Curriculum

of

Diploma Programme

in

Computer Science & Engineering

State Board of Technical Education (SBTE)
Bihar

Semester – III
Teaching & Learning Scheme

Board of Study	Course Codes	Course Titles	Teaching & Learning Scheme (Hours/Week)					
			Classroom Instruction (CI)		Lab Instruction (LI)	Notional Hours (TW+ SL)	Total Hours (CI+LI+TW+SL)	Total Credits (C)
			L	T				
	2418301	**Data structure and Algorithm** (CSE, AIML)	3	-	4	2	9	6
	2418302	**Operating System**	2	1	-	2	5	4
	2418303	**Discrete Structures**	2	1	-	2	5	4
	2418304	**Digital Electronics & Microprocessor**	3	-	4	2	9	6
	2418305	**Python Programming** (CE, CSE, AIML, ME, ME (Auto)., ELX, ELX (R), MIE, FTS, CRE, CHE, TE, CACDDM, GT)	3	-	4	2	9	6
	2400008	**Sports, Yoga and Meditation** (Common for All Programmes)	-	-	1	1	2	1
	2418306	**Summer Internship – I** (After 2nd Sem) (Common for All Programmes)	-	-	2	2	4	2
	2400111	**Principles of Management** (Non-exam course) (CE, AIML, AE, CHE, CSE, ME, ME (Auto), FTS, MIE)	1	-	-	-	1	1
Total			**14**	**2**	**15**	**13**	**44**	**30**

Legend:

CI: Classroom Instruction (Includes different instructional/implementation strategies i.e. Lecture (L), Tutorial (T), Case method, Demonstrations, Video demonstration, Problem based learning etc. to deliver theoretical concepts)

LI: Laboratory Instruction (Includes experiments/practical performances /problem-based experiences in laboratory, workshop, field or other locations using different instructional/Implementation strategies)

Notional Hours: Hours of engagement by learners, other than the contact hours for ensuring learning.

TW: Term work (includes assignments, seminars, micro projects, industrial visits, any other student activities etc.)

SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.

C: Credits = (1 x CI hours) + (0.5 x LI hours) + (0.5 x Notional hours)

Note: TW and SL have to be planned by the teacher and performed by the learner under the continuous guidance and feedback of teacher to ensure outcome of learning.

Semester - III

Assessment Scheme

Board of Study	Course Codes	Course Titles	Assessment Scheme (Marks)						Total Marks (TA+TWA+LA)
			Theory Assessment (TA)		Term work & Self-Learning Assessment (TWA)		Lab Assessment (LA)		
			Progressive Theory Assessment (PTA)	End Theory Assessment (ETA)	Internal	External	Progressive Lab Assessment (PLA)	End Laboratory Assessment (ELA)	
	2418301	**Data structure and Algorithm** (CSE, AIML)	30	70	20	30	20	30	200
	2418302	**Operating System**	30	70	20	30	-	-	150
	2418303	**Discrete Structures**	30	70	20	30	-	-	150
	2418304	**Digital Electronics & Microprocessor**	30	70	20	30	20	30	200
	2418305	**Python Programming** (CE, CSE, AIML, ME, ME (Auto)., ELX, ELX (R), MIE, FTS, CRE, CHE, TE, CACDDM, GT)	30	70	20	30	20	30	200
	2418306	**Summer Internship – I** (After 2nd Sem) (Common for All Programmes)	**-**	**-**	10	15	10	15	50
	2400008	**Sports, Yoga and Meditation** (Common for All Programmes)	-	-	10	-	6	9	25
	2400111	**Principles of Management** (Non-exam course) (CE, AIML, AE, CHE, CSE, ME, ME (Auto), FTS, MIE)	**25**	**-**	-	-	-	-	25
Total			**175**	**350**	**120**	**165**	**76**	**114**	**1000**

Legend:

PTA: Progressive Theory Assessment in class room (includes class test, mid-term test and quiz using online/offline modes)
PLA: Progressive Laboratory Assessment (includes process and product assessment using rating Scales and rubrics)
TWA: Term work & Self Learning Assessment (Includes assessment related to student performance in assignments, seminars, micro projects, industrial visits, self-learning, any other student activities etc.

