIT-II: Discrete probability distributions: Binomial and Poisson distributions, Mean, variance, moment generating function and evaluation of statistical parameters for these distributions, Moments, Skewness and Kurtosis.

UNIT-III: Continuous probability distributions, Uniform, Exponential and Normal distributions, Mean, variance, moment generating function and evaluation of statistical parameters for these distributions

UNIT-IV: Curve fitting by the method of least squares: Fitting of straight lines, second degree parabolas and more general curves, Correlation, regression and Rank correlation. Test of significance: Large sample test for single proportion, difference of proportions, single mean, difference of means, and difference of standard deviations.

UNIT-V: Test for single mean, difference of means and correlation coefficients, test for ratio of variances, Chi-square test for goodness of fit and independence of attributes, -control charts for measurements (X and R charts) – Control charts for attributes (p, c and np charts) – Tolerance limits – Acceptance sampling

Suggested Readings:

1. R.K.Jain & Iyengar, “Advanced Engineering Mathematics”, Narosa Publications.
2. B.S. Grewal, “Higher Engineering Mathematics”, Khanna Publishers, 2000.
3. P.Sivaramakrishna Das & C.Vijaya Kumar, “Engineering Mathematics” , Pearson India Education Services Pvt. Ltd.
4. N.P. Bali & M. Goyal, “A Text Book of Engineering Mathematics”, Laxmi Publications, 2010.
5. S.C.Gupta & V.K.Kapoor, “Fundamentals of Mathematical Statistics” , S.Chand Pub.
6. P. G. Hoel, S. C. Port & C. J. Stone, “Introduction to Probability Theory”, Universal Book Stall, 2003.
7. W. Feller, “An Introduction to Probability Theory and its Applications”, Vol. 1, Wiley, 1968.

Course Code	Course Title						Core/ Elective
HS105CSM	**Finance and Accounting**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

The course will introduce the students

- To provide basic understanding of Financial and Accounting aspects of a business unit
- To provide understanding of the accounting aspects of business
- To provide understanding of financial statements
- To provide the understanding of financial system
- To provide inputs necessary to evaluate the viability of projects
- To provide the skills necessary to analyse the financial statements

Course Outcomes

After successful completion of the course the students will be able to

1. Evaluate the financial performance of the business unit.
2. Take decisions on selection of projects.
3. Take decisions on procurement of finances.
4. Analyse the liquidity, solvency and profitability of the business unit.
5. Evaluate the overall financial functioning of an enterprise.

UNIT-I

Basics of Accounting: Financial Accounting–Definition- Accounting Cycle – Journal - Ledger and Trial Balance-Cash Book-Bank Reconciliation Statement (including Problems)

UNIT-II

Final Accounts: Trading Account-Concept of Gross Profit- Profit and Loss Account-Concept of Net Profit- Balance Sheet (including problems with minor adjustments)

UNIT-III

Financial System and Markets: Financial System-Components-Role-Considerations of the investors and issuers- Role of Financial Intermediaries. Financial Markets-Players- Regulators and instruments - Money Markets Credit Market- Capital Market (Basics only)

UNIT-IV

Basics of Capital Budgeting techniques: Time Value of money- Compounding-Discounting- Future Value of single and multiple flows- Present Value of single and multiple Flows- Present Value of annuities- Financial Appraisal of Projects– Payback Period, ARR-NPV, Benefit Cost Ratio, IRR (simple ratios).

UNIT-V

Financial statement Analysis: Financial Statement Analysis- Importance-Users-Ratio Analysis-liquidity, solvency, turnover and profitability ratios.

Suggested Readings:

1. Satyanarayana. S.V. and Satish. D., Finance and Accounting for Engineering, Pearson Education
2. Rajasekharan, Financial Accounting, Pearson Education
3. Sharma. S.K. and Rachan Sareen, Financial Management, Sultan Chand
4. Jonathan Berk, Fundamentals of Corporate Finance, Pearson Education
5. Sharan, Fundamentals of Financial Management, Pearson Education

<table>
<tr><td>Course Code</td><td colspan="4">Course Title</td><td colspan="2"></td><td>Core/ Elective</td></tr>
<tr><td><b>PC301CSM</b></td><td colspan="6"><b>Data Structures and Algorithms</b></td><td><b>Core</b></td></tr>
<tr><td rowspan="2">Prerequisite</td><td colspan="4">Contact Hours per Week</td><td rowspan="2">CIE</td><td rowspan="2">SEE</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>D</td><td>P</td></tr>
<tr><td><b>-</b></td><td><b>3</b></td><td><b>-</b></td><td><b>-</b></td><td><b>-</b></td><td><b>30</b></td><td><b>70</b></td><td><b>3</b></td></tr>
</table>

Objectives:

1. To develop proficiency in the specification, representation, and implementation of abstract data types and data structures.
2. To discuss the linear and non-linear data structures and their applications
3. To introduce the creation, insertion and deletion operations on binary search trees and balanced binary search trees.
4. To introduce various internal sorting, searching techniques and their time complexities

Outcomes:

After completing this course, the student will be able to:

1. Understand the importance of abstract data type and implementing the concepts of data structureusing abstract data type.
2. Evaluate an algorithm by using algorithmic performance and measures.
3. Distinguish between linear and non-linear data structures and their representations in the memoryusing array and linked list.
4. Apply the suitable data structure for a real world problem and think critically for improvement insolutions.
5. Determine the suitability of the standard algorithms: Searching, Sorting and Traversals

UNIT – I
Algorithms: Introduction, Algorithm Specifications, Recursive Algorithms, Performance Analysis ofan algorithm- Time and Space Complexity, Asymptotic Notations. **Arrays:**Arrays-ADT,Polynomials,Sparsematrices,Strings-ADT,PatternMatching.
UNIT – II
StacksandQueues: Stacks, Stacks using Arrays, Stacksusingdynamicarrays,Evaluation ofExpressions –EvaluatingPostfixExpression,InfixtoPostfix. **Queues:** Queues ADT, operations, Circular Queues, Applications
UNIT – III

Linked Lists: Singly Linked Lists and Chains, Linked Stacks and Queues,Polynomials, Operations forCircularly linked lists, Equivalence Classes,Sparsematrices,DoublyLinkedLists.

Hashing: Static Hashing, Hash Tables, Hash Functions, Overflow Handling, Theoretical Evaluation ofOverflow Techniques

UNIT – IV

Trees: Introduction, Binary Trees, Binary Tree Traversals, Heaps, BinarySearch trees (BST) : Definition, Searching an element, Insertion into a BST,Deletion from a BST.

Efficient Binary Search Trees: AVL Trees: Definition, Searching an element, Insertion into aAVL

UNIT – V

Graphs: Graph Abstract Data Type, Elementary Graph operations (DFS and BFS), Minimum CostSpanning Trees (Prim‘s and Kruskal‘s Algorithms).
Sorting and Searching: Insertion sort, Quick sort, Best computing time for Sorting, Merge sort, Heapsort, shell sort, Sorting on Several Keys, List and Table Sorts, Summary of Internal Sorting, Linear and
Binary Search algorithms.

