SEMESTER -3

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITT201	DATA STRUCTURES	PCC	3	1	0	4

Preamble: The syllabus is prepared with the view of preparing the Engineering Graduates capable of understanding essential concept of data structures, designing algorithms to perform operations involving these data structures and to choose appropriate data structures to solve real world problems.

Prerequisite: programming in C

Course Outcomes: After the completion of the course the student will be able to

CO_No.	Course Outcome(CO)	Bloom's Category
CO 1	**Summarize** different categories of data structures	Level 2 : Understand
CO 2	**Identify different** parameters to analyze the performance of an algorithm.	Level 3 : Apply
CO 3	**Explain** the significance of dynamic memory management Techniques.	Level 2 : Understand
CO 4	**Design** algorithms to perform operations with Linear and Nonlinear data structures	Level 3 : Apply
CO 5	**Illustrate** various technique to for searching, Sorting and hashing	Level 2 : Understand
CO 6	**Choose** appropriate data structures to solve real world problems efficiently.	Level 3 : Apply

Mapping of course outcomes with program outcomes

COs	PROGRAMME OUTCOMES (PO)											
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	2	2	-	-	1	-	-	-	-	-	1
CO 2	3	2	2	2	1	1	-	-	-	-	-	1
CO 3	3	3	3	2	1	1	-	-	-	-	-	1
CO 4	3	3	3	2	1	1	-	-	-	-	-	1
CO 5	3	2	2	1	1	-	-	-	-	-	-	1
CO 6	3	3	3	2	1	1	-	-	-	-	-	1

3/2/1: high/medium/low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	10
Understand	20	20	20
Apply	20	20	70
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Differentiate CDT and ADT
2. Classify classical datstructures
3. Compare array and linked list
4. Represent single and double dimensional array
5. Describe any three applications of array

Course Outcome 2 (CO2):

1. Identify the needs of algorithm analysis
2. Select two parameters to do the performance analysis of an algorithm
3. Identify 3 possible cases of time complexity

Course Outcome 3 (CO3):

1. Classify linked list
2. Illustrate different operations on singly, doubly and circular linked list
3. Represent linked list in memory (static and dynamic).

4. Summarize different dynamic memory management schemes.
5. Demonstrate the first fit , best fit , worst fit and next fit allocation of given process queue and free list

Process queue	85 K	35 K	70 K	100 K

	80K	130K	90 K	40 K
Free list				

Course Outcome 4 (CO4):

1. Design the algorithms to perform PUSH()and POP() and STATUS() operations on stack using array and linked list
2. Apply Stack data structure in infix to postfix conversion, expression evaluation, recursion and delimiter matching.
3. Design the algorithms for pre-order, in-order and post-order traversal on binary trees
4. Develop the algorithms for ENQUEUE(), and DEQUEUE() operations on queue data structures
5. Construct the algorithms for graph traversal(BFS, DFS)

Course Outcome 5 (CO5):

1. Classify Sorting Techniques (internal and external, n2 and n logn)
2. Compare Linear and binary search
3. Illustrate bubble, selection and insertion sort.
4. Describe quick and merge sort
5. Represent the following values in the given order in a hash table (Size of hash table is 7 and hash function used is h(k)=k mod 7) for each of the scenario.
   19, 26, 13, 48, 17
   a) When collisions are handled by linear probing
   b) When collisions are handled by double using second hash function h=5-(k mod 5)

Course Outcome 6 (CO6):

1. Develop an application program which is to be used in Ticket counter, where First person gets ticket first and go out first, using suitable data structure.
2. Make use of suitable data structure to store the details of pass percentage of the college in chronological order of years (oldest to newest) and retrieve the information in reverse chronological order of years, using suitable data structure.

Model Question Paper

PART A **(10*3=30)**

(Each Question carries 3 Marks)

1. Classify Sorting Techniques
2. Differentiate CDT and ADT
3. Classify linked list
4. Compare First fit and Next Fit Algorithms
5. Design the algorithms to perform PUSH()and POP()
6. Summarize operations Queue data structure
7. List out the features of binary tree
8. Explain binary sorting
9. Define hashing
10. Illustrate separate chaining with an example

PART B **(5*14=70)**

11. Classify classical data structures

OR

12. Illustrate Quick sort with the help of an example

13. Illustrate different operations on singly linked list

OR

14. Demonstrate the first fit , best fit , worst fit and next fit allocation of given process queue and free list

Process queue	85 K	35 K	70 K	100 K

	80K	130K	90 K	40 K
Free list				

15. Apply Stack data structure in infix to postfix conversion

OR

16. Develop the algorithms for ENQUEUE(), and DEQUEUE() operations on queue data structures

17. Construct the algorithms for graph traversal(BFS, DFS)

OR

18. Explain 3 types of binary tree traversal

19. Explain Any 3 types of hash functions

OR

20. Represent the following values in the given order in a hash table (Size of hash table is 7 and hash function used is h(k)=k mod 7) for each of the scenario.

    19, 26, 13, 48, 17

    a) When collisions are handled by linear probing

    b) When collisions are handled by double using second hash function h=5-(k mod 5)

Syllabus

Module 1: Introduction to data structures (9 Hours)
Data Structures-Introduction and Overview- Arrays, Algorithm/Program Development, , Searching and Sorting.
Module 2: Linked lists (10 Hours)
Linked lists, singly linked list, Doubly linked list, Circular linked list, Applications of linked list, Dynamic Memory management.
Module 3 : Stacks and Queues (9 Hours)
Stack, Applications of stacks, Queues, Types of queues
Module 4 : Trees and graphs (10 Hours)
Trees, Binary Tree Traversals, Binary tree Applications, Graph, and Graph Applications.
Module 5 : Hash Table (7 Hours)
Hash Tables, Different Hash Functions, Collision Resolution Techniques, closed hashing and Open Hashing (Separate Chaining).

Text Books

T1. Samanta D., Classic Data Structures, Prentice Hall India, 2/e, 2009.

T2. Ellis horowitz, Sartaj Sahni, Fundamentals of Data structures, Galgotia Booksource

Reference Books

R1. Horwitz E., S. Sahni and S. Anderson, Fundamentals of Data Structures in C, University Press (India), 2008.

R2. Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication,1983**.**

R3. Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill, 1995.

R4. Peter Brass, Advanced Data Structures, Cambridge University Press, 2008

R5. Lipschuts S., Theory and Problems of Data Structures, Schaum's Series, 1986.

R6. Wirth N., Algorithms + Data Structures = Programs, Prentice Hall, 2004.

R7. Hugges J. K. and J. I. Michtm, A Structured Approach to Programming, PHI, 1987.

R8. Martin Barrett, Clifford Wagner, And Unix: Tools For Software Design, John Wiley, 2008 reprint.

Course Contents and Lecture Schedule

	Module 1: Introduction to data structures	**9hrs**
1.1	**Data Structures-Introduction and Overview**: Definitions, Concept of data structure, classifications of data structure- ADT and CDT- Linear and nonlinear.	1
1.2	**Arrays:** definition, Representation of Single/Two dimensional arrays, Applications of array – searching –Sorting - Sparse Matrix- conversion of sparse matrix into 3 tuple form.	2
1.3	**Algorithm/Program Development:** Analysis of algorithms. Space Complexity, Time Complexity - Best case, worst case, average case. **Searching** : linear and binary search – Complexity Analysis (Detailed analysis is not required)	2
1.4	**Sorting**: classifications- Internal sorting – External sorting , N^2 Sorting : Selection, bubble and insertion- Complexity analysis (Detailed analysis is not required)	2
1.5	N log $_n$ Sorting : Quick Sort and Merge Sort (Recursive Algorithms)-Complexity Analysis (Detailed analysis is not required)	2
	Module 2: Linked lists	**10 hrs**
2.1	**Linked lists**: static and dynamic representation, Classification -Singly linked list- Doubly linked list- Circular linked list, array and linked list. **Singly linked list**: Operations on Singly linked list- Traversal-Insertion-deletion, copying -searching - Merging.	2
2.2	**Doubly linked list**: Operations on doubly linked list- Insertion-deletion.	2
2.3	**Circular Linked list** : Operations on circular linked list-Insertion and deletion	2
2.4	**Applications of linked list**: Polynomial representation and manipulation (addition)- Dynamic Memory management.	2
2.5	**Dynamic Memory management**: Fixed sized and variable sized memory allocation and de-allocation. First-fit, best-fit and worst-fit allocation schemes and problems.	2
	Module 3: Stacks and Queues	**9 hrs**
3.1	**Stack:** Definition, Schematic Diagram of stack, Array and Liked list representation of stack , operations on stack using array and linked list (PUSH(),POP(),STATUS()) .	2
3.2	**Applications of stacks**: Infix to postfix conversion- post fix evaluation, string reversal, delimiter matching.	3
3.3	**Queues**: Definition, Schematic Diagram of queue, Array and Liked list representation of queue , operations on queue using array and linked list (EQUEUE(),DEQUEUE(),STATUS()) .	2
3.4	**Types of queue** : circular queue-priority queue- doubly ended queue	2