Note:

- ETA & ELA are to be carried out at the end of the term/ semester.
- Term Work is to be done by the students under the guidance of internal faculty but its assessment will be done **internally (40%)** as well as **externally (60%)**. Assessment related to planning and execution of Term Work activities like assignment, micro project, seminar and self-learning is to be done by internal faculty (Internal Assessment) whereas assessment of output/product/ presentation related to these activities will be carried out by external faculty/expert (External Assessment). However, criteria of internal as well as external assessment may vary as per the requirement of respective course. For valid and reliable assessment, the internal faculty should prepare checklist & rubrics for these activities.

A) Course Code : 2418301 (T2418301/P2418301/S2418301)

B) Course Title : Data Structure and Algorithm (AIML, CSE)

C) Pre- requisite Course(s) : Programming with C

D) Rationale :

Data structures are ways of organizing and storing data to be accessed and manipulated efficiently. An algorithm is a set of instructions or procedures designed to solve a particular problem or accomplish a specific task. Selecting the appropriate data structures optimizes the performance of algorithms that operate on that data.

This course fosters students to select appropriate data structures and algorithms for a given problem so as to optimize the performance of the program and improve its overall efficiency.

E) Course Outcomes (COs): After the completion of the course, teachers are expected to ensure the accomplishment of following course outcomes by the learners. For this, the learners are expected to perform various activities related to three learning domains (Cognitive, Psychomotor and Affective) in classroom/ laboratory/ workshop/ field/ industry.

After completion of the course, the students will be able to-

CO-1 Analyze the efficiency of algorithm

CO-2 Implement operations on linear data structures

CO-3 Implement operations on non-linear data structures

CO-4 Apply different searching, sorting and hashing techniques to solve real world problems.

CO-5 Design efficient algorithms to solve the real-world problems.

F) Suggested Course Articulation Matrix (CAM):

Course Outcomes (COs)	Programme Outcomes (POs)							Programme Specific Outcomes* (PSOs)	
	PO-1 Basic and Discipline Specific Knowledge	**PO-2** Problem Analysis	**PO-3** Design/ Developmentof Solutions	**PO-4** EngineeringTools	**PO-5** Engineering Practices for Society, Sustainability and Environment	**PO-6** Project Management	**PO-7** Life Long Learning	**PSO-1**	**PSO-2**
CO-1	1	-	-	-	-	-	1		
CO-2	2	2	1	1	-	-	-		
CO-3	2	2	1	1	-	-	-		
CO-4	2	3	1	1	-	-	-		
CO-5	2	3	1	1	-	-	1		

Legend: High (3), Medium (2), Low (1) and No mapping (-)

***** PSOs will be developed by respective programme coordinator at institute level. As per latest NBA guidelines, formulating PSOs is optional

G) Teaching & Learning Scheme:

Board of Study	Course Code	Course Title	Scheme of Study (Hours/Week)					
			Classroom Instruction (CI)		Lab Instruction (LI)	Notional Hours (TW+ SL)	Total Hours (CI+LI+TW+SL)	Total Credits (C)
			L	T				
AIML	**2418301**	Data Structure and Algorithm	03	-	04	02	09	06

Legend:

CI: Classroom Instruction (Includes different instructional/implementation strategies i.e. Lecture (L), Tutorial (T), Case method, Demonstrations, Video demonstration, Problem based learning etc. to deliver theoretical concepts)

LI: Laboratory Instruction (Includes experiments/practical performances /problem-based experiences in laboratory, workshop, field or other locations using different instructional/Implementation strategies)

Notional Hours: Hours of engagement by learners, other than the contact hours for ensuring learning.

TW: Term Work (includes assignments, seminars, micro projects, industrial visits, any other student activities etc.)

SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.

C: Credits = (1 x CI hours) + (0.5 x LI hours) + (0.5 x Notional hours)

Note: TW and SL have to be planned by the teacher and performed by the learner under the continuous guidance and feedback of teacher to ensure outcome of learning.

H) Assessment Scheme:

Board of Study	Course Code	Course Title	Assessment Scheme (Marks)						Total Marks (TA+TWA+LA)
			Theory Assessment (TA)		Term Work & Self-Learning Assessment (TWA)		Lab Assessment (LA)		
			Progressive Theory Assessment (PTA)	End Theory Assessment (ETA)	Internal	External	Progressive Lab Assessment (PLA)	End Laboratory Assessment (ELA)	
	2418301	Data Structure and Algorithm	30	70	20	30	20	30	200

Legend:

PTA: Progressive Theory Assessment in class room (includes class test, mid-term test and quiz using online/offline modes)

PLA: Progressive Laboratory Assessment (includes process and product assessment using rating Scales and rubrics)

TWA: Term work & Self Learning Assessment (Includes assessment related to student performance in assignments, seminars, micro projects, industrial visits, self-learning, any other student activities etc.