SuggestedBooks:
1.Horowitz E, Sahni S and Susan Anderson-Freed, Fundamentals ofDatastructuresinC, 2ndEdition(2008),UniversitiesPress

ReferenceBooks:

1. Mark A Weiss, Data Structures and Algorithm Analysis In C, SecondEdition(2002),Pearson
2. Kushwaha D. S and Misra A.K, Data structures A ProgrammingApproachwithC,SecondEdition(2014),PHl.
3. Gilberg R. F and Forouzan B. A,Data structures: A PseudocodeApproachwithC,SecondEdition(2007),CengageLeaming
4. Tanenbaum A. M ,Langsam Y.Augenstein M. J,Data StructuresusingC,SecondEdition(2008),Pearson.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, CliffordStein,IntroductiontoAlgorithms,ThirdEdition(2009),MITPress
6. YedidyahLangsam, MosheJ.Augenstein ,AaronM.Tenenbaum, Data Structures Using C and C++ , Second Edition(2009),PHI

Course Code	Course title	Core/ Elective

ES302EC	**Digital Electronics**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- To learn the principles of digital hardware and support given by it to the software.
- To explain the operation and design of combinational and arithmetic logic circuits.
- To design hardware for real world problems.

Course Outcomes

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

1. Understand the design process of digital hardware, use Boolean algebra to minimize the logical expressions and optimize the implementation of logical functions.
2. Understand the number representation and design combinational circuits like adders, MUX etc.
3. Design Combinational circuits using PLDS and write VHDL code for basic gates and combinational circuits.
4. Analyse sequential circuits using flip-flops and design registers, counters.
5. Represent a sequential circuit using Finite State machine and apply state minimization techniques to design a FSM

UNIT – I

Design Concepts: Digital Hardware, Design process, Design of digital hardware. Introduction to logic circuits – Variables and functions, Logic gates and networks. Boolean algebra, Synthesis using gates, Design examples. Optimized implementation of logic functions using K-Map and Quine-McCluskey Tabular method

UNIT – II

Number representation: Addition and Subtraction of signed and unsigned numbers.

Combinational circuit building blocks: Half adder, Full adder, Multiplexers. Decoders. Encoders. Code converters, BCD to 7-segment converter, Arithmetic comparator circuits.

UNIT – III

Design of combinational circuits using Programmable Logic Devices (PLDs): General structure of a Programmable Array Logic (PAL), Programmable Logic Arrays(PLAs), Structure of CPLDs and FPGAs, 2- input and 3-input lookup tables (LUTs)

Introduction to Verilog HDL: Verilog code for basic logic gates, adders, decoders.

UNIT – IV

Sequential Circuits: Basic Latch, Gated SR Latch, gated D Latch, Master-Slave edge triggered flip-flops, T Flip-flop, JK Flip-flop, Excitation tables. Registers, Counters, Verilog code for flip-flops.

UNIT – V

Synchronous Sequential Circuits: Basic Design Steps, Finite State machine (FSM) representation using Moore and Mealy state models, State minimization, Design of FSM for Sequence Generation and Detection, Algorithmic State Machine charts.

Suggested Readings:

1. Morris Mano and Michael D CIletti, Digital Design, Pearson, fourth edition,2008
2. Zvi Kohavi, Switching and Finite Automata Theory, 3rd ed., Cambridge University Press-New Delhi, 2011.
3. R. P Jain, Modern Digital Electronics,4th ed., McGraw Hill Education (India) Private Limited, 2003
4. Ronald J.Tocci, Neal S. Widmer &Gregory L.Moss, “Digital Systems: Principles and Applications,” PHI, 10/e, 2009.

Samir Palnitkar, “Verilog HDL A Guide to Digital Design and Synthesis,” 2nd Edition, Pearson Education, 2006.

Course Code	Course title	Core/ Elective
ES303CSM	**PYTHON PROGRAMMING**	**Core**

Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

To learn

- Learn Syntax and Semantics and create Functions in Python.
- Handle Strings and Files in Python.
- Understand Lists, Dictionaries and Regular expressions in Python.
- Implement Object Oriented Programming concepts in Python
- Build Web Services and introduction to Network and Database Programming in Python.

Course Outcomes

After learning the contents of this course the student is able to

- Examine Python syntax and semantics and be fluent in the use of Python flow control and functions. Demonstrate proficiency in handling Strings and File Systems.
- Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Implement exemplary applications related to Network Programming, Web Services and· Databases in Python

UNIT – I: **Python Basics,** Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types

Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules (object, class, method creation, calling).**Sequences -** Strings, Lists, Tuples, Mapping and Set Types.

UNIT - II

FILES: File Objects, File Built-in Function [open()], File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules, Related Modules

Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management, *Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions, *Creating Exceptions, Why Exceptions (Now)?, Why Exceptions at All?, Exceptions and the sys Module, Related Modules: Modules and Files, Namespaces, Importing Modules, Importing Module Attributes, Module Built-in Functions, Packages, Other Features of Modules.

UNIT - III

Regular Expressions: Introduction, Special Symbols and Characters, Res and Python

Multithreaded Programming: Introduction, Threads and Processes, Python, Threads, and the Global Interpreter Lock, Thread Module, Threading Module, Related Modules

UNIT - IV

Web Basics: HTTP protocol, HTML, URL Basics, Web server, Web Framework, Introduction to WSGI.

FLASK Basics: FLASK installation, Basic Structure of application, Routing, variable rules, URL building, HTTP methods, Template, static files.

FLASK Advance: Request object, Response object, sending form data to template, Redirect errors, message flashing, file uploading, define and access database.

UNIT - V

Database Programming: Introduction, Python Database Application Programmer’s Interface (DB-API), Object Relational Managers (ORMs), Related Modules.

TEXT BOOKS:

1. Core Python Programming, Wesley J. Chun, Second Edition, Pearson. **(UNIT 1, UNIT 2, UNIT3, UNIT5)**
2. Flask Web Development, 2nd Edition, Miguel Grinberg, March 2018, O'Reilly Media, Inc., **(UNIT 4)**

REFERENCE BOOKS:

1. Think Python, Allen Downey, Green Tea Press

2. Introduction to Python, Kenneth A. Lambert, Cengage

3. Python Programming: A Modern Approach, Vamsi Kurama, Pearson

4. Learning Python, Mark Lutz, O'Reilly

5. Flask Framework Cookbook - Second Edition, Shalabh Aggarwal, July 2019, Packt Publishing.

Course Code	Course title	Core/ Elective

PC404CSM	**AUTOMATA LANGUAGES AND COMPUTATION**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- ➔ Develop a formal notation for strings, languages and machines.
- ➔ Design finite automata to accept a set of strings of a language.
- ➔ Design context free grammars to generate strings from a context free language and Convert them into normal forms.
- ➔ Identify the hierarchy of formal languages, grammars and machines.
- ➔ Distinguish between computability and non-computability and Decidability and undecidability.