	Module 4: Trees and graphs	**10 hrs**
4.1	**Trees:** Basic terminologies, Binary Trees, Properties of binary trees, linear and linked representations, Complete and full Binary Tree.	2
4.2	**Binary Tree Traversals**: Preorder -In order and post order (Recursive, non-recursive)-problems	1
4.3	**Binary tree Applications:** Expression tree creation, heap trees (concepts), Binary search tree – creation, insertion and deletion and search operations	3
4.4	**Graph**: Terminologies, set representations, linked/adjacency list representation, Adjacency matrix linear representation **Graph traversal**: Breadth First Search (BFS), Depth First Search (DFS) - related problems.	2
4.5	**Graph Applications:** Shortest Path Problem-Dijkstras Algorithm	2
	Module 5: Hash Table	**7 hrs**
5.1	**Hash Tables**-Hash Functions- Features of hash function.	1
5.2	**Different Hash Functions:** Division Method- Multiplication Method - Mid Square Method, Folding Method- related problems.	2
5.3	**Collision Resolution Techniques:** Closed hashing (Linear probing) and Open Hashing (Separate Chaining) . Closed hashing(Linear probing) -Drawbacks- Remedies - Radom Probing – Double hashing/Re-hashing –Quadratic Probing, problems to create hash tables using linear probing and Random probing, double hash and quadratic probing .	3
5.4	Open Hashing (Separate Chaining)	1

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITT203	DIGITAL SYSTEM DESIGN	PCC	3	1	0	4

Preamble: The syllabus is prepared with the view of preparing the Engineering Graduates capable of understanding the basic digital logic design and implementation. All students of computing should acquire some understanding and appreciation of a computer system's functional components, their characteristics, their performance, and their interactions.

Prerequisite: NIL

Course Outcomes: After the completion of the course, the student will be able

CO No.	Course Outcome(CO)	Bloom's Category
CO 1	To perform base conversion and arithmetic operations in various number systems.	Apply
CO 2	To design digital circuits using simplified Boolean functions	Create
CO 3	To develop simple design of combinational circuits	Apply
CO 4	To develop simple design of sequential circuits	Apply
CO 5	To interpret the generalization of synchronous and asynchronous sequential circuits	Understand

Mapping of course outcomes with program outcomes

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3			3	-	-	-	-	-	-	-	2
CO 2	3	3	3	3	-	-	-	-	-	-	-	2
CO 3	3	3	3	3	-	2	-	1	-	-	-	2
CO 4	3			3	-	-	-	-	-	-	-	2
CO 5	3	3	3	3	2	2	-	1	-	-	-	2

3/2/1: high/medium/low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Marks)	Test 2 (Marks)	
Remember	10	5	20
Understand	15	10	20
Apply	10	10	25
Analyse	10	10	15
Evaluate	5	10	10
Create		5	10

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	4 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1): To understand the basic concepts of Number systems

1. Convert the given number from decimal number system to binary, octal, and hexadecimal number system.
2. Perform Arithmetic operations on different number system.
3. Represent the different coding schemes.

Course Outcome 2 (CO2): To design digital circuits using simplified Boolean functions

1. Simplify the given expression using Postulates of Boolean algebra.

2. Convert a given expression to standard and canonical forms.
3. Simplify the given expression using Karnaugh Map or Quine –McClusky minimization technique.

Course Outcome 3 (CO3): To analyze and design combinational circuits

1. Analyse a given circuit and explain the results obtained by the circuit.
2. Design a Ccarry look ahead adder.
3. Design a four-bit magnitude comparator.

Course Outcome 4 (CO4): To understand the basics of sequential circuits

1. Understand the functioning of Latches and Flip Flops.
2. Design Master-Slave Flip Flops.
3. Understand the basics of different types of Flip Flops.

Course Outcome 5(CO5): To analyze and design synchronous and asynchronous sequential circuits

1. Analyse a given circuit and explain the results obtained by the circuit.
2. Implement a serial adder using a shift register.
3. Design and construct a 4-bit ring counter with only one flip-flop is clear at any particular time and all other flip-flops are set. Give its timing diagram.
4. Using an example, show the Race-Free State Assignment in an asynchronous sequential circuit.

Model Question Paper

Course Code: ITT203
Course Name: DIGITAL SYSTEM DESIGN

Max. Marks: 100 Duration: 3 Hours

PART A
Answer all questions, each carries 3 marks.

1. Convert $(76.75)_{10}$ to binary, octal and hexadecimal.
2. Determine the base of the numbers in the operation; $\mathbf{58/4}$= 15.
3. Simplify the Boolean expression to minimum number of literals.
   $F=\bar{B}C+A\bar{B}+ABC+ABC\bar{D}+\bar{A}\bar{B}CD+A\bar{B}CD$
4. Find the complement of the Boolean function $F=\bar{A}+AB\bar{C}$.And prove that $F+\bar{F}=1$and $F.\bar{F}=0$.
5. Design a 4-to-2 line priority encoder.

6. Explain the difference between a latch and a flip-flop.

7. Give the characteristics equations for D, JK and T flip-flops.

8. Discuss in detail about Race condition.

9. With a neat diagram, discuss about SISO.

10. Design a 4-bit ring counter.

PART B
Answer all questions, each carries 14 marks.

11. a) Using Booth algorithm, perform multiplication of (-14) and (-7). (5)

b) Represent the unsigned decimal numbers 572.36 and 382.71 in BCD. Show the necessary steps to form their sum and difference. (9)

OR

12. a) (i) Find the decimal equivalent of $(A40F)_{16}$
(ii) Find the 16's complement of $(A40F)_{16}$
(iii) Convert to binary $(A40F)_{16}$
(iv) Finds the 2's complement of the result in (iii) (8)

b) Perform addition, subtraction, multiplication, and division of the following binary numbers without converting them to decimal : 1000110 and 110. (6)

13. a) For the Boolean function

$$F = w'xy' + xy'z + x'y'z + w'xy + wx'y + wxy$$

(i) Draw the logic diagram, using the original Boolean expression.
(ii) Simplify the Boolean algebra to a minimum number of literals.
(iii) Obtain the truth table of the function from the simplified expression and show that it is the same as the original Boolean expression. (9)

b) Prove that $A+\bar{A}B = A + B$ using Boolean postulates. (5)

OR

14. a) Simplify the following functions using Quine- McClusky method :

$$f(a,b,c,d) = \Sigma m\,(2, 3, 4, 5, 13, 15) + \Sigma d\,(8, 9, 10, 11).$$ (7)

b) Using K-map simplify following Boolean expression & give implementation of same using gates $F(A,B,C,D) = \Sigma\,(2,4,8,15) + \Sigma D(0,3,9,12)$ (7)

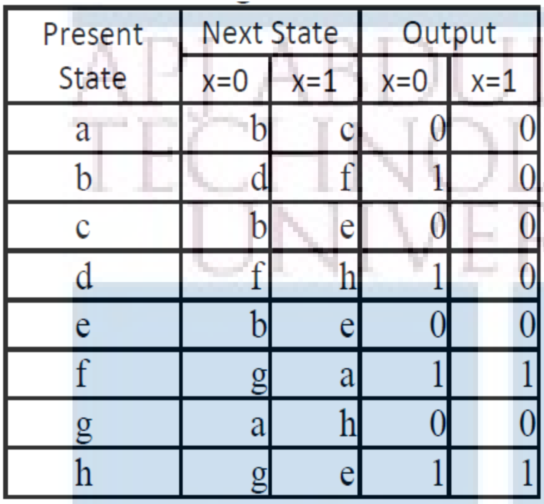
15. a) Design a combinational circuit to implement a 4-bit carry look-ahead adder. (7)

b) Design a 4-bit code-converter to convert BCD to gray code. (7)

OR

16. a) Implement the Boolean function f(w,x,y,z) = Σm(0,1,5,6,7,9,12,15) using 8-to-1 multiplexer. (7)
    b) Implement a 2-bit Magnitude comparator and write down its design procedure. (7)

17. a) For the following state table



Present State	Next State		Output	
	x=0	x=1	x=0	x=1
a	b	c	0	0
b	d	f	1	0
c	b	e	0	0
d	f	h	1	0
e	b	e	0	0
f	g	a	1	1
g	a	h	0	0
h	g	e	1	1

i. Draw the corresponding state diagram.
ii. Tabulate the reduced state table.
iii. Draw the state diagram corresponding to the reduced state table.
iv. Design the sequential circuit using flip-flops. [Hint:Unused states may be considered, as don’t cares.] (9)

b) Design D Flip Flop by using SR Flip Flop and draw the timing diagram. (5)

OR

18. a) Explain the state reduction in the sequential circuits using an example. (9)

    b) Draw the circuit of JK flip flop using NAND gates and explain its operation. (5)

19. a) Implement a four-bit universal shift register. Explain its design. (7)

    b) What do you mean by ripple counter? Design and implement a BCD ripple counter. (7)

OR

20. a) Tabulate the PLA programming table for the four Boolean functions listed below. Minimize the numbers of product terms.
    A(x, y, z) = Σm(1, 3, 5, 6)
    B(x, y, z) = Σm (0, 1, 6, 7)
    C(x, y, z) = Σm (3, 5)
    D(x, y, z) = Σm (1, 2, 4, 5, 7) (9)

    b) What are the operations that can be performed on a RAM? (5)