Note:

- ETA & ELA are to be carried out at the end of the term/ semester.
- Term Work is to be done by the students under the guidance of internal faculty but its assessment will be done internally (40%) as well as externally (60%). Assessment related to planning and execution of Term Work activities like assignment, micro project, seminar and self-learning is to be done by internal faculty (Internal Assessment) whereas assessment of output/product/ presentation related to these activities will be carried out by external faculty/expert (External Assessment). However, criteria of internal as well as external assessment may vary as per the requirement of respective course. For valid and reliable assessment, the internal faculty should prepare checklist & rubrics for these activities.

I) Course Curriculum Detailing: This course curriculum detailing depicts learning outcomes at course level and session level and their attainment by the students through Classroom Instruction (CI), Laboratory Instruction (LI), Sessional Work (SW) and Self Learning (SL). Students are expected to demonstrate the attainment of Theory Session Outcomes (TSOs) and Lab Session Outcomes (LSOs) leading to attainment of Course Outcomes (COs) upon the completion of the course. While curriculum detailing, NEP 2020 related reforms like Green skills, Sustainability, Multidisciplinary aspects, Indian Knowledge System (IKS) and others must be integrated appropriately.

J) Theory Session Outcomes (TSOs) and Units: T2418301

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1a. Describe different data types in data structure.	**Unit-1.0 Fundamentals of Algorithms and its Analysis** 1.1. Data Types	**CO-1**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1b. Classify the types of data structure based on its characteristics *TSO 1c.* Calculate the complexity of a given algorithm in terms of time and space. *TSO 1d.* Determine the running time of an algorithm using the given notation *TSO 1e.* Determine the time complexity of recursive algorithm	System defines data types, User defined data types 1.2. Basic concept of data structure Linear data structure, Non-linear data structure , Abstract data types 1.3. Algorithm and its analysis - Introduction of algorithm, Runtime analysis of algorithm, Space Complexity of algorithm, Worst case analysis, Best case analysis, Average case analysis 1.4. Asymptotic Notation Big-O Notation, Omega- Ω Notation, Theta Notation 1.5 Time complexity of recursive algorithm Basic concept of recursion, Time complexity analysis using Master theorem	
TSO 2a. Create resizable arrays TSO 2b. Implement basic operations on arrays and string TSO 2c. Create linked lists that can dynamically allocate and deallocate memory TSO 2d. Identify the different types of Linked List TSO 2e. Implement basic operations on linked lists, such as insertion, deletion, and traversal. TSO 2f. Evaluate postfix and infix expression TSO 2g. Implement basic operation on stack such as insertion, deletion, and traversal TSO 2h. Implement basic operations on queue, such as insertion, deletion, and traversal. TSO 2i. Explain the use of Queue data structure for real-world problems. TSO 2j. Implement enqueue and dequeue operations	**Unit 2. Linear Data Structures** 2.1 Array and String Concept of arrays, Single and Multi-dimensional arrays, Dynamic arrays, Array operations, Time and space complexity of array operations, Introduction to string, String manipulation 2.2 Linked List Introduction to linked list, Singly Linked List, circular Linked List, Basic operation on Linked List: Traversing List, Insertion, deletion, and modification in Linked List 2.3 Stacks and Queue Introduction to Stack, Stack operations, Implementation of Stack using simple array, dynamic array, and Linked List, Application of stack for evaluating Infix or Postfix Expression, balancing the symbols, function calls, Introduction to Queue, Queue operations, Implementation of Queue using simple array, dynamic array, and Linked List, Application of Queue	**CO-2**
TSO 3a. Create Binary search tree (BST) for given data set TSO 3b. Find minimum/maximum or k^{th}- smallest element in tree TSO 3c. Performs different traversal order of tree TSO 3d. Create a heap(min/max) for given array data TSO 3e. Perform different operation on heap such as insertion and deletion of an element, TSO 3f. Represent the given graph using: Adjacency Matrix, Adjacency List, and Adjacency Set	**Unit 3. Non-linear Data Structure** 3.1 Tree - Basic terminologies: tree, Degree of a node, Degree of tree, level of node, Depth/height of tree, In-degree, Out-degree, Path, Ancestor & Descendent node - Types of trees: Binary Tree, Binary Search tree (BST), Balance tree, B-tree - Traversal of Binary tree: In order, pred order, post order traversal 3.2 Priority Queue and Heaps - Introduction to priority queue, Different operations in priority queue, Implementation of priority queue using BST - Basics of Min heap, Max heap, and Binary heap, Basic operation on Binary heap, Heapifying the elements of binary heap	**CO-3**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 3g. Perform graphs traversal using different methods TSO 3h. Find shortest path in various types of graphs TSO 3i. Evaluate minimum spanning tree of a graph using given algorithm	3.3 Graphs - Basics terminologies: Vertex and edge of graph, weighted and unweighted Graph, directed and undirected graph, Degree, in-degree and out-degree of a node (vertex), Articulation point - Graph representation: Adjacency Matrix, Adjacency List, Adjacency Set - Graph Traversal: BFS, DFS - Shortest Path in unweighted, weighted, and negative edge graph, Shortest Path algorithm in weighted graph [Dijkstra's], Shortest Path algorithm in negative edge graph [Bellman-Ford Algorithm] - Shortest Path algorithm in weighted directed graph [Floyd-Warshall algorithm] - Spanning tree in graph, Minimum Spanning tree algorithm: Prim's algorithm, Kruskal's algorithm	
TSO 4a. Develop algorithm for sorting a given dataset using the specified sorting method. TSO 4b. Explain the working of given searching method with an example TSO 4c. Develop an algorithm for searching an element a using binary search technique. TSO 4d. Perform basic operations of Hash Table TSO 4e. Apply Hash Tables to various data structures such as arrays, linked lists	**Unit 4. Sorting and Searching Techniques** 4.1 Sorting techniques: - bubble sort, selection sort, insertion sort, quicksort, merge sort 4.2 Searching techniques: Linear search, Binary search 4.3 Hash Table Introduction to Hash Table, Hash Function, Hash Collision resolution Techniques: Direct chaining, Open addressing	**CO-4**
TSO 5a. Apply Huffman coding algorithm for solving real world problems TSO 5b. Apply divide and conquer techniques to solve a problem TSO 5c. Explain the features of dynamic programming approaches TSO 5d. Find shortest path of a given graph using dynamic algorithm TSO 5e. Find longest common subsequence from given strings	**Unit 5. Algorithm Design Techniques** 5.1 Element of Greedy algorithm - Greedy choice property, Optimal substructure - Huffman coding algorithm 5.2 Divide and Conquer Techniques - Divide and Conquer Visualization 5.3 Dynamic Programming Approaches - Top-down and button-up Dynamic programming - Basics of Overlapping subproblem and Memorization techniques 5.4 Dynamic Programming Problem Longest common subsequence, Knapsack problem, Matrix chain multiplication	**CO-5**

