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

						eme of uction		neme of ninatio		s
S. No.	Course Code	Course Title	L	т	P / D	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	Credits
		Theo	ry C	ours	es					
1	BS207MT	Mathematics – III (Probability & Statistics)	3	-	-	3	30	70	3	3
2	HS105CSM	Finance and Accounting	3	-	١-	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/ La	bora	tory	Co	urses				
8	PC 351 CSM	Data Structures and Algorithms Lab	-,	-	2	2	25	50	3	1
9	PC 352 CSM	Python Programming Lab	-1	-	2	2	25	50	3	1
			21	-	4	25	260	590	-	20

PC: Professional Course
HS: Humanities and social Science
L: Lecture T: Tutorial
CIE: Continuous Internal Evaluation,

PE: Professional Elective,
MC: Mandatory Course
P: Practical D: Drawing
SEE: Semester End Examination (Univ. Exam)

## Note:

- Each contact hour is a Clock Hour
   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)



38 / 80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

Course Code			Co	;		Core/Elective	
BS207MT	M	<b>fathema</b> t	tics – III	(Probabi	ility & Statis	tics)	Core
	C	ontact Ho	urs per W	/eek			
Prerequisite	L	T	D	P	CIE	SEE	Credits
-	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.
- 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







39/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

## Suggested Readings:

- 1. R.K.Jain & Iyengar, "Advanced Engineering Mathematics", Narosa Publications.
- B.S. Grewal, "Higher Engineering Mathematics", Khanna Publishers, 2000.
   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.
- S.C.Gupta & V.K.Kapoor, "Fundamentals of Mathematical Statistics", S.Chand Pub.
   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	Course Title	5.		Core/			
Code				Elective			
HS105CSM	Finance and Ac	Finance and Accounting					
	Contact Hours per Week						





40 / 80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

	Prerequisite	L	Т	D	P	CIE	SEE	Credits
-	•	3	-	-	-	30	70	3

### **Course Objectives**

The course will introduce the students

- $\hfill \Box$  To provide basic understanding of Financial and Accounting aspects of a business unit
- □ To provide understanding of the accounting aspects of business
- $\hfill \Box$  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







41/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

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

Course Code		Core/ Elective				
PC301CSM	D	Core				
	Contact 1	Hours po	er Week			
Prerequisite	LT	D	P	CIE	SEE	Credits





42 / 80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

		-	3	-	-		30	70	3
--	--	---	---	---	---	--	----	----	---

### 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.
- 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 of an algorithm- Time and Space Complexity, Asymptotic Notations.

 ${\bf Arrays:} Arrays-ADT, Polynomials, Sparse matrices, Strings-ADT, Pattern Matching. \\$ 

## 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 for Circularly linked lists, Equivalence Classes, Sparsematrices, Doubly Linked Lists.

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

UNIT – IV



43/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

**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:

 $1. \\ Horowitz~E,~Sahni~S~and~Susan~Anderson-Freed,~Fundamentals~of Datastructures in C, \\ 2^{nd} Edition (2008), Universities Press$ 

### ReferenceBooks:

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

Course	Cou	rse title			Core/
Code		Elective			
ES302EC	Digita	Core			
	Contact Hours pe	er Week			
Prerequisite	L T D	P	CIE	SEE	Credits





44/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

-	3	-	-	-	30	70	3
Course Objectives							

#### - - -

- $\hfill \Box$  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

- Understand the design process of digital hardware, use Boolean algebra to minimize the logical expressions and optimize the implementation of logical functions.
- Understand the number representation and design combinational circuits like adders, MUX etc.
- 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.
- 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 - I

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.





45 / 80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

#### 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:

- Morris Mano and Michael D Clletti, Digital Design, Pearson, fourth edition,2008
   Zvi Kohavi, Switching and Finite Automata Theory, 3<sup>rd</sup> ed., Cambridge University Press-New Delhi, 2011.
- 3. R. P Jain, Modern Digital Electronics, 4th ed., McGraw Hill Education (India) Private Limited, 2003
- 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.





46 / 80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

Course		Course title									
Code											
ES303CSM		PYTHON PROGRAMMING									
		Contact	Hours pe	er Week							
Prerequisite	L	T	D	P	CIE	SEE	Credits				
•	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.



47 / 80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

#### 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:

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





48 / 80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

## 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.







49/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

Course Code		Course title									
PC404CSM	A	AUTOMATA LANGUAGES AND COMPUTATION									
		Contact 1	Hours per	Week							
Prerequisite	L	T	D	P	CIE	SEE	Credits				
-	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

- 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.
- 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.





50/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

### **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





51/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

Course		Course Title								
Code										
*MC406HS		GENDER SENSITIZATION								
		Contact	Hours p	er Week						
Prerequisite	L	T	D	P	CIE	SEE	Credits			
-	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





52/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

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:**

- 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, 1<sup>st</sup> Edition, 2015.
- 2. www.halfthesky.cgg.gov.in





53/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

Course			Course 7	litle .		Core/
Code						Elective
PC351CSM	Da	ta Structu	res And A	lgorithms Lab	i	Core
Prerequisite		ntact Ho eek	ours per	CIE	SEE	Credits
	I T	D	P			
	-	-	2	25	50	1

## Objectives:

- 1. To develop skills to design and analyse simple linear and nonlinear data structures, such a 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.
- Implementation of Circular Queue.
- 3. Implementation of Infix to Postfix Conversion, Postfix Expression Evaluation.
- 4. Implementation of SinglyLinkedList
- 5. Implementation of DoublyLinkedList.
- Implementation of CircularLinkedList.
- 7. Implementation of Stacks, Queues using Linked Lists.
- 8. Implementation of BinarySearch and Hashing
- 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 on AVL Trees.





54/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

- 13. Implementation of Traversal on Graphs.
- 14. Implementation of Prims and Kruskals Algorithm.
- 15. Implementation of Selection, Merge, Quick, Heap, and Insertion Sort.





55/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

Course							Core/
Code							Elective
PC352ES	Python Programming Lab						Core
	Contact Hours per Week						
Prerequisite	L	T	D	P	CIE	SEE	Credits
-		-	-	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:

- 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'.





56 / 80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

- 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'.
- 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.
- 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=

105168421

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.





57/80

Faculty of Engineering (OU)

With effect from the academic year 2020-2021

- 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
- 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