Syllabus

Module 1: NUMBER SYSTEM (9 Hours)
Number Systems – Decimal, Binary, Octal, Hexadecimal - conversion from one system to another – Representation of negative numbers using 2's compliment. Arithmetic Operations – Addition, Subtraction, Multiplication, Division of Binary numbers, Booths algorithm for multiplication, Representation of negative numbers, Representation of floating point numbers. Representation of BCD numbers, BCD Addition, Binary Codes – Gray codes – excess 3 code- Character Coding Schemes – ASCII, EBCDIC.
Module 2: BOOLEAN ALGEBRA & LOGIC GATES (9 Hours)
Boolean Algebra - Postulates of Boolean algebra - Canonical and Standard Forms - Simplification of Boolean Functions using Karnaugh Map - Product-of-Sums Simplification — Don't-Care Conditions – Quine –McClusky minimization technique – Basic Gates-Universal Gates.
Module 3: COMBINATIONAL LOGIC (9 Hours)
Combinational Circuits – Analysis and Design Procedures - Binary Adder-Sub tractor (Half & Full) - Carry look ahead adder, BCD adder, code converter, - Magnitude Comparator - Decoders – Encoders Parity Generator– Multiplexers – DE multiplexers – Implementation of Boolean functions using MUX.
Module 4: SEQUENTIAL LOGIC CIRCUITS (9 Hours)
Sequential Circuits - Storage Elements: Latches , Flip-Flops – RS, JK, D, T, Triggering of flip-flops, Master-Slave- Analysis of Clocked Sequential Circuits - Design Procedure-using JK,D & T.
Module 5: COUTERS AND SHIFT REGISTERS (9 Hours)
Registers - Shift Registers – SISO, PIPO, SIPO, PISO- Universal shift registers, Counters-Design of Counters- Synchronous & Asynchronous Counters — up-down counter, Decade counter, BCD counter, Johnson counter, Ring counter ,Memory & Programmable logic-RAM, ROM, PLA,PAL

Text Books

1. Mano M. M. and Michael D. Ciletti, *Digital Design*, 4/e, Pearson Education, 2013.
2. Thomas L. Floyd, *Digital Fundamentals*, 11th Edition, Pearson Education, 2015.
3. N. N. Biswas, *"Minimization of Boolean Functions,"* in *IEEE Transactions on Computers*, vol. C-20, no. 8, pp. 925-929, Aug. 1971. doi: 10.1109/T-C.1971.223373

Reference Books

1. Charles H Roth ,Jr, Lizy Kurian John, *Digital System Design using VHDL*,2/e, Cengage Learning
2. Mano M. M. and Michael D. Ciletti, Digital Design with an Introduction to the Verilog HDL, 5/e, Pearson Education, 2013.
3. Tokheim R. L., Digital Electronics Principles and Applications, 7/e, Tata McGraw Hill, 2007.
4. Rajaraman V. and T. Radhakrishnan, *An Introduction to Digital Computer Design*, 5/e, Prentice Hall India Private Limited, 2012.
5. Leach D, Malvino A P, Saha G, *Digital Principles and Applications,* 8/e, McGraw Hill Education, 2015.
6. M. Morris Mano, *Computer System Architecture*, 3/e, Pearson Education, 2007.

7. Harris D. M. and, S. L. Harris, *Digital Design and Computer Architecture*, 2/e, Morgan Kaufmann Publishers, 2013

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**NUMBER SYSTEM**	**9 Hours**
1.1	Number Systems – Decimal, Binary, Octal, Hexadecimal - conversion from one system to another – Representation of negative numbers using 2's compliment.	3 hours
1.2	Arithmetic Operations – Addition, Subtraction, Multiplication, Division of Binary numbers, Booths algorithm for multiplication, representation of negative numbers, Representation of floating point numbers.	4 Hours
1.3	Representation of BCD numbers , BCD Addition Binary Codes – Gray codes – excess 3 code- Character Coding Schemes – ASCII, EBCDIC	2 Hours
2	**BOOLEAN ALGEBRA & LOGIC GATES**	**9 Hours**
2.1	Boolean Algebra - Postulates of Boolean algebra - Canonical and Standard Forms	2 Hours
2.2	Simplification of Boolean Functions using Karnaugh Map - Product-of-Sums Simplification — Don't-Care Conditions	2 Hours
2.3	Quine –McClusky minimization technique	2 Hours
2.4	Basic Gates-Universal Gates.	3 Hours
3	**COMBINATIONAL LOGIC**	**9 Hours**
3.1	Combinational Circuits – Analysis and Design Procedures - Binary Adder-Subtractor - Carry look ahead adder, BCD adder	3 Hours
3.2	Code converter, - Magnitude Comparator - Decoders – Encoders – Multiplexers	3 Hours
3.3	Parity Generator– Multiplexers – DE multiplexers – Implementation of Boolean functions using MUX.	3 Hours
4	**SEQUENTIAL LOGIC CIRCUITS**	**9 Hours**
4.1	Sequential Circuits - Storage Elements: Latches , Flip-Flops – RS, JK, D, T, Triggering of flip-flops, race condition- Master-Slave	3 Hours
4.2	Analysis of Clocked Sequential Circuits	3 Hours
4.3	State Reduction and Assignment - Design Procedure- using JK,D & T	3 Hours
5	**COUTERS AND SHIFT REGISTERS**	**9 Hours**
5.1	Registers - Shift Registers – SISO, PIPO, SIPO, PISO- Universal shift registers	2 Hours
5.2	Design of Counters- Synchronous & Asynchronous Counters — up-down counter.	3 Hours
5.3	Counters-, Decade counter, BCD counter, Johnson counter, Ring counter	2 Hours
5.4	Memory & Programmable logic-RAM, ROM, PLA,PAL	2 Hour

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITT205	PROBLEM SOLVING USING PYTHON	PCC	3	1	0	4

Preamble: The syllabus is prepared with the view of preparing the Engineering Graduates capable of writing readable PYTHON programs to solve computational problems that they may have to solve in their professional life. The course content is decided to cover the essential programming fundamentals which can be taught within the given slots in the curriculum. This course has got 3 lecture hours and 1 tutorial hour per week for learning and practicing programming using PYTHON. The instructor is supposed to give homework/assignments to write simple programs in the rough record as and when the required theory part is covered in the class. The programs that require time and effort can be done in the Lab sessions. The students are expected to come prepared with the required program written in the rough record for the lab classes.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcomes	Bloom's Category
CO 1	Write programs using Python and learn its execution environment	Understand
CO 2	Apply programs to implement various computational tasks which requires loops and conditional statements	Apply
CO 3	Write programs using functions and packages	Understand
CO 4	Apply programs to implement the concept of file handling using python	Apply
CO 5	Design object oriented programs to implement daily life problems and their solutions	Apply

Mapping of course outcomes with program outcomes

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	1	1	1	-	-	-	-	-	-	-	-	-
CO 2	2	2	1	1	-	-	-	-	-	-	-	-
CO 3	3	2	2	2	2	-	-	-	-	-	-	1
CO 4	1	1	1	2	1	-	-	-	-	-	-	1
CO 5	3	1	3	2	1	1	-	-	-	-	-	1

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	25
Understand	20	20	35
Apply	20	20	40
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

Describe identifiers, variables, keywords, expressions and statements, Operators and operands.

Describe the expression evaluation in Python.

Describe the syntax of control statements in Python.

Write programs to solve problems using various control structures.

Differentiate between Break and Continue.

Course Outcome 2 (CO2):

Build applications of various string manipulations by using methods and functions available with string module in python.

Build applications of various List operations, including matrix representation.
Build applications of Tuple and various Tuple operations.
Build applications of Dictionary and its related operations, functions and methods.

Course Outcome 3(CO3):
Describe function definition and function access in python.
Differentiate between parameters and arguments.
Differentiate between type conversion and coercion.
Describe mathematical & lambda functions in Python.
Explain the concept of composition of functions in Python.
Explain Recursion and its implementation in Python.
Explain the concepts of modules and packages in Python. How and why import method is used.
Write programs to solve problems using the concept of functions and recursion.

Course Outcome 4 (CO4):
Apply the concept of file operations including opening, writing to and reading from files, and writing variables using Python.
Build applications to demonstrate the concept of Pickling.
Build programs to demonstrate the concept of Exception Handling in python.

Course Outcome 5 (CO5):
Build classes using python & Access class variables.
Build programs in Python to demonstrate the use of instances as arguments and return values.
Build programs in Python to demonstrate the concept of Constructors, class attributes and destructors.
Model the concept of Inheritance using Python.

Model Question paper

Course Code: ITT 205

Course Name: Problem Solving Using Python

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Write a python program to find the sum of all odd terms in a group of n numbers entered by the user.
2. What is the use of *pass* statement in Python?
3. Write a Python code to check whether two strings are equal or not.
4. Write a Python code to search an element in a list.
5. List the advantages of using functions in a program.
6. State the use of dump method with suitable example.

7. Write a function exists() which returns True if the given file exists and False if it does not.
8. Why exceptional handling is required in programming?
9. Describe the concept of Constructor with an example.
10. Explain the purpose of __init__() method in Python.