Note: One major TSO may require more than one Theory session/Period.

K) Suggested Laboratory (Practical) Session Outcomes (LSOs) and List of Practical: P2418301

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 1.1. Find size of different data types	1.	a. Write Program to find size of different data types.	CO-1
LSO 2.1. Implement insertion and deletion operation on array. *LSO 2.2.* Implement different operations on given strings. *LSO 2.3.* Apply insertion, deletion, traversing over a singly linked list. *LSO 2.4.* Implement insertion, deletion, traversing over a circular linked list. *LSO 2.5.* Create stack using array and linked list *LSO 2.6.* Implement stack for the evaluation of given expression *LSO 2.7.* Implement enqueue and dequeue operations on Queue using array and linked list	2.	a. Write a program to insert an element in a given array. b. Write a program to delete an element from a given array c. Write a program to modify a character in a string d. Write a program to insert a node at beginning, mid, and end of a given singly linked list e. Write a program to insert a node at beginning, mid, and end of a given circular linked list f. Write a program using stack for a given expression evaluation. g. Write a program to perform enqueue and dequeue operations on Queue	CO-2
LSO 3.1. Develop program to create a tree *LSO 3.2.* Develop program to perform traversal operations on a given tree. *LSO 3.3.* Create a priority queue using heap *LSO 3.4.* Create a priority queue using BST *LSO 3.5.* Perform the following operations on the heap: a. Insert an element into the heap. b. Delete the root element (highest priority) from the heap. c. Retrieve the root element without removing it. d. Check if the heap is empty *LSO 3.6.* Develop program to perform following operation on a given Priority Queue: a. Enqueue of an element b. Dequeue of an element c. Find the element with highest priority d. Determine the size of Priority Queue e. Empty check of Priority Queue *LSO 3.7.* Develop program to detect a cycle in a given graph using DFS *LSO 3.8.* Develop program to find an articulation point in a given undirected graph.	3.	a. Write programs to perform in order pre order, and post order traversal on a tree. b. Write functions to perform the following operations on the heap: i. Insert an element into the heap. ii. Delete the root element (highest priority) from the heap. iii. Retrieve the root element without removing it. iv. Check if the heap is empty c. Write a program to perform following operation on a given Priority Queue: i. Enqueue of an element ii. Dequeue of an element iii. Find the element with highest priority iv. Determine the size of Priority Queue v. Empty check of Priority Queue d. Write programs to perform following operation on graph i. To detect a cycle in a given graph using DFS ii. To find an articulation point in a graph using DFS	CO-3