Course Outcomes :After learning the contents of this course the student is able to

1. Write a formal notation for strings, languages and machines, Design finite automata to accept a set of strings of a language.
2. Design context free grammars to generate strings of context free languages.
3. Determine equivalence of languages accepted by Pushdown Automata and languages generated by context free grammars
4. Write the hierarchy of formal languages, grammars and machines.
5. Distinguish between computability and non-computability and Decidability and undecidability.

UNIT-I

Introduction: Finite state automata, Non-deterministic finite state automata, FA with €-transitions, Regular expressions, Applications of FA, Properties of regular sets, Pumping Lemma, Closure properties,

Myhill-Nerode Theorem, Minimization of FA.

UNIT-II

Context Free Grammars and Languages: Derivations, Parse-trees, Ambiguity in Grammars and Languages. Pushdown Automata–Definitions, The languages of PDA, Equivalence of PDAs and CFGs, Deterministic Pushdown Automata.

UNIT-III

Properties of CFLs: Normal forms for CFGs, Pumping Lemma, Closure properties, Deterministic Context Free Languages, Decision properties.

UNIT-IV

Turing Machines: Introduction, Computational Languages and Functions, Techniques for construction of Turing machines. Modifications of TM, TM as enumerator, Restricted TM.

UNIT-V

Undecidability: Recursive and Recursively enumerable languages, UTM and undecidable problem, Rice Theorem, Post's correspondence problem. Chomsky's Hierarchy–Regular grammars, Unrestricted grammar, CSL, Relationship between classes of languages.

Suggested Books:

1. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman,

Suggested Reference Books:

1. Zvi Kohavi, Switching and Finite Automata Theory, TMH, 1976

2. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia.

3. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.

4. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing.

5. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill.

Course Code	Course Title	Core/ Elective

***MC406HS**	**GENDER SENSITIZATION**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**0**

Course Objectives:

- To develop students' sensibility with regard to issues of gender in contemporary India.
- To provide a critical perspective on the socialization of men and women.
- To introduce students to information about some key biological aspects of genders.
- To help students reflect critically on gender violence.
- To expose students to more egalitarian interactions between men and women.

Course Outcomes:

- Students will have developed a better understanding of important issues related to gender in contemporary India.
- Students will be sensitized to basic dimensions of the biological, sociological, psychological and legal aspects of gender. This will be achieved through discussion of materials derived from research, facts, everyday life, literature and film.
- Students will attain a finer grasp of how gender discrimination works in our society and How to counter it.
- Students and professionals will be better equipped to work and live together as equals.
- Students will develop a sense of appreciation of women in all walks of life.

UNIT – I

Understanding Gender: Why Should We Study It? Socialization: Making Women, Making Men: Introduction-Preparing for Womanhood-Growing up male-First lessons in caste-Different Masculinities; Just Relationships: Being Together as Equals: Mary Kom and Onler-Love and acid just do not mix-Love Letters-Mothers and Fathers-Further reading: Rosa Parks-The brave heart.

UNIT – II

Gender and Biology: Missing Women: Sex selection and Its Consequences – Declining sex ratio. Demographic Consequences; Gender

Spectrum: Beyond the Binary – Two or many – Struggles with discrimination; Our Bodies, Our Health.

UNIT - III

Gender and Labour: Housework: the Invisible Labour: “My mother doesn’t work”- Share the Load"; Women's Work; Its Politics and Economics: Fact and fiction-Unrecognized and unaccounted work- Wages and conditions of work.

UNIT - IV

Issues of Violence: Sexual Harassment: Say No! : Sexual harassment – not eve-teasing-Coping with everyday harassment-“Chupulu”; Domestic Violence: Speaking Out: Is home a safe place? When women unite-Rebuilding lives-New forums for justice; thinking about Sexual Violence: Blaming the victim – “I fought for my life”. The caste face of violence

UNIT - V

Gender Studies: Knowledge - Through the Lens of Gender - Point of view - Gender and the structure of knowledge – Unacknowledged women artists of Telangana: Who’s History?

Questions for Historians and Others: Reclaiming a past-Writing other histories-Missing pages from modern Telangana history.

Suggested Readings:

1. A.Suneetha, Uma Bhrugubanda, DuggiralaVasanta, Rama Melkote, VasudhaNagarajAsmaRasheed, GoguShyamala, Deepa Srinivas and Susie Tharu, “Towards a World of Equals: A Bilingual Textbook on Gender" Telugu Akademi, Hyderabad, 1st Edition, 2015.
2. www.halfthesky.cgg.gov.in

Course	Course Title	Core/

Code							Elective
PC351CSM	**Data Structures And Algorithms Lab**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-		**-**	**-**	**2**	**25**	**50**	**1**

Objectives:

1. To develop skills to design and analyse simple linear and nonlinear data structures, suchas stacks, queues and lists and their applications.
2. To gain programming skills to implement sorting and searching algorithms
3. To Strengthen the ability to identify and apply the suitable data structures for the given real world problem
4. To Gain knowledge in practical applications of data structures

Outcomes:

After completing this course, the student will be able to:

1. Implement various data structures using arrays, linked lists
2. Develop ADT necessary for solving problems based on Stacks and Queues
3. Implement binary trees, general tree structures, advanced search trees, heaps, graphs.
4. Implement hash functions and handle collisions
5. Implement various kinds of sorting techniques and apply appropriate techniques for solving a given problem

List of Experiments (Using C programming Language):

1. Implementation of Stacks and Queues using Arrays.
2. Implementation of Circular Queue.
3. Implementation of Infix to Postfix Conversion,Postfix ExpressionEvaluation.
4. Implementation of SinglyLinkedList
5. Implementation of DoublyLinkedList.
6. Implementation of CircularLinkedList.
7. Implementation of Stacks,Queues using Linked Lists.
8. Implementation of BinarySearch and Hashing
9. Implementation of Operations on Binary Tree (Delete Entire Tree,Copy Entire Tree, Mirror Image, Level Order, Search for a Node etc.)
10. Implementation of Tree Traversals on Binary Trees.
11. Implementation of BinarySearch Tree. (Insertion, Deletion and Search operations)
12. Implementation of operations onAVL Trees.
13. Implementationof Traversal on Graphs.
14. Implementation of Prims and Kruskals Algorithm.
15. Implementation of Selection,Merge,Quick,Heap,and InsertionSort.

Course Code	Course Title	Core/ Elective
PC352ES	**Python Programming Lab**	**Core**

Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-		**-**	**-**	**2**	**25**	**50**	**1**

Course Objectives:

• To be able to introduce core programming basics and program design with functions using Python programming language.

• To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.

• To understand the high-performance programs designed to strengthen the practical expertise.

Course Outcomes: After learning the contents of this course the student is able to

• Explore Basics of Python programming

• Understand the concepts of Decision Making and Functions in Python

List of Programs:

1. Write the following classes with class variables, instance variable and illustration the self variable
   i) Robot (to greet the world)
   ii) ATM (to deposit and withdraw amount from ATM machine)
2. Make a class called Restaurant. The __init__() method for Restaurant should store two attributes: a restaurant_name and a cuisine_type. Make a method called describe_restaurant() that prints these two pieces of information, and a method called open_restaurant() that prints a message indicating that the restaurant is open. Create three different instances from the class, print the two attributes individually, and then call both methods for each instance.
3. Write a program to check whether the given number is Consecutive Four Sum Number or not. Consecutive Four Sum Number: A positive integer is called a `Consecutive Four Sum (CFS) number' if that number can be expressed as the sum of four consecutive positive integers.
4. Given a positive integer 'x' (with even number of digits in it), compute the difference between the sum of the digits occurring in the alternate positions (starting from the first position) and the sum of the digits occurring in the alternate positions, starting from the last rightmost position of 'x'.