(10x3=30)

Part B

Answer any one Question from each module. Each question carries 14 Marks

11. (a) Write a Python program to find the largest and second largest of n numbers. Assume n>=3 and all the numbers are distinct. No sorting algorithm should be used. (7)

(b) What are arithmetic operators used in Python? Explain each using an example (7)

OR

12. (a) Write a Python program to print the odd composite numbers between m and n, where m and n are positive integers greater than 1. (8)

(b) Define the following

i) Atoms
ii) Identifiers
iii) Literals (6)

13. (a) Write a Python code to add two matrices using list. (6)

(b) Write a Python program to reverse a string and print whether its palindrome or not. (8)

OR

14. **(a)** How to create Dictionary in Python? Write a Python program to read and display a sparse matrix using dictionary. (7)

(b) Write a program to

i) convert all small letters in a string into capital letters
ii) find the occurrence of a given substring. (7)

15.(a) Compare the built-in functions int() and str() with examples. (4)

(b) Write a program using function to display a multiplication table of n*n size, for any given 'n'. (5)

(c) Write a program using function to display a find the binomial coefficient, $^{n}c_{r}$. (5)

OR

16 (a). What is recursion? Write a recursive function to find the factorial of a number. (6)

(b) Write a program using function to check the type of a triangle by getting the values from the user. (8)

17. (a) How exceptions are handled in Python? Illustrate with example. (10)

(b) Write a program to read numbers sorted in one file and store the sorted numbers in another file after deleting duplicates. (4)

OR

18. (a). Describe the use of try-except method in Python with suitable Illustrations. (6)

(b) Write a Python code to read a text file, copy the contents to another file after removing the blank lines. (8)

19. (a). Write a Python code to create a class named 'Member' having the following members: Data members Name, Age, Phone number, Address, Salary. It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same. (8)

(b) Create a class person with attributes Name, age, salary and a method display() for showing the details. Create two instances of the class and call the method for each instance. (6)

OR

20. (a) Define the terms class, attribute, method and instance with the help of an example. (4)

(b) Write a Python program to find out the total mark of a student using multiple inheritance. Declare a Student class. Student class should have the member functions for accept and display student details. Declare a Sports class to accept and display sports marks of the student. Derive a class statement from both the classes. This class should have the member functions to accept marks of three subjects and add those marks with sports marks and display the result. (10)

(14 x 5=70)

Syllabus

Module 1	**9 hours**
Introduction To Python: Understanding Python-identifiers, variables, keywords, expressions and statements, evaluation of expressions, Operators and operands, operator precedence, indentation. **Python Program Flow Control:** Decision making- if, if..else, elif. Loops - for, while, for...else, while...else, Control statements using pass, continue, break.	
Module 2	**9 hours**
Strings and lists – string traversal, string slices and comparison with examples, The string module, character classification. List- List values, accessing elements, list membership, Lists and for loops, List operations, List slices, List deletion, Matrices. Tuples - mutability and tuples, tuple assignment, Tuples as return values, Tuple operations. Dictionaries – operations and methods.	
Module 3	**9 hours**
Python Functions, Modules and Packages: Function definition, calling functions, parameters and arguments, the return statement, type conversion and coercion, composition of functions, Lambda function, mathematical functions, user-defined functions, Recursion, Modules- Built-in modules, creating modules, import statement. Packages in Python - importing modules from a package.	

Module 4	9 hours
Python Files and exceptions: Python file handling, open, write, read text files, writing variables, Directories in Python, Pickling, Exception Handling.	

Module 5	9 hours
Python Object Oriented Programming: Introduction to classes and objects - class definition, attributes, instances, sameness, instances as arguments and return values. Constructor, class attributes and destructors, Inheritance.	

Text Books

1. Allen Downey, Jeffrey Elkner, Chris Meyers, " How to think like a Computer Scientist-Learning with Python", Green Tea Press, First edition, 2002.
2. Mark Lutz,"Learning Python: Powerful Object-Oriented Programming" , O'Reilly Media Inc.,5th,2013

Reference Books

1. Kenneth A. Lambert, B. L. Juneja, "Fundamentals of Python", Cengage Lerning India Pvt. Ltd., 2015.
2. S.A.Kulkarni, "Problem Solving and PYTHON Programming", 2nd edition, Yes Dee Publishing Pvt Ltd, 2018
3. Mark Summerfield,"Programming in Python 3: A Complete Introduction to the Python Language", Pearson Education, 2nd,2018
4. Yashavant Kanetkar ,Aditya Kanetkar ,"Let Us Python ",BPB Publications, 1st Edition, 2019
5. Allen Downey, "Learning with Python", Dreamtec Press, 1st Edition, 2015
6. https://docs.python.org/3/reference/

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Introduction To Python:**	**9 hours**
1.1	Understanding Python-identifiers, variables, keywords, expressions and statements.	2
1.2	Evaluation of expressions, Operators and operands, operator precedence, indentation	1

1.3	Python Program Flow Control: Decision making- if, if..else, elif.	2
1.4	Loops - for, while, for...else, while...else	2
1.5	Control statements using pass, continue, break.	2
2	**Strings and lists:**	**9 hours**
2.1	String traversal, string slices and comparison with examples	1
2.2	The string module, character classification.	1
2.3	List- List values, accessing elements, list membership, Lists and for loops, List operations, List slices, List deletion	2
2.4	Matrices	1
2.5	Tuples- mutability and tuples, tuple assignment, tuples as return values, Tuple operations.	2
2.6	Dictionaries – operations and methods.	2
3	**Python Functions, Modules And Packages:**	**9 hours**
3.1	Function definition, calling functions, parameters and arguments, the return statement.	1
3.2	Type conversion and coercion, composition of functions	1
3.3	Lambda function, mathematical functions	1
3.4	user-defined functions	1
3.5	Recursion	1
3.6	Modules -Built-in modules	1
3.7	Creating modules, import statement.	1
3.8	Packages in Python - importing modules from a package.	2
4	**Python Files and exceptions:**	**9 hours**
4.1	Python file handling, open, write, read text files	4
4.2	Writing variables	1
4.3	Directories in Python	1
4.4	Pickling	1
4.5	Exception Handling.	2

5	**Python Object Oriented Programming:**	**9 hours**
5.1	Introduce classes and objects	1
5.2	Class definition, attributes, instances, sameness	1
5.3	Instances as arguments and return values.	1
5.4	Constructor	2
5.5	Class attributes and destructors	2
5.6	Inheritance	2

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITL201	DATA STRUCTURES LAB	PCC	0	0	3	2

Preamble:

This lab is intended to make the students capable of

- understanding the importance of data structures, abstract data type, and their basic usability in different application,
- Implementing linear and non-linear data structures using linked lists and arrays.
- Applying various data structure such as stacks, queues, trees, graphs, etc. to solve various computing problems
- Identifying suitable data structure and algorithm to solve a real world problem.

Prerequisite: programming in C

Course Outcomes: After the completion of the course the student will be able to

CO_No	Course Outcome(CO)	Bloom's Category
CO 1	**Compare** various kinds of searching and sorting techniques	level 5: Evaluate
CO 2	**Construct** Linear and nonlinear data structures using arrays and linked list	level 6:Create
CO 3	**Develop** Programs employing dynamic memory management	level 6:Create
CO 4	**Choose** appropriate data structure to solve various computing problems.	level 5: Evaluate
CO 5	**Originate** hash tables and collision resolution Techniques	level 6:Create
CO 6	**Identify** suitable data structure and algorithm to solve a real world problem.	level 3:Apply

Mapping of course outcomes with program outcomes

COs	PROGRAMME OUTCOMES (PO)										
		4	5	5	6	3	2	3	3	3	3	3
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3	3	2	2	1	-	-	-	-	2	1
CO 2	3	3	3	3	3	1	-	-	-	-	2	1
CO 3	3	3	3	3	3	1	-	-	-	-	2	1
CO 4	3	3	3	3	3	1	-	-	-	-	2	1
CO 5	3	3	3	3	3	1	-	-	-	-	2	1
CO 6	3	3	3	3	3	1	-	-	-	-	2	1

3/2/1: high/medium/low

Assessment Pattern

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	75	75	2.5 hours

Continuous Internal Evaluation Pattern:

Attendance	:	15 marks
Continuous Assessment	:	30 marks
Internal Test (Immediately before the second series test)	:	30 marks

End Semester Examination Pattern: The following guidelines should be followed regarding award of marks

(a) Preliminary work	: 15 Marks
(b) Implementing the work/Conducting the experiment	: 10 Marks
(c) Performance, result and inference (usage of equipments and trouble shooting)	: 25 Marks
(d) Viva voce	: 20 marks
(e) Record	: 5 Marks

General instructions: Practical examination to be conducted immediately after the second series test covering entire syllabus given below. Evaluation is a serious process that is to be conducted under the equal responsibility of both the internal and external examiners. The number of candidates evaluated per day should not exceed 20. Students shall be allowed for the University examination only on submitting the duly certified record. The external examiner shall endorse the record.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Develop a C program to implement insertion sort, selection sort and bubble sort.
2. Design a program to Implement i) Quick sort ii) Merge sort.
3. Create programs for i) Linear Search ii) Binary Search.