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
		iii. To find the shortest path between two given nodes using BFS e. Apply Bellman-Ford algorithm to find shortest path for a given negative edge graph. f. Apply Floyd-Warshall algorithm to find shortest path for a given weighted directed graph.	
LSO 4.1. Apply an insertion sort, selection sort, and bubble sort on a given unsorted array. *LSO 4.2.* Implement a quick sort on a given unsorted array. *LSO 4.3.* Implement a merge sort on a given unsorted array. *LSO 4.4.* Apply a counting sort on a given list of elements *LSO 4.5.* Write the steps to separate even and odd numbers for given array. *LSO 4.6.* Apply a binary search to search an element *LSO 4.7.* Write program to search an element which appears maximum number of times in given array. *LSO 4.8.* Create hash table data structure using array data structure. *LSO 4.9.* Perform the following operations on the hash table: a. Insert a key-value pair into the hash table. b. Retrieve the value associated with a given key from the hash table. c. Delete a key-value pair from the hash table. d. Check if a key exists in the hash table.	4	a. Develop a Program to: i. Apply insertion sort, quicksort, and merge sort on given dataset. ii. Apply binary search to find an element in given array. b. Write a program to create a hash table using array data structure. c. Write a program to perform the following operations on the hash table: i. Insert a key-value pair into the hash table. ii. Retrieve the value associated with a given key from the hash table. iii. Delete a key-value pair from the hash table. iv. Check if a key exists in the hash table.	CO-4
LSO 5.1. Find Longest common sequence in given string *LSO 5.2.* Find shortest path using Bellman-Ford algorithm for a given graph *LSO 5.3.* Apply divide and conquer method Find minimum and maximum value from a list of elements using.	5.	Develop Program to: i. Find Longest common sequence in given strings. ii. Find shortest path using Bellman-Ford algorithm for a given graph. iii. Find minimum and maximum value from n elements using divide and conquer method.	CO-5

L) Suggested Sessional Work and Self Learning: S2418301 Some sample suggested assignments, micro project and other activities are mentioned here for reference.

a. Assignments: Questions/Problems/Numerical/Exercises to be provided by the course teacher in line with the targeted COs.

b. Micro Projects:

1. Build a phonebook application that stores contacts using doubly linked list.
2. Implement an algorithm to solve a Sudoku puzzle.
3. Build a spell checker that suggests corrections for misspelled words
4. Implement the Huffman coding algorithm to compress and decompress text files
5. Create a calculator that uses a stack data structure to evaluate expressions.

c. Seminar topics:

1. Scope of Data Structure and Algorithm in real world.
2. Height balance tree
3. Comparative analysis of given sorting methods

M) Suggested Course Evaluation Matrix: The course teacher has to decide and use the appropriate assessment strategy and its weightage in theory, laboratory and sessional work for ensuring CO attainment. The response/performance of the student in each of these designed activities is to be used to calculate **CO attainment**.

COs	Course Evaluation Matrix						
	Theory Assessment (TA)**		Term Work Assessment (TWA)			Lab Assessment (LA)#	
	Progressive Theory Assessment (PTA) Class/Mid Sem Test	End Theory Assessment (ETA)	Term Work & Self Learning Assessment			Progressive Lab Assessment (PLA)	End Laboratory Assessment (ELA)
			Assignments	Micro Projects	Other Activities*		
CO-1	15%	15%	15%	20%	20%	5%	5%
CO-2	20%	20%	20%	20%	20%	20%	20%
CO-3	25%	25%	25%	20%	20%	25%	25%
CO-4	20%	20%	20%	20%	20%	25%	25%
CO-5	20%	20%	20%	20%	20%	25%	25%
Total Marks	**30**	**70**	**20**	**20**	**10**	**20**	**30**
			50				

Legend:

*: Other Activities include self- learning, seminar, visits, surveys, product development, software development etc.