5. Given a number A which contains only digits 0's and 1's. Your task is to make all digits the same by just flipping one digit (i.e. 0 to 1 or 1 to 0) only. If it is possible to make all the digits the same by just flipping one digit then print 'YES' else print 'NO'.
6. Write a program to create a list of tuples from a given list having a number and its cube in each tuple.
7. A professor calls out student IDs of students one by one while marking attendance. He notices that the number of students recorded in the attendance sheet is far more than the number of students who are actually present in the classes. Hence, he decides to use a chitti, the robot which can record the students' voices and keep track of which students have responded to attendance calls. At the end of each session, the robot outputs the student IDs of the students who have responded to attendance calls. With this information, the professor needs your help to find out which students were absent. Write a program which takes an integer array denoting the student IDs recorded by the robot and print the list of student IDs of the students which were absent in increasing order.

   Input Format: The first line of input contains a single integer n denoting the number of students. The second line contains n space-separated integers a1,a2....and denoting the student IDs recorded by the robot. The students have IDs from 1 to n, inclusive.

   Output Format: Print a single line containing the student IDs of the students which were absent, space-separated and in increasing order.
8. Let us assume paper as the plane and a letter as a curve on the plane, then each letter divides the plane into regions. For example letters "A", "D", "O", "P", "R" divide the plane into two regions so we say these letters each have one hole. Similarly, the letter "B" has two holes and letters such as "C", "E", "F", "K" have no holes. We say that the number of holes in the text is equal to the total number of holes in the letters of the text. Write a program to determine how many holes are in a given text.
9. Write a program to print each line of a file in reverse order. Also compute the number of characters, words and lines in a file.
10. Write a function named collatz() that has one parameter named number. If the number is even, then collatz() should print number // 2 and return this value. If number is odd, then collatz() should print and return 3 * number + 1. Then write a program that lets the user type in an integer and that keeps calling collatz() on that number until the function returns the value 1. (Amazingly enough, this sequence actually works for any integer—sooner or later, using this sequence, you'll arrive at 1! Even mathematicians aren't sure why. Your program is exploring what's called the Collatz sequence, sometimes called "the simplest impossible math problem.")

    The input and output of this program could look something like this:

    Input=

    Enter number: 3

    Output=

    10 5 16 8 4 2 1

    Input Validation Add try and except statements to the previous project to detect whether the user types in a non-integer string. Normally, the int() function will raise a ValueError error if it is passed a non-integer string, as in int('puppy'). In the except clause, print a message to the user saying they must enter an integer.

11. Say you have the boring task of finding every phone number and email address in a long web page or document. Write a program to search for the phone numbers and email addresses from a given text file and store them in a separate text file.
12. Using Python flask, develop a Government ”E-Seva E-Pass Portal” to support the COVID management team to provide travel passes for the citizens to travel from one location to another location.

TEXT BOOKS:

1. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning.

2. Think Python First Edition, by Allen B. Downey, Orielly publishing

REFERENCES:

1. Introduction to Computation and Programming Using Python. John V. Guttag, The MIT Press.

2. James Payne, Beginning Python using Python 2.6 and Python 3, Wrox publishing

3. Paul Gries, Practical Programming: An Introduction to Computer Science using Python 3,The Pragmatic Bookshelf, 2nd edition (4 Oct. 2013)

4. Charles Dierach, Introduction to Computer Science using Python

SCHEME OF INSTRUCTION & EXAMINATION
B.E. - IV SEMESTER

(AI&ML)

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination			Credits
			L	T	P/D	Contact Hrs/W	CIE	SEE	Duration in H	
Theory Courses										
1	HS104EG	Effective Technical Communication in English	3	-	-	3	30	70	3	3
2	PC401CSM	Design and Analysis of Algorithms	3	-	-	3	30	70	3	3
3	PC402CSM	Database Management Systems	3	-	-	3	30	70	3	3
4	PC403CSM	Software Engineering	3	-	-	3	30	70	3	3
5	PC404CSM	Introduction to Machine Learning	3	1	-	4	30	70	3	4
6	PC405CSM	Java Programming	3	-	-	3	30	70	3	3
Practical/ Laboratory Courses										
7	PC451CSM	Database Management Systems Lab	-	-	2	2	25	50	3	1
8	PC452CSM	Machine Learning Lab	-	-	2	2	25	50	3	1
9	PC453CSM	Java Programming Lab	-	-	2	2	25	50	3	1
			18	**1**	**6**	**25**	**255**	**570**	**-**	**22**

B.E IV SEMESTER Syllabus Computer Science and Engineering (AI & ML)

<table>
<tr><td>Course Code</td><td colspan="6">Course Title</td><td>Core/Elective</td></tr>
<tr><td><b>HS104EG</b></td><td colspan="6"><b>Effective Technical Communication in English</b></td><td><b>Core</b></td></tr>
<tr><td rowspan="2">Prerequisite</td><td colspan="4">Contact Hours per Week</td><td rowspan="2">CIE</td><td rowspan="2">SEE</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>D</td><td>P</td></tr>
<tr><td><b>-</b></td><td><b>3</b></td><td><b>-</b></td><td><b>-</b></td><td><b>-</b></td><td><b>30</b></td><td><b>70</b></td><td><b>3</b></td></tr>
</table>

Course Objectives

To expose the students to:

- Features of technical communication
- Types of professional correspondence
- Techniques of report writing
- Basics of manual writing
- Aspects of data transfer and presentations.

Course Outcomes

On successful completion of the course, the students would be able to:

1. Handle technical communication effectively
2. Use different types of professional correspondence
3. Use various techniques of report writing
4. Acquire adequate skills of manual writing
5. Enhance their skills of information transfer and presentations

UNIT-I

Definition and Features of Technical communication: Definition and features of technical communication (precision, relevance, format, style, use of visual aids), Differences between general writing and technical writing, Types of technical communication (oral and written)

UNIT-II

Technical Writing-I (Official correspondence): Emails, IOM, Business letters, Business proposals.

UNIT-III

Technical writing-II (Reports): Project report, Feasibility report, Progress report, Evaluation report.

UNIT-IV

Technical writing- III (Manuals): Types of manuals, User manual, Product manual, Operations manual.

UNIT-V

Information Transfer and Presentations: Non-verbal (bar diagram, flow chart, pie chart, tree diagram) to verbal (writing), Verbal (written) to non-verbal, Important aspects of oral and visual presentations.

Suggested readings:

1. Raman, Meenakshi & Sharma, Sangeeta. (2015). *Technical Communication: Principles and Practice*
(3rd ed.). New Delhi, OUP.