Course Outcome 2 (CO2)

1. Design a Menu driven program to implement singly linked list operations with options for insertion, deletion, search and traversal.
2. Develop Menu driven program to implement doubly linked list operations with options for insertion at front, insertion at end, deletion at front, deletion at end and traversal.

Course Outcome 3(CO3):

1. Design a Menu driven program to implement singly linked list operations with options for insertion, deletion, search and traversal.
2. Simulate first fit, best fit and worst fit memory allocation strategies using linked list.

Course Outcome 4 (CO4):

1. Device Dijkstra's Algorithm for finding Shortest path
2. Apply Queue and stack in Breadth First Search and Depth First Search respectively

Course Outcome 5 (CO5):

1. Implement hash table using various mapping functions
2. Resolve the collisions if any using collision resolution techniques like linear Probing, Random Probing, Double hashing and Quadratic Probing

Course Outcome 6 (CO6):

1. Design and implement an application program which is to be used in Ticket counter, where First person gets ticket first and go out first, using suitable data structure. The program should do the following functions

   a) When new person come to the counter the details of the person (Name and age) should be added to the data structure.

   b) After issuing tickets, the details of the corresponding person should be deleted from the data structure

   (Hint: Each node of linked list should contain fields Name, Age and Rink)
2. Design and implement an application program to store the details of pass percentage of the college in chronological order of years (oldest to newest) and retrieve the information in reverse chronological order of years, using suitable data structure.

   Hint: A Node of linked list may contain the fields Year, Pass Percentage and Rlinlk.

   Menu may contain

   a) Store details

   b) Retrieve details

LIST OF EXPERIMENTS

1. Develop a C program to implement insertion sort, Selection sort and bubble sort*.
2. Design a program to Implement i) Quick sort ii) Merge sort*.
3. Create Programs for i) Linear Search ii) Binary Search*.
4. Create a menu driven program to implement singly linked list operations with options for insertion, deletion, search and traversal*.
5. Device a menu driven program to implement doubly linked list operations with options for insertion at front, insertion at end, deletion at front, deletion at end and traversal.
6. Apply linked list concept to perform polynomial addition*
7. Simulate first fit, best fit and worst fit memory allocation strategies using linked list*.
8. Develop a program to perform stack operations using i) array ii) linked list*.
9. Perform queue operations using i) array ii) linked list*.

10. Apply stack to perform i) Infix to postfix conversion ii) Postfix evaluation *
11. Develop a program to perform preorder, in-order, post order traversals on binary trees*
12. Construct binary search trees to perform insertion, deletion, search
13. Apply Queue and stack in Breadth First Search and Depth First Search respectively *
14. Device Dijkstra’s Algorithm for finding Shortest path
15. Resolve the collisions if any using collision resolution techniques like linear Probing, Random Probing, Double hashing and Quadratic Probing*
    (* indicates mandatory experiments.)

Text Books

1. Samanta D., Classic Data Structures, Prentice Hall India, 2/e, 2009.
2. Ellis horowitz, Sartaj Sahni, Fundamentals of Data structures, Galgotia Booksource

Reference Books

1. Horwitz E., S. Sahni and S. Anderson, F
2. undamentals of Data Structures in C, University Press (India), 2008.
3. Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication,1983.
4. Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill, 1995.
5. Peter Brass, Advanced Data Structures, Cambridge University Press, 2008
6. Lipschuts S., Theory and Problems of Data Structures, Schaum’s Series, 1986.
7. Wirth N., Algorithms + Data Structures = Programs, Prentice Hall, 2004.
8. Hugges J. K. and J. I. Michtm, A Structured Approach to Programming, PHI, 1987.
9. Martin Barrett, Clifford Wagner, And Unix: Tools For Software Design, John Wiley, 2008 reprint.

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITL203	PROGRAMMING AND SYSTEM UTILITIES LAB	PCC	0	0	3	2

Preamble: This laboratory course is meant for understanding the fundamental system utilities. The course is also aimed for understanding and practicing the programming language Python.

Prerequisite: PYTHON programming knowledge and Computer Fundamentals.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcomes	Bloom's Category
CO 1	Develop readable* Python programs by making use of basic constructs- Decision controls, Looping controls, Lists, Tuple and Strings	Create
CO 2	Design modular Python programs using normal and recursive functions	Create
CO3	Design programs using Dictionaries and Files	Create
CO 4	Experiment with the basic Windows/ Linux administration & network configuration utilities	Apply
CO 5	Experiment with version control tools using git	Apply
readable* - readability of a program means the following: 1. Logic used is easy to follow 2. Standards to be followed for indentation and formatting 3. Meaningful names are given to variables 4. Concise comments are provided wherever needed		

Mapping of course outcomes with program outcomes

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	2	3	1	3	-	-	-	-	-	-	-
CO 2	2	3	3	3	3	-	-	-	-	-	-	-
CO 3	2	3	3	3	3	-	-	-	-	-	-	-
CO 4	2	2	1	2	1	-	-	-	-	-	-	-
CO 5	-	-	2	-	3	-	-	-	3	2	-	-

3/2/1: High/Medium/Low

Assessment Pattern

Mark distribution

Total Marks	**CIE**	**ESE**	**ESE Duration**
150	75	75	2.5 hours

Continuous Internal Evaluation Pattern:

Attendance	:	15 marks
Continuous Assessment	:	30 marks
Internal Test (Immediately before the second series test)	:	30 marks

End Semester Examination Pattern: The following guidelines should be followed regarding award of marks

(a) Preliminary work	: 15 Marks
(b) Implementing the work/Conducting the experiment	: 10 Marks
(c) Performance, result and inference (Usage of equipments and troubleshooting)	: 25 Marks
(d) Viva voce	: 20 marks
(e) Record	: 5 Marks

General instructions: Practical examination to be conducted immediately after the second series test covering entire syllabus given below. Evaluation is a serious process that is to be conducted under the equal responsibility of both the internal and external examiners. The number of candidates evaluated per day should not exceed 20. Students shall be allowed for the University examination only on submitting the duly certified record. The external examiner shall endorse the record.

Course Level Assessment Questions

Course Outcome 1 (CO1):

Develop a python program to

1. Print all prime numbers with in an interval
2. Search an element in a list
3. Input a list of n numbers. Calculate and display the average of numbers. Also display the square of each value in the list
4. Add two matrices.
5. Find the number of occurrences of a given substring in a string.
6. Count the number of vowels, consonants, words and question marks in a given string.

Course Outcome 2 (CO2):

Develop a python program to

1. Find the value of **`nCr`** using function.
2. Implements calculator with functions like add, subtract, multiply, divide, exponent etc.
3. Find factorial of a given number using recursion.
4. Find nth Fibonacci number using recursion.

Course Outcome 3 (CO3):

1. Develop a python program to create dictionary of phone numbers and names of n persons. Display the contents of the dictionary in alphabetical order of their names

2. Develop a Python program to implement the following scenario. A book shop details contains the Title of book and Number of copies of each title. As books are added to shop the number of copies in each should increase and as books are sold the number of copies in each should decrease.
3. Develop a python code to read a text file, copy the contents to another file after removing the blank lines.
4. Develop a python program to implement the following scenario. Given a file "`data.txt`" with three columns of data separated by spaces. Read it into 3 separate simple sequences.
5. Create a class student with attributes name, rollno and a method showData() for showing the details. Create two instances of the class and call the method for each instance. Develop a python program to implement the scenario.

Course Outcome 4 (CO4):

Perform the following operations:

1. Apply the use of ATTRIB windows command to change the attributes of a file.
2. Create a file **`xyz.txt`** and change the ownership of this file to some other user on your machine.
3. Create a file **`hello.txt`** and make it executable.
4. Create a new user account and home directory called "Duck" and Set the user account "Duck's" expiry date as 07 - 07 - 2020
5. Check the network connectivity of your computer using suitable Linux commands

Course Outcome 5 (CO5):

Perform the following:

1. Create a directory in your machine and make it as a repository and perform the following
   a. Create a text file and add some content into it.
   b. Add the file to the staging area of the Git repository.
   c. Commit the file to your repository.
   d. See the commit details using git log command.
2. Go to your Git repository and perform the following
   a. Do some modifications in your text file. Commit the changes.
   b. Try to revert to your old revision, again do some modifications in your text file and try to discard the changes.