**: Mentioned under point- (N)

#: Mentioned under point-(O)

Note:

- The percentage given are approximate
- In case of Micro Projects and End Laboratory Assessment (ELA), the achieved marks will be equally divided in all those COs mapped with total experiments.
- For CO attainment calculation indirect assessment tools like course exit survey need to be used which comprises of questions related to achievement of each COs.

N) Suggested Specification Table for End Semester Theory Assessment: Specification table represents the reflection of sample representation of assessment of cognitive domain of full course.

Unit Title and Number	Total Classroom Instruction (CI) Hours	Relevant COs Number(s)	Total Marks	ETA (Marks)		
				Remember (R)	Understanding (U)	Application & above (A)
Unit-1. Fundamentals of algorithms and its analysis	8	CO-1	10	3	3	4
Unit 2. Linear Data Structures	10	CO-2	14	4	4	6
Unit 3. Non-linear data structure	10	CO-3 and CO-4	18	5	3	10

Unit 4. Sorting and Searching Techniques	12	CO-5	14	4	3	7
Unit 5. Algorithm Design Techniques	8	CO-6	14	4	4	6
Total	**48**	**-**	**70**	**20**	**17**	**33**

Note: Similar table can also be used to design class/mid-term/ internal question paper for progressive assessment.

O) Suggested Assessment Table for Laboratory (Practical):

S. No.	Laboratory Practical Titles	Relevant COs Number(s)	PLA /ELA		
			Performance		Viva-Voce (%)
			PRA* (%)	PDA** (%)	
1.	Write Program to find size of different data types.	CO-1	30	60	10
2.	a. Write a program to insert an element in a given array. b. Write a program to delete an element from a given array c. Write a program to modify a character in a string d. Write a program to insert a node at beginning, mid, and end of a given singly linked list e. Write a program to insert a node at beginning, mid, and end of a given circular linked list f. Write a program using stack for a given expression evaluation. g. Write a program to perform enqueue and dequeue operations on Queue	CO-2	30	60	10
3.	a. Write programs to perform in order pre order, and post order traversal on a tree. b. Write functions to perform the following operations on the heap: v. Insert an element into the heap. vi. Delete the root element (highest priority) from the heap. vii. Retrieve the root element without removing it. viii. Check if the heap is empty c. Write a program to perform following operation on a given Priority Queue: vi. Enqueue of an element vii. Dequeue of an element viii. Find the element with highest priority ix. Determine the size of Priority Queue x. Empty check of Priority Queue d. Write programs to perform following operation on graph iv. To detect a cycle in a given graph using DFS v. To find an articulation point in a graph using DFS vi. To find the shortest path between two given nodes using BFS e. Apply Bellman-Ford algorithm to find shortest path for a given negative edge graph. f. Apply Floyd-Warshall algorithm to find shortest path for a given weighted directed graph.	CO-3	30	60	10

S. No.	Laboratory Practical Titles	Relevant COs Number(s)	PLA /ELA		
			Performance		Viva-Voce (%)
			PRA* (%)	PDA** (%)	
4.	a. Develop a Program to: i. Apply insertion sort, quicksort, and merge sort on given dataset. ii. Apply binary search to find an element in given array. b. Write a program to create a hash table using array data structure. c. Write a program to perform the following operations on the hash table: i. Insert a key-value pair into the hash table. ii. Retrieve the value associated with a given key from the hash table. iii. Delete a key-value pair from the hash table. iv. Check if a key exists in the hash table.	CO-4	30	60	10
5.	Develop Program to: i. Find Longest common sequence in given strings. ii. Find shortest path using Bellman-Ford algorithm for a given graph. iii. Find minimum and maximum value from n elements using divide and conquer method.	CO-5	30	60	10

A) Course Code : **2418302 (T2418302/S2418302)**

B) Course Title : Operating System

C) Pre- requisite Course(s) :

D) Rationale :

To work with an application on a computer system, an operating system is required which provides a platform to run applications and manage systems activities. An Operating System is basically a system program that controls the execution of application programs and acts as an interface between applications and the computer hardware. It manages the computer system resources to be used in an efficient manner. This course enables to learn internal functioning of operating system and will help in identifying appropriate Operating System for given applications/task.

E) Course Outcomes (COs): After the completion of the course, teachers are expected to ensure the accomplishment of following course outcomes by the learners. For this, the learners are expected to