2. Rizvi, Ashraf, M. (2017). *Effective Technical Communication* (2nd ed.). New Delhi, Tata McGraw Hill Education.

3. Sharma, R. C., & Mohan, Krishna. (2017). *Business Correspondence and Report Writing: A Practical Approach to Business & Technical Communication* (4th ed.). New Delhi, Tata McGraw Hill Education.

4. Tyagi, Kavita &Misra, Padma. (2011). *Advanced Technical Communication*. New Delhi, PHI Learning.
5. Jungk, Dale. (2004). *Applied Writing for Technicians*. New York, McGraw-Hill Higher Education.

Course Code	Course Title	Core/ Elective
PC401CSM	**Design And Analysis Of Algorithms**	**Core**

Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
Data Structures	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives

- Analyze the asymptotic performance of algorithms
- Write rigorous correctness proofs for algorithms
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis
- Synthesize efficient algorithms in common engineering design situations.

Course Outcomes After learning the contents of this course the student is able to:

1. Analyze the performance of algorithms.
2. Choose appropriate algorithm design techniques for solving problems.
3. Apply the Dynamic programming to solve problems of the real world.
4. Solve problems Which use Graphs as their data structure.
5. Distinguishes NP class of problems.

UNIT I

Introduction: Algorithm definition, and specification, asymptotic analysis – best, average, and worst-case behavior; Performance measurements of Algorithms, Time and Space complexities, Analysis of recursive algorithms.

Basic Data Structures: Disjoint set operations, union and find algorithms, Dictionaries, Graphs, Trees.

UNIT II

Divide and Conquer: General method, Control abstraction, Merge sort, Quicksort – Worst, Best and average case. Binary search.

Greedy method: General method, applications- Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees, Single source shortest path problem.

UNIT III

Dynamic Programming: General Method, applications- All pairs shortest path problem, Optimal binary search trees, 0/1 knapsack problem, Reliability design, Traveling salesperson problem.

Backtracking: General method, Recursive backtracking algorithm, Iterative backtracking method. 8-Queen problem, Hamiltonian Cycle, 0/1 Knapsack Problem.

Branch and Bound: Control abstractions for Least Cost Search, Bounding, FIFO branch and bound, LC branch and bound, 0/1 Knapsack problem – LC branch and bound and FIFO branch and bound solution, Traveling salesperson problem.

UNIT IV

Graph Algorithms: Graph Traversals DFS, BFS, Transitive Closure, Directed Acyclic Graphs - Topological Ordering, Network Flow algorithms.

Tries: Standard Tries, Compressed Tries, Suffix Tries, Search Engine Indexing. External Searching and B-Trees: (a, b) Trees, B-Trees

UNIT V

Computational Complexity: Non Deterministic algorithms, The classes: P, NP, NP Complete, NP Hard, Satisfiability problem, Proofs for NP Complete Problems: Clique, Vertex Cover.

Parallel Algorithms: Introduction, models for parallel computing, computing with complete binary tree.

REFERENCES

1. E. Horowitz, S. Sahni, Fundamentals of Computer Algorithms.
2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006.
3. M T Goodrich, Roberto Tamassia, Algorithm Design, John Wiley, 2002.
4. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, Third Edition, PHI Learning Private Limited, 2012.

<table>
<tr><td>Course Code</td><td colspan="6">Course Title</td><td>Core/ Elective</td></tr>
<tr><td><b>PC402CSM</b></td><td colspan="6"><b>DATABASE MANAGEMENT SYSTEMS</b></td><td><b>Core</b></td></tr>
<tr><td rowspan="2">Prerequisite</td><td colspan="4">Contact Hours per Week</td><td rowspan="2">CIE</td><td rowspan="2">SEE</td><td rowspan="2">Credits</td></tr>
<tr><td>L</td><td>T</td><td>D</td><td>P</td></tr>
<tr><td><b>-</b></td><td><b>3</b></td><td><b>-</b></td><td><b>-</b></td><td><b>-</b></td><td><b>30</b></td><td><b>70</b></td><td><b>3</b></td></tr>
</table>

Course Objectives:

• **:** To learn

- The basic concepts and the applications of database systems.
- To master the basics of SQL and construct queries using SQL.
- Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

Course Outcomes:

After learning the contents of this course the student is able to

- Understand the basic concepts and the applications of database systems
- Design ER-models to represent simple database application scenarios.
- Master the basics of SQL and construct queries using SQL.
- Demonstrate creation and usage of Triggers, Views and Stored Procedures using SQL.
- Recognize and identify the use of normalization and functional dependency in database design.
- **Apply** and **relate** various advances SQL queries related to Transaction Processing & Locking using concept of Concurrency control.
- To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS

UNIT - I

Database System Applications: File Systems versus a DBMS**,** the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS, Advantages of DBMS

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT – II

Introduction to the Relational Model: Data Definition Language, Integrity constraint over relations, Types of Integrity Constraints Domain Constraint-String, character, Integer, date, Entity Integrity Constraint-, Primary Key, Referential Integrity Constraint-Foreign Key, Other Key Constraint – NULL,NOT NULL,CHECK and etc. querying relational data, logical data base design, introduction to views, destroying/altering tables and views.

UNIT - III

Introduction to SQL : Select Queries, Constraints: Data Manipulation Language –Insert ,Delete, Update, form of basic SQL query , UNION, INTERSECT, and EXCEPT, Nested Queries, Co-related Queries aggregation operators, NULL values, complex integrity constraints in SQL.

Concept of Joins: Join, Outer Join, Left Outer Join, Right Outer Join, Self Join

Schema Refinement : Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multi-valued dependencies, FOURTH normal form, FIFTH normal form.

UNIT - IV

Introduction to Transactions: Transaction Concept, Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, Lock Based Protocols, Timestamp Based Protocols, Validation- Based Protocols, Multiple Granularity, Recovery and Atomicity, Log–Based Recovery, Recovery with Concurrent Transactions. TCL Commands – Save point Commit and Rollback

UNIT – V

Overview of Triggers,Stored Procedures:triggers-Row level table level and active databases, Stored Procedures IN, OUT parameters, Execution of Stored Procedure from Java.

DBA – Introduction to DBA, Creating Users, Grant/Revoke Permissions on tables using DML Commands.

TEXT BOOKS:

1.DatabaseManagementSystems,RaghuramaKrishnan,JohannesGehrke,TataMcGrawHill,3rd Edition**(UNITS - I ,II,III).**

2. Database System Concepts, Silberschatz, Korth,McGrawHill,**(UNITS - IV ,V)**

REFERENCES:

1. Database Systems design, Implementation and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, *Pearson Education*
3. Introduction to Database Systems, C. J. Date, *Pearson Education*
4. Oracle for Professionals, The X Team, S.Shah and V.Shah, *SPD.*

Course Code	Course Title						Core/Elective
PC403CSM	**SOFTWARE ENGINEERING**						**Core**
Prerequisite	Contact hours per week				CIE	SEE	Credits
	L	T	D	P			
-	3	0	-	-	**30**	**70**	**3**

Course Objectives

- To introduce the basic concepts of software development- processes from defining a product to shipping and maintaining that product
- To impart knowledge on various phases, methodologies and practices of software development
- To understand importance of software modelling using UML.
- To understand the importance of testing in software development and study various testing strategies and software quality metrics.