List of Experiments

Part A : Programming in Python

1. **Basic programming experiments** to familiarization of data types and input-output statements
2. **Decision making, branching and looping statements**
3. **Function & Function calls**
   1. Function definitions and access
   2. Parameters and arguments
   3. Recursion
4. **Strings**
   a) String traversal, join, slicing
   b) String searching, Comparison

c) Other important String methods

5. **Lists, Tuples and Dictionaries**
   a) Creation of List & List Operations
   b) Tuple and Tuple operations
   c) Creation of Dictionary and Operations
   d) Comparison of List and Tuple
6. **Matrix representation**
   a) Creating matrix
   b) Matrix operations - addition, subtraction and multiplication
7. **Files and Operations**
   a) Files - defining, opening/closing, read/write operations
   b) Exceptions in Python
   c) Pickling
8. **Object Oriented Programming using Python**
   c) Creation of Classes & Instances, method calling
   d) Constructor & Destructor concepts
   e) Implementation of Inheritance

Part B : System Utilities

Basic Windows/Linux Administration Utilities

1. **Experiments on Windows Operating System**

   a. Perform the following commands

   DIR, TYPE, DEL, ERASE, MD, CD, COPYCON, RMDIR, REN, VER, DATE, TIME, TREE, PATH, CLS, RMDIR, BREAK, SET, EXIT, APPEND, CHKDISK, ATTRIB, SYS, EDIT, XCOPY, DISKCOPY

   b. Explore and describe some system utility like **regedit**, memory partioning, control panel and window tools

2. **Experiments on Linux Operating System**

   a) Perform general purpose utilities in Linux:

   echo, uname, whoami, passwd, date, date +%T, date +%h, date +%m, date +%y, date +”%h%y”, cal, cal 12 2030, echo $HOME, pwd, ls, ls –all, ls –l, cat, cat > file1, cat >> file2, ls -l >fileinfo

   b) Familiarize working with files and managing file attributes

3. **Network Configuration Utilities**

   a) **ifconfig** utility, enable/disable network interface, **traceroute**,

   telnet, **nslookup, netstat, w, scp**, etc

   b) Connecting to the internet

4. **GIT for version control**
   a. Installation and configuration of Git on Ubuntu and Windows operating systems
   b. Perform Basic Git Commands (**`gitinit, add, status, commit`**, and **`log`**) and Git **`checkout`** command

Text Books

1. Allen Downey, Jeffrey Elkner, Chris Meyers, " How to think like a Computer Scientist-Learning with Python", Green Tea Press, First edition, 2002.
2. Mark Lutz,"Learning Python: Powerful Object-Oriented Programming" , O'Reilly Media Inc.,5th,2013

Reference

1. S.A.Kulkarni, "Problem Solving and PYTHON Programming", 2nd edition, Yes Dee Publishing Pvt Ltd, 2018
2. Kenneth A. Lambert, B. L. Juneja, "Fundamentals of Python", Cengage Learning India Pvt. Ltd., 2015.
3. Mark Summerfield," Programming in Python 3: A Complete Introduction to the Python Language", Pearson Education, 2nd,2018
5. Yashavant Kanetkar ,Aditya Kanetkar ,"Let Us Python ",BPB Publications, 1st Edition, 2019
6. Allen Downey, "Learning with Python", Dreamtec Press, 1st Edition, 2015
7. https://docs.python.org/3/reference/
8. Version Control with Git: Powerful tools and techniques for collaborative software development 2nd Edition, Kindle Edition by Jon Loeliger, Matthew McCullough
9. https://spoken-tutorial.org/

SEMESTER -3

MINOR

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITT281	JAVA PROGRAMMING	VAC	3	1	0	4

Preamble: The syllabus is prepared with the intended to deliver students the elementary concepts of Java Programming and equip them to code java application built over those concepts. It also introduces to them advanced level areas like event driven programming with Java.

Prerequisite: Basics of Programming

Course Outcome (CO): After completion of the course, the student will be able to

CO No.	Course Outcome	Bloom's Category
CO1	Summarize Object Oriented Programming concepts and basic characteristics of Java	Understand
CO2	Summarize basic java packages, inheritance and interfaces	Understand
CO3	Summarize exceptions and I/O streams concepts	Understand
CO4	Demonstrate the usage of threads and generics classes	Understand
CO5	Build simple Graphical User Interface programs with Java	Apply

Mapping of Course Outcomes with Program Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	-	1	-	-	-	-	-	-	-
CO2	1	1	2	1	-	-	-	-	-	-	-	-
CO3	3	1	1	-	-	-	-	-	-	-	-	-
CO4	3	1	1	1	-	-	-	-	-	-	-	-
CO5	3	1	1	2	2	-	-	-	-	-	-	-

3/2/1: High/Medium/Low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests (Marks)		End Semester Examination (Marks)
	1	2	
Remember	15	15	30
Understand	25	25	50
Apply	10	10	20
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain core principles of Object Oriented Design.
2. Describe the characteristics of Java.
3. Illustrate the concept of Polymorphism with an example.
4. Describe Java Source File Structure.
5. Explain constructors in detail

Course Outcome 2 (CO2):

1. Explain the concept of Super classes and sub classes.
2. Differentiate classes and interfaces.
3. Illustrate the concept of Interfaces with an example.
4. Describe Final methods and classes in java.
5. Explain extending interfaces in Java.

Course Outcome 3 (CO3):

1. Explain throwing and catching exceptions in Java.
2. Describe the different built-in exceptions in Java.
3. Illustrate the concept of Exception handling with an example.
4. Describe Byte and character streams in java.
5. Explain stack trace elements.

Course Outcome 4 (CO4):

1. Explain the concept of multithreading in Java.
2. Describe the different bounded types in Java.
3. Illustrate thread synchronization with example.
4. Describe inter-thread communication in java.
5. Explain the concept of generic programming in Java.

Course Outcome 5 (CO5):

1. Experiment with AWT in Java.
2. Build AWT programs for simulating Calculator.
3. Experiment with different Form components in Swing.

Model Question Paper

PART A

Answer all questions, each carries 3 marks

1. Why are java programs said to be platform independent?
2. Explain how access modifiers are used to control the visibility of identifiers?
3. What is the use of interface in java? Give example.
4. What is a package? How a class within a package is compiled and executed?
5. List out different exception classes in Java.
6. Write a note on byte stream and character stream classes in java
7. List out different bounded types in java
8. Draw the life cycle of a thread showing different stages and methods invoked.
9. List any 3 event sources and their corresponding generated event type and listeners used
10. What are the advantages of using swing?

PART B

11.

a. What is the use of constructor in java? Give examples for different types of constructors. (7)

b. Define a java class having overloaded methods to calculate the area of a rectangle and circle. (7)

OR

12.

a. Write a Java program that counts the number of odd and even numbers in an array of 10 integers. (8)

b. Briefly explain the architecture of JVM. (6)

13.

a. List any five methods of String class, give examples. (5)

b. With the help of examples, explain how inheritance is implemented in java. (9)

OR

14.

a. Write a java program to count the number of occurrences of a particular word in a sentence using string handling methods. (8)

b. Differentiate between abstract class and interface. (6)

15.

a. Explain exception handling in java. Briefly explain various exception handling keywords in java with examples(9)

b. Write a Java program that counts the number of words in a text file. (6)

OR

16.

a. Write a Java program that accepts N integers through console and sort them in ascending order.(8)

b. Explain the scenario under which the following three exceptions occur, NumberFormatException, ArrayIndexOutOfBoundsException and ArithmeticException (6)

17.

a. Explain the different ways of creating threads in java (9)

b. Explain the concept of generic programming in Java. (5)

OR

18.

a. Write a java program to create two threads, one for writing even numbers and the other for writing odd numbers upto 100 in two different files. (12)

b. What are the uses of synchronized keyword?(2)

19.

a. Demonstrate the usage of any four Form components in Swing with an example. (4)

b. Implement a Java AWT program for simulating Calculator(10)

OR

20. Write a java AWT based java program to display Fibonacci numbers in a list control upto a limit entered using TextField.The event handling as well as code for clearing the

components must be provided (14)

Syllabus

MODULE 1: INTRODUCTION TO OOP AND JAVA FUNDAMENTALS (10 Hours)
Object Oriented Programming – Abstraction – objects and classes – Encapsulation- Inheritance – Polymorphism- OOP in Java – Characteristics of Java – The Java Environment – Java Virtual Machine- Java Source File Structure – Compilation. data types, operators, control statements, Introduction to Java programming.– Classes fundamentals, objects, methods, constructors, parameter passing, overloading, access control keywords– static members -Comments, Arrays-Java Documentation usage
MODULE 2: INHERITANCE AND INTERFACES (10 Hours)
Inheritance – Super classes- sub classes –Protected members – constructors in sub classes- the Object class – abstract classes and methods- final methods and classes – Interfaces – defining an interface, implementing interface, differences between classes and interfaces and extending interfaces – Defining and importing packages. Strings-Java Built-in Classes and it's usage
MODULE 3: EXCEPTION HANDLING AND I/O (7 Hours)
Exceptions – exception hierarchy – throwing and catching exceptions – built-in exceptions, creating own exceptions, Stack Trace Elements. Input / Output Basics – Streams – Byte streams and Character streams – Reading and Writing Console – Reading and Writing Files
MODULE 4: MULTITHREADING AND GENERIC PROGRAMMING (8 Hours)
Differences between multithreading and multitasking, thread life cycle, creating threads, synchronizing threads, Inter-thread communication. Generic Programming – Generic classes – generic methods – Bounded Types – Restrictions and Limitations
MODULE 5: EVENT DRIVEN PROGRAMMING (10 Hours)
Graphics programming – Frame – Components – working with 2D shapes – Using color, fonts, and images – Basics of event handling – event handlers – adapter classes – actions – mouse events – AWT event hierarchy – Introduction to Swing – layout management – Swing Components – Text Fields , Text Areas – Buttons- Check Boxes – Radio Buttons – Lists- choices- Scrollbars – Windows –Menus – Dialog Boxes.

Text Books

1. Herbert Schildt, ―Java The complete reference‖, 8th Edition, McGraw Hill Education, 2011.
2. Cay S. Horstmann, Gary cornell, ―Core Java Volume –I Fundamentals‖, 9th Edition, Prentice Hall, 2013.