Course Outcomes:

Students will be able to:

- Acquire knowledge about different software development processes and their usability in different problem domains.
- Understand the process of requirements collection, analysing, and modelling requirements for effective understanding and communication with stakeholders.
- Design and develop the architecture of real world problems towards developing a blueprint for implementation.
- Use the UML language to design various models during software development life cycle.
- Understand the concepts of software quality, testing and maintenance.

UNIT-I:

Introduction to Software Engineering: A generic view of Process: Software Engineering, Process Framework, CMM Process Patterns, and Process Assessment.

Process Models: Prescriptive Models, Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, the Unified Models, Personal and Team Process Models, Process Technology, Product and Process.

An Agile View of Process: Introduction to Agility and Agile Process, Agile Process Models.

UNIT-II:

Requirements Engineering: A Bridge to Design and Construction, Requirements Engineering Tasks, Initiating Requirements Engineering Process, Eliciting Requirements, Developing Use- Cases, Building the Analysis Model, Negotiating Requirements, Validating Requirements.

Building the Analysis Model: Requirements Analysis Modelling Approaches, Data Modelling Concepts, Object-Oriented Analysis, Scenario-based Modelling, Flow-oriented Modelling, Class-based Modelling, Creating a Behavioural Model.

UNIT-III:

Design Engineering: Design within the context of SE, Design Process and Design Quality, Design Concepts, The Design Model, Pattern-based Software Design.

Creating an Architectural Design: Software Architecture, Data Design, Architectural Styles and Patterns, Architectural Design, Assessing Alternative Architectural Designs, Mapping Data Flow into Software Architecture.

UNIT-IV:

Introduction to UML: Importance of Modelling, Principles of Modelling, Conceptual model of the UML, Basic Building Blocks of UML Basic Structural Modelling: Classes, Relationships, Common Mechanisms and Diagrams, Class Diagrams. Modelling techniques for Class Diagrams Basic Behavioural Modelling: Interactions, Interaction diagrams, Use cases, Use case Diagrams, Activity Diagrams, State chart diagrams Architectural Modelling: Component Diagrams and Deployment Diagrams.

UNIT-V:

Software Quality Assurance: Basic Elements, Tasks, Goals and Metrics, Formal Approaches, Statistical Software Quality Assurance, Software Reliability, ISO 9000 Quality Standards, SQA Plan.

Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for O-O Software, Validation Testing, System Testing, The Art of Debugging.

Testing Tactics: Software Testing Fundamentals, Black-box and White-box Testing, Basis Path Testing, Control Structure Testing.

Suggested Reading:

1. Roger S.Pressman, Software Engineering: A Practitioners Approach, Seventh Edition, McGrawHill, 2009.
2. Grady Booch, James Rumbaugh, Ivor Jacobson, “The Unified Modelling Language-User Guide (Covering UML 2.0)”, Second Edition, Pearson Education, India, 2007
3. Ali Behforoz and Frederic J.Hadson, Software Engineering Fundamentals, Oxford University Press, 1996.
4. Pankaj Jalote “An Integrated Approach to Software Engineering, Third Edition, Narosa Publishing house, 2008.

Course Code	Course Title						Core/ Elective
PC403CM	**INTRODUCTION TO MACHINE LEARNING**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**1**	**-**	**-**	**30**	**70**	**4**

Course Objectives:

- To introduce students to the basic concepts of Data Science and techniques of Machine Learning.
- To develop skills of using recent machine learning software for solving practical problems.
- To gain experience of doing independent study and research.

Course Outcomes: After learning the contents of this course the student is able to

1. Design and implement machine learning solutions of classification, regression problems.
2. Evaluate and interpret the results of the machine learning algorithms.
3. Evaluate exploratory data analysis and Data preparation and preprocessing on different datasets.
4. Calculate Statistical measurements of the given data.
5. Analyze and identify the best algorithm matches for a given dataset.

UNIT – I

Introduction: What is Machine Learning, Use Machine Learning, and Types of Machine Learning Systems: supervised, unsupervised, semi-supervised, Reinforcement Learning, Batch and Online Learning, Main Challenges of Machine Learning.

UNIT – II

Descriptive Statistics: Data representation, types of data- nominal, ordinal, interval and continuous, central tendency- calculating mean mode median, mean vs median, variability, variance, standard deviation, Mean Absolute Deviation using sample dataset, finding the percentile, interquartile range, Box Plot, Outlier, whisker, calculating correlation, covariance, causation.

Exploratory data analysis, Data preparation and preprocessing, Data visualization.

UNIT – III

Regression: Introduction to Regression analysis, measure of linear relationship, Regression with stats models, Determining coefficient, meaning and significance of coefficients, coefficient calculation with least square method, Types of regression, Simple Linear Regression, Using Multiple features, Polynomial Regression, Metrics for Regression: MSE, RMSE, MAE.

UNIT – IV

Classification: Classification problem, Probability based approach, Logistic Regression- log-odd, sigmoid transformation, Metrics: Confusion Matrix, Accuracy, Error Rate, Precision, Recall, ROC curve, F1 score, and introduction to gradient descent.

UNIT – V

Non Parametric & SVM classification: About Non parametric classification, Decision Trees: Entropy, Gain ratio, Information Gain, Splitting criteria,

Ensemble Method: Introduction to Random Forest, Accuracy measure & performance

Instance based learning- Introduction, KNN algorithm, Distance measures, model building, locally weighted regression, radial basis functions, SVM classifier, hyper-plane, slack variables, geometric transformation kernel trick, kernel transformation.

TEXT BOOKS / REFERENCES:

1. Booz, Allen, Hamilton, The Field Guide to Data Science
2. AurélienGéron, Hands-On Machine Learning with Scikit-Learn and TensorFlow,O’Reilly Media, 2017-03-10
3. Peter Harrington, Machine Learning in Action, Manning Publications
4. Python For Data Analysis by wes McKinny 2nd edition,O'REILLY publications.
5. Jason Brownlee data analysis for machine learning.

Course Code	Course Title	Core/ Elective
PC401CM	**JAVA PROGRAMMING**	**Core**

Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**3**	**-**	**-**	**-**	**30**	**70**	**3**

Course Objectives:

• **:** To learn

- The object oriented programming concepts,
- To understand object oriented programming concepts, and apply them in solving problems,
- To introduce the principles of inheritance and polymorphism; and demonstrate how they relate
- To design abstract classes and to introduce the implementation of packages and interfaces,
- To introduce the concepts of exception handling and multithreading.

Course Outcomes:

After learning the contents of this course the student is able to

- Use concepts of OOPs such as data abstraction, inheritance, polymorphism, encapsulation and method overloading principles in structuring computer applications for solving problems.
- Choose appropriate collections to solve programming problems.
- Utilize the concepts of I/O streams and exception handling in a given real time problem.
- Build java applications to utilize advanced mechanisms like multi-threading, database connectivity, etc.
- Apply the concepts and principles of the programming language to the real-world problems and solve the problems through project-based learning

UNIT- I

Object Oriented Programming: Principles, Benefits of Object Oriented Programming.