Reference Books

1. Paul Deitel, Harvey Deitel, ―Java SE 8 for programmers‖, 3rd Edition, Pearson, 2015.
2. Steven Holzner, ―Java 2 Black book‖, Dreamtech press, 2011.
3. Timothy Budd, ―Understanding Object-oriented programming with Java‖, Updated Edition, Pearson Education, 2000.

Course Contents and Lecture Schedule

No.	Topic	No. of Lectures
1	**Introduction**	**10**
1.1	**Fundamentals of Object Oriented Programming:** Abstraction, objects and classes, Encapsulation, Inheritance, Polymorphism	3
1.2	**OOP in Java**: Characteristics of Java,The Java Environment – Java Virtual Machine-, Java Source File Structure , Compilation, data types, operators, control statements	3
1.3	**Introduction to Java programming:** Classes fundamentals, objects, methods, constructors, parameter passing, overloading, access control keywords, static members ,Comments, Arrays, -Java Documentation usage	4
2	**Inheritance And Interfaces**	**10**
2.1	**Inheritance:** Super classes, sub classes, Protected members, – constructors in sub classes, the Object class, abstract classes and methods, final methods and classes	5
2.2	**Interfaces**: defining an interface, implementing interface, differences between classes and interfaces and extending interfaces, Defining and importing packages. Strings, Java Built-in Classes and it's usage	5
3	**Exception Handling And I/O**	**7**
3.1	**Exceptions:** exception hierarchy, throwing and catching exceptions, built-in exceptions, creating own exceptions, Stack Trace Elements.	4
3.2	**Input / Output Basics:** Streams, Byte streams and Character streams, Reading and Writing Console, Reading and Writing Files	3
4	**Multithreading And Generic Programming**	**8**
4.1	**Multithreading:** Differences between multi threading and multitasking, thread life cycle, creating threads, synchronizing threads, Inter-thread communication.	4

4.2	**Generic Programming:** Generic classes, generic methods, Bounded Types, Restrictions and Limitations	4
5	**Event Driven Programming**	**10**
5.1	**Graphics programming:** Frame, Components, working with 2D shapes, Using color, fonts, and images	3
5.2	**Basics of event handling:** Event handlers, adapter classes, actions, mouse events, AWT event hierarchy	3
5.3	**Introduction to Swing:** layout management, Swing Components, Text Fields , Text Areas, Buttons, Check Boxes, Radio Buttons, Lists, choices, Scrollbars, Windows, Menus, Dialog Boxes.	4

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITT283	DATA COMMUNICATION	VAC	3	1	0	4

Preamble: The syllabus is prepared in view of devising students capable of understanding the essential concepts and terminology used for data communication.

Prerequisite: NIL

Course objectives

- To introduce basic terminology and concepts used in data transmission
- To understand encoding techniques used in data communication
- Familiarize students the fundamental knowledge about computer networks

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome(CO)	Bloom's Category
CO 1	**Describe** the fundamental concepts of data communication, network models, and standards and wired networks.	Understand
CO 2	**Identify** different transmission media, data and signals.	Apply
CO 3	**Demonstrate** different encoding techniques used for analog to digital conversion and vice versa.	Understand
CO 4	**Describe** the different types of bandwidth utilization techniques and basic principles of switching.	Understand
CO 5	**Describe** the different access methods, channelization and wireless networks.	Understand

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	2	-	-	-	-	-	-	-	-	-	1
CO 2	1	2	-	2	2	-	-	-	-	-	-	1
CO 3	2	1	1	-	-	-	-	-	-	-	-	1
CO 4	3	2	2	1	1	-	-	-	-	-	-	1
CO 5	1	-	1	-	2	-	-	-	-	-	1	2

3/2/1: high/medium/low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	30
Understand	20	20	50
Apply	20	20	20
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

End Semester Examination Pattern: There will be two parts; **Part A** and **Part B**. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer *all* questions. Part B contains 2 questions from each module of which student should answer *any one*. Each question can have maximum 2 sub-divisions and carry 14 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain the fundamentals of data communication.

2. Explain the different types of networks.

3. Explain the different types of network models and standards.

4. Explain the different types of wired networks.

Course Outcome 2 (CO2)

1. Identify the different types of transmission media.

2. Identify the different types of data and signals.

3. Identify data rate limits for noisy and noiseless channel.

Course Outcome 3 (CO3):

1. Explain the fundamentals of data transmission.

2. Describe about different transmission modes.

3. Demonstrate the analog data conversion to analog signals and analog data into digital signals.

4. State and explain different keying techniques.

Course Outcome 4 (CO4):

1. Describe the functionality of different multiplexing techniques.

2. Illustrate about different spread spectrum techniques.

3. Describe basic principles of switching.

Course Outcome 5 (CO5):

1. Illustrate the random access methods used for collision avoidance.

2. Explain different controlled access methods.

3. Describe about different channelization protocols.

4. Demonstrate different services provided by wireless networks

Model Question Paper

PART A **(10*3=30)**

(Each Question carries 3 Marks)

1. List out the key components of data communication.

2. Summarize the essential criteria's required for an ideal network?

3. Mention Shannon's Theorem. Find out the channel capacity of a noisy channel which is having signal to noise ratio almost zero.

4. Analyze different characteristics used to measure the network performance.

5. Define the role of scrambling in digital transmission.

6. Calculate the baud rate for the given bit rate and type of modulation

   a. 4000 bps,QPSK

b. 36000 bps, 64-QAM

7. Identify the different phases used for communication in circuit switched network.

8. Define frequency division multiplexing and mention its applications.

9. Compare and contrast Pure ALOHA and Slotted ALOHA.

10. Examine the significance of transmission convergence sublayer in WiMax.

PART B **(5*14=70)**

11. Illustrate different types of connections and topologies used in network for connecting devices with the help of diagram.

OR

12. Substantiate the need of OSI model in network communication? Briefly explain the functionalities of each layer in OSI model.

13. Analyse different types of transmission impairments occurring in transmission media in detail.

OR

14. Explain in detail about guided and unguided media used for data transmission.

15. Discuss in detail about different line coding schemes.

OR

16. Identify the different techniques used for changing an analog signal to digital data.

17. Illustrate in detail about Time division multiplexing. With the help of neat sketch explain different schemes of Time division multiplexing.

OR

18. Examine the need of spread spectrum? Which are different spread spectrum techniques?

19. Discuss in detail about different channelization protocols.

OR

20. Demonstrate in detail about Bluetooth .List out its applications.

Syllabus

Module 1 (7 Hours)
Introduction to Data Communication- Components, Data Representation, Data Flow. Networks - Network Criteria, Physical structures, Physical Topology, Network Types- LAN, WAN, Switching -Internet -Network Models-OSI Model.
Module 2 (9 Hours)
Transmission media – Guided media – Twisted pair cable, coaxial cable, fiber optic cable, Unguided media - Radio waves, Microwaves, Infrared. Data and signals - Periodic analog signals-digital signals-transmission impairment - Attenuation, Distortion - Noise- different types of noise – Data rate limits-Noiseless channel, Noisy Channel, Performance
Module 3 (10 Hours)
Digital data transmission – Digital to Digital Conversion –Line Coding, Line Coding Schemes, Block coding, Transmission modes- Serial, Parallel, Synchronous, Asynchronous and Isochronous transmission. Encoding analog data into analog signals - AM, FM, PM. Encoding analog data into digital signals - PCM, DM – Keying Techniques - ASK, FSK, PSK, QAM
Module 4 (7 Hours)
Multiplexing- Frequency Division Multiplexing (FDM) – Time Division Multiplexing (TDM), Synchronous Time Division Multiplexing –Statistical time Division multiplexing Spread spectrum-The concept of spread spectrum – frequency hopping spread spectrum – direct sequence spread spectrum. Basic Principles of Switching-Circuit Switch Networks, Packet Switching, Structure of Switch
Module 5 (12 Hours)
Media Access Control – Random Access - Controlled Access-Channelization. Introduction to Wireless LAN-IEEE 802.11 –Bluetooth- WiMax, Cellular Telephony - 1G, 2G, 3G, 4G, 5G.

Text Books

1. Behrouz A Forouzan, “Data Communication and Networking”, McGraw Hill Education(india)Private limited,Fifth edition, 2013.

Reference Books

1. Stallings W., Data and Computer Communications, 8/e, Prentice Hall, 2007.

2. Tanenbaum A. S and D. Wetherall, Computer Networks, Pearson Education, 2013

3. Taub & Schilling, Principles of Communication Systems: Tata McGraw-Hill

4. Simon Haykin,Communication Systems: John Wiley & Sons. Pvt. Ltd

5. Das, Mullick & Chatterjee, Principles of Digital Communication: Wiley Eastern Ltd.

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**Overview (7 Hours)**	
1.1	Introduction to Data Communication and networks	3
1.2	Network Models-OSI Model	4
2	**Transmission media (9 Hours)**	
2.1	Guided media and UnGuided media	2
2.2	Data and signals	3
2.3	Transmission impairments	2
2.4	Data rate limits and performance	2
3	**Digital data transmission (10 Hours)**	
3.1	Digital to digital Conversion	5
3.2	Transmission modes	1
3.3	Encoding analog data into analog signals	1
3.4	Encoding analog data into digital signals	2
3.5	Keying Techniques	1
4	**Bandwidth utilization: Multiplexing and Spectrum spreading, Switching (7 Hours)**	
4.1	Multiplexing	3
4.2	Spread spectrum	2
4.3	Basic Principles of switching	2
5	**Media Access Control and Wireless LANs (12 Hours)**	
5.1	Random Access	2
5.2	Channelization	1
5.3	Wireless LANs	5
5.4	Other Wireless networks	4

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITT285	SOFTWARE ENGINEERING	VAC	3	1	0	4

Preamble: The syllabus is prepared with the view of preparing the Engineering Graduates capable of understanding essential concept of software engineering and software development process.

Prerequisite: Basics of programming

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category
CO1	**Summarize** different software development models	Level 2: Understand
CO2	**Identify** methods for requirement analysis, specification, design & testing	Level 3: Apply
CO3	**Explain** the software quality maintenance measures	Level 2: Understand
CO4	**Explain** the role of people in Software Engineering	Level 2: Understand
CO5	**Analyze** the risk factors and project management in Software Development	Level 3: Apply
CO6	**Illustrate** the legal and business aspects of Software Engineering	Level 2: Understand

Mapping of course outcomes with program outcomes

COs	PROGRAMME OUTCOMES (PO)											
	K3 PO1	4 PO2	5 PO3	5 PO4	6 PO5	3 PO6	2 PO7	3 PO8	3 PO9	3 PO10	3 PO11	3 PO12
CO1	1	2	3	-	3	2	-	2	3	1	2	1
CO2	2	3	3	1	3	-	-	1	2	2	3	1
CO3	1	1	-	-	3	1	-	1	-	1	-	1
CO4	-	1	-	1	-	1	-	1	3	3	2	1
CO5	3	3	-	1	2	-	-	-	-	-	2	1
CO6	-	-	-	-	-	3	-	3	-	-	2	1

3/2/1: high/medium/low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	10	10	10
Understand	20	20	20
Apply	20	20	70
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain importance of different phases of software development
2. List out the advantages and disadvantages of Spiral model
3. Illustrate the advantages of Agile model over other models.
4. Describe waterfall model in detail.
5. Explain Agile model in detail

Course Outcome 2 (CO2):

1. List out the requirement specification methods
2. Describe different architecture styles
3. Identify different UML diagrams to do software design
4. Explain unit, integration and system testing in detail.
5. Differentiate different system models.
6. Develop an SRS for a MobileApp that does caller identification.

Course Outcome 3 (CO3):

1. Write short note on reliability metrics.

2. Illustrate the steps in software configuration management.
3. Explain the functioning of DevOps.

Course Outcome 4 (CO4):

1. Analyze the role of users and staff in problem solving.
2. Identify the people factors in human driven software engineering.
3. Discuss the responsibilities of each personnel in a project.

Course Outcome 5 (CO5):

1. Analyze the risk factors in a MobileApp project
2. Describe the phases of software project management
3. Examine different types of documents generated during the software development.
4. Explain the software maintenance methods.
5. Discuss the responsibilities of a project manager.

Course Outcome 6 (CO6):

1. Describe intellectual property in the context of software.
2. Summarize the data privacy policies in software engineering
3. Illustrate different business models with examples.

Model Question Paper

PART A (10*3=30)

(Each Question carries 3 Marks)

1. List out the advantages and disadvantages of Spiral model
2. Examine the importance of different phases of software development
3. Classify the software requirements.
4. Compare and contrast decision tree and decision table.
5. Summarize the objectives of testing.
6. Differentiate cohesion and coupling in software design.
7. Write short note on reliability metrics.
8. Analyze the role of users in problem definition.
9. Analyze the risk factors in a MobileApp project
10. Summarize the data privacy policies in software engineering

PART B (5*14=70)

11. Describe each phase is software development life cycle in detail.

OR

12. Explain Agile model in detail
13. Discuss the formal methods of requirement specification

OR

14. Demonstrate the format of an SRS with a suitable example.
15. Illustrate different architecture styles with neat diagrams

OR

16. Identify different UML diagrams to do software design
17. Illustrate the steps in software configuration management

OR

18. Discuss the responsibilities of each personnel in a project
19. Describe the phases of software project management

OR

20. Illustrate different business models with examples.

Syllabus

Module 1: Introduction to Software Engineering
Software Engineering -Introduction and Overview- Software Process Models.
Module 2: Feasibility Study, Requirements Analysis & specification
Feasibility Study. Requirements Analysis & specification, Modeling Techniques.SRS
Module 3: Software Design, Software Testing
Software design, System Models, Design Methods. System Architecture, Object Oriented Design. Software Testing -Testing Documentation and Help Facilities.
Module 4: Software Reliability, SCM, People and Software Engineering
Software Reliability, Testing and bug fixing Tools, Software Configuration Management, Software Project Management, People and Software Engineering.
Module 5: Software documentation, delivery & maintenance, SPM, legal & business aspects
Software documentation, delivery & maintenance, Software Project Management, Legal Aspects of Software Engineering, Business Aspects of Software Engineering.

Text Books

T1. Roger S. Pressman, Software Engineering: A practitioner's approach, 8th Edition (Indian Edition), McGraw Hill. 2019
T2. Rajib Mall, Fundamentals of Software Engineering, 5th Edition, Prentice Hall India. 2018

Reference Books

R1. Pankaj Jalote, An integrated approach to Software Engineering, 3rd Edition, Springer/Narosa.
R2. Ian Sommerville, Software Engineering, 10th Edition, Addison-Wesley
R3. Sunitha EV, Sarath KS, Software Development Life Cycle: Theory vs Practice, Jyothis Publishers, 2019.
R4. Pfleeger, Atlee, Software Engineering Theory and Practice, 4 edition, 2009, Pearson.
R5. Grady Booch, Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson.

Course Contents and Lecture Schedule

Sl.No	**Module 1: Introduction to Software Engineering**	**7hrs**
1.1	**Introduction to Software Engineering**- overview of the software development process, importance of each step in software development, Cases of software project failures of major companies.	1
1.2	**Software Process Models**- Waterfall model - phases, pros and cons of the model, application (detailed description of each phase is required)	2
1.3	**Prototyping model** - phases, pros and cons of the model, application	1
1.4	**Spiral model** - phases, pros and cons of the model, application	1
1.5	**Rapid Software Development**, Agile model - phases, pros and cons of the model, application	2
	Module 2: Feasibility Study, Requirements Analysis & specification	**9 hrs**
2.1	**Feasibility Study** - methods, feasibility report	1
2.2	**Requirements Analysis & specification** - importance of requirements, types of requirements, Requirement Analysis, modeling and specification steps.	1
2.3	**Scenarios and Use Cases** - case study	1
2.4	**Informal Methods of Specification** - advantages and disadvantages **Formal Methods of Specification -** Axiomatic and Algebraic specifications	2
2.5	**Modeling Techniques** for Requirements Analysis and Definition - DFD – ER Diagrams – Decision tables – Decision Trees	3
2.6	**Software Requirement Specification** - format, importance, fit and gap analysis	1
	Module 3: Software Design, Software Testing	**12 hrs**
3.1	**Software design** – Cohesion and Coupling, Usability, UI/UX design, System Models: Data-flow models, Semantic data models, Object models. Design Methods- object oriented and function oriented.	3
3.2	**System Architecture** - Architectural Styles, Software Considerations of System Architectures.	1

3.3	**Object Oriented Design** – overview of UML diagrams, Tools and Techniques, design reuse.	2
3.4	Web App and Mobile App design – responsive design.	1
3.5	**Software Testing** - Objectives of testing, Testing Principles. Functional and Structural testing	3
3.6	Generation of test data - Test Plan - Unit testing – Integration testing – System testing. Testing GUIs, Test reporting, Testing Documentation and Help Facilities.	1
3.7	Testing Object-Oriented Applications, Web Apps and Mobile Apps	1
	Module 4: Software Reliability, SCM, People and Software Engineering	**9 hrs**
4.1	**Software Reliability** - Reliability metrics, The development process Reviews, Different aspects of reliability Programming techniques, Testing and bug fixing Tools, Performance testing.	3
4.2	**Software Configuration Management** – steps, features and tools, case study with DevOps.	2
4.3	**People and Software Engineering** - Software Development Staff and roles, The importance of people in problem solving process: The Role of Users in Problem definition;	2
4.4	**Human driven software engineering**; The people factor – Multidisciplinary aspects; The team factor; The customer factor.	2
	Module 5: Software documentation, delivery & maintenance, SPM, legal & business aspects	**8 hrs**
5.1	**Software documentation, delivery & maintenance**, Categories of Documentation, categories of software products, software maintenance methods.	2
5.2	**Software Project Management** - phases of Project Management, Project plan, Risk analysis.	2
5.3	**Legal Aspects of Software Engineering-** Contracts and licenses, Software Copyright, Software Patents, Trade Secrets and Non-Disclosure Agreements, Privacy.	2
5.4	**Business Aspects of Software Engineering** - Business Models. Emerging Trends in Software Engineering	2

SEMESTER -4