Introduction to Java: Java buzzwords, bytecode, Java Programming Fundamentals: Applet and Application program using simple java program, data types, variables, arrays, operators, expressions, control statements, type conversion and casting, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, introducing access control, static, final, nested and inner classes, exploring string class, using command-line arguments.

Inheritance: Inheritance concept, types of inheritance, Member access rules, use of super and final. Polymorphism - dynamic binding, method overriding, abstract classes and methods.

UNIT - II

Interfaces: Defining an interface, implementing interfaces, extending interface.

Packages: Defining, Creating and Accessing a Package, importing packages

Exception handling: Benefits of exception handling, classification, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, built in exceptions, creating own exception sub classes

Multithreading: Java Thread Model, The Main Thread, creating a Thread, creating multiple threads, using is Alive() and join(), thread priorities, synchronization, inter thread communication, deadlock

UNIT- III

Collections: Overview of Java Collection frame work, commonly used Collection classes – Array List, Linked List, Hash Set, Tree Set, Collection Interfaces – Collection, List, Set. Accessing Collection via iterate, working with Map. Legacy classes and interfaces – Vector, Hashtable, Stack, Dictionary, Enumeration interface.

Other Utility classes: String Tokenizer, Date, Calendar, Gregorian Calendar, Scanner

Java Input/Output: exploring java.io, Java I/O classes and interfaces, File, Stream classes, byte stream, character stream, serialization.

UNIT- IV

GUI Programming with java: The AWT class hierarchy, MVC architecture. Applet Revisited: Basics, architecture and skeleton, simple applet program.

Event Handling: Delegation Event Model, Event Classes, Source of Events, Event Listener Interfaces. Handling mouse and keyboard events, Adapter classes.

Database Programming using JDBC: Introduction to JDBC, JDBC Drivers & Architecture, CURD operation Using JDBC, Connecting to non-conventional Databases.

UNIT- V

Exploring Swing: JLabel, ImageIcon, JTextField, the Swing buttons, JTabbedpane, JScrollPane, JList, JComboBox.

Servlet: Life cycle, using tomcat, simple servlet, servlet API, javax.servlet package, reading servlet parameters, javax.servlet.http package, handling HTTP requests and responses

Suggested Readings:

1. Herbert Scheldt, “The Complete Reference Java, 7th Edition, Tata McGraw Hill, 2006.
2. James M Slack, Programming and Problem Solving with JAVA, Thomson Learning, 2002.
3. C Thomas Wu, An Introduction to Object Oriented Programming with Java 5th Edition, McGraw Hill Publishing, 2010.
4. H. M. Dietel and P. J. Dietel, Java How to Program, Sixth Edition, Pearson Education /.

Course Code	Course Title						Core/ Elective
PC451CSM	**DATABASE MANAGEMENT SYSTEMS LAB**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	**2**	**25**	**50**	**1**

Course Outcomes:

- Design database schema for a given application and apply normalization
- Gather skills in using SQL commands for data definition and data manipulation.
- Develop solutions for database applications using procedures, cursors and triggers

LIST OF EXPERIMENTS

Scenario 1:

Product-Sales database **: SouthWind**

Southwind database is a sample database used by Organization. The database contains the sales data for SouthWind Traders, a foods export-import company. Using this schema to demonstrate how customers can choose and order products, how orders are placed and how those products get delivered to the customer.

Products: This Entity will have all the product details where suppliers will supply products based on customers demand.

Supplies: This Entity will supply the products demanded by the customers. Shippers: This Entity will take the orders from suppliers and deliver to customers. Employees : Employees will monitor the orders placed by customers.

Invoices: This Entity will take care of the billing process based on customer order. Etc..identify some more entities and find out the relationship between them.

A product-sales the above process involves many steps like

1. Analyzing the problem and identifying the Entities and Relationships,
2. E-R Model
3. Relational Model
4. Normalization
5. Creating the database
6. Querying.

Experiment 1: E-R Model

Analyze and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc.

Identify the primary keys for all the entities. Identify the other keys like Foreign Key and constraints like NULL, NOT NULL, CHECK etc.

Example to create for **products**, **customers**, **suppliers**, **orders**, , **employees, order details, categories**, among others.

Students should submit E-R diagrams using the above tables.

Experiment 2: Installation & DDL

Installation of Mysql and practicing DDL commands.

Creating databases, How to create tables, altering the database or tables, dropping tables and databases if not required. You will also try truncate, rename commands etc.

Data Definition Language (DDL) : create , alter, drop.

Experiment 3: DML

Data Manipulation Language Commands (DML) commands are used to for managing data within schema objects.

Exercising the commands using **DML :** insert, delete, update on the following tables : products, customers, suppliers, orders, , employees, order details, categories.

- INSERT – insert data into a table.
- UPDATE – updates existing data within a table.
- DELETE – deletes single or all records from a table.

Data Query Language – Select

Populate all the tables designed in experiment : 2 with appropriate data.

Experiment 4: Querying

Practice queries on **Aggregate functions** like count, max , min ,avg ,sum Practice queries like nested queries/co-related queries using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, groupby ,having etc.

Joins: Join , Left Outer Join, Right Outer Join, Self Join

Experiment 5 : Querying(continued…)

Some example to practice the queries:

- Display all the order details given to a customer.
- Display all the products.
- Get the highest sold product from given supplier ID
- List all products grouped by category
- List the products, whose product unit price is greater than all the products on average.
- List Details of order and customer of each order
- List the products which were sold in year 1997
- Display the total amount for each order
- Display Order Details for given an order ID

Order Details: product name and unit price for

given order ID Exercising Simple to complex Queries

using joins, nested and correlated queries.

Experiment 6 : Stored Procedures :

Create a stored procedure, Alter and Drop a procedure, IN, OUT, IN & OUT parameters

- Create a Procedure to display order details of given customer ID like ordered, order Date , Required Date, Shipped Date
- Create a procedure to accept a customer ID and display the customer order history(product name and how much quantity ordered for that particular product)
  Ex: product name, Total quantity he/she ordered.

- Create a procedure to display Ten Most Expensive Products Columns should be displayed Product name & Unit price

Experiment 7: Views

Create a view to display the current product list which are available(not discontinued)

Create a view to display the products by category

Display product name, quantity Per Unit, units In Stock, Discontinued

Create a view as “Invoices” to display all the information from order, customer, shipper for each “Order Details”

Experiment 8: Triggers

Demonstrate Create Trigger, Alter Trigger, Drop Trigger, Row Level , Table Level triggers, Before Insert ,After Insert, Before Update, After Update, Before Delete, After Delete

Experiment 9 :

Demonstrate the role of DBA using DCL commands

TEXT BOOKS:

1. Raghurama Krishnan, Johannes Gehrke, “Database Management Systems”, Tata McGraw Hill, 3rd Edition, 2008.
2. Silberschatz, Korth, “Database System Concepts”, McGraw Hill, V edition, 2005.

REFERENCES BOOKS:

1. Rick F. Vander Lans, “Introduction to SQL”, Pearson education, 2007.
2. B. Rosenzweig and E. Silvestrova, “Oracle PL/SQL”, Pearson education, 2004.
3. Dr. P. S. Deshpande, “SQL & PL/SQL for Oracle 10g”, Black Book, Dream Tech, 2006.
4. M. Mc Laughlin, “Oracle Database 11g PL/SQL Programming”, TMH, 2017

Course Code	Course Title						Core/ Elective
PC452CSM	**MACHINE LEARNING LAB**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	**2**	**25**	**50**	**1**

Course Objectives: The objective of this lab is to get an overview of the various machine learning techniques and can able to demonstrate them using python.

- To introduce students to the basic concepts of Data Science and techniques of Machine Learning.
- To develop skills of using recent machine learning software for solving practical problems.
- To gain experience of doing independent study and research.

Course Outcomes:
After the completion of the course the student can able to:

- After learning the contents of this paper the student must be able to design and implement machine learning solutions to classification, regression problems.
- Understand complexity of Machine Learning algorithms and their limitations
- Be capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
- Be capable of performing experiments in Machine Learning using real-world data.·
- Able to evaluate and interpret the results of the algorithms.

LIST OF EXPERIMENTS

1. Write a program to demonstrate the following

a) Operation of data types in Python.
b) Different Arithmetic Operations on numbers in Python.
c) Create, concatenate and print a string and access substring from a given string.
d) Append, and remove lists in python.
e) Demonstrate working with tuples in python.
f) Demonstrate working with dictionaries in python.

2. Using python write a NumPy program to compute the

a) Expected Value
b) Mean
c) Standard deviation
d) Variance
e) Covariance
f) Covariance Matrix of two given arrays.

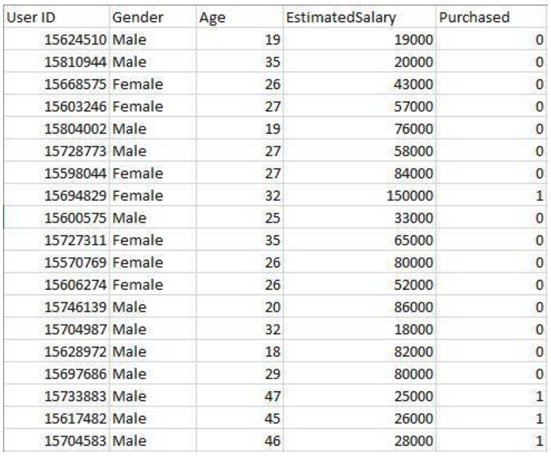
3. For a given set of training data examples stored in a .CSV file, demonstrate Data Preprocessing in Machine learning with the following steps

a) Getting the dataset.
b) Importing libraries.
c) Importing datasets.
d) Finding Missing Data.
e) Encoding Categorical Data.
f) Splitting dataset into training and test set.
g) Feature scaling.

4. Build a linear regression model using python for a particular data set by

a) Splitting Training data and Test data.
b) Evaluate the model (intercept and slope).
c) Visualize the training set and testing set
d) predicting the test set result
e) compare actual output values with predicted values

5. The dataset contains information of users from a company's database. It contains information about UserID, Gender, Age, EstimatedSalary, and Purchased. Use this dataset for predicting that a user will purchase the company's newly launched product or not by Logistic Regression model.



User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0
15570769	Female	26	80000	0
15606274	Female	26	52000	0
15746139	Male	20	86000	0
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1
15617482	Male	45	26000	1
15704583	Male	46	28000	1

6. a) The probability that it is Friday and that a student is absent is 3 %. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is the probability that a student is absent given that today is Friday? Apply Bayes rule in python to get the result. (Ans: 15%)

b) Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the detection of diabetic patients using standard diabetic Disease Data Set. use Python ML library classes.

8. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

9. Implement k-nearest neighbor's classification to classify the iris data set using python.

10. Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3 centroids)

VAR1	**VAR2**	**CLASS**
1.713	1.586	0
0.180	1.786	1
0.353	1.240	1
0.940	1.566	0
1.486	0.759	1
1.266	1.106	0
1.540	0.419	1
0.459	1.799	1
0.773	0.186	1

11. Evaluate the metrics for all types of machine learning algorithms using sample data.

12. Implement an algorithm to demonstrate the significance of SVM.

TEXT BOOKS / REFERENCES:

1. Booz, Allen, Hamilton, The Field Guide to Data Science
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow,O'Reilly Media, 2017-03-10
3. Peter Harrington, Machine Learning in Action, Manning Publications

course Code	Course Title						Core/Elective
PC453CSM	**JAVA Programming Lab**						**Core**
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	**-**	**-**	**-**	**2**	**25**	**50**	**1**

Course Objectives

- To build software development skills using java programming for real world applications.
- To implement frontend and backend of an application
- To implement classical problems using java programming.

Course Outcomes

After completing this course, the student will be able to:

1. Develop Java applications using the concepts of Inheritance, interfaces, packages, access control specifiers.
2. Implement the concepts of Exception Handling in java Applications.
3. Read and write data using different Java I/O streams.
4. Create graphical user interfaces and Applets by applying the knowledge of Event Handling.
5. Create robust applications using Java standard class libraries and retrieve data from a database with JDBC.
6. Ability to solve real-world problems by designing user friendly GUI with befitting backend through the APIs of Java.

List of Experiments

1) Write a Java program to illustrate the concept of class with method overloading
2) Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java. util)
3) Write a Java program to illustrate the concept of Single level and Multi level Inheritance.
4) Write a Java program to demonstrate the Interfaces & Abstract Classes.
5) Write a Java program to implement the concept of exception handling.
6) Write a Java program to illustrate the concept of threading using Thread Class and runnable Interface.
7) Write a Java program to illustrate the concept of Thread synchronization.
8) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
9) Write a Java program to illustrate collection classes like Array List, Linked List, Tree map and Hash map.
10) Write a Java program to illustrate Legacy classes like Vector, Hashtable, Dictionary & Enumeration interface
11) Write a Java program to implement iteration over Collection using Iterator interface and List Iterator interface
12) Write a Java program that reads a file name from the user, and then displays

information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

13) Write a Java program to illustrate the concept of I/O Streams
14) Write a Java program to implement serialization concept
15) Write a Java applet program to implement Colour and Graphics class
16) Write a Java applet program for handling mouse & key events
17) Write a Java applet program to implement Adapter classes
18) Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result.
19) Write an example for JDBC prepared statement with Result Set
20) Program to get primary key value (auto-generated keys) from inserted queries using JDBC
21) Program to create a simple JList
22) java Program to create a simple checkbox using JCheckBox
23) Program to create a checkbox and ItemListener to it.
24) 1. Write Servlet application to print current date & time
    2. Html & Servlet Communication
    3. Auto refresh a page
    4. Demonstrate session tracking
    5. Select record from database
    6. Application for login page
    7. Insert record into database
    8. Count the visits on web page
    9. Insert teacher record in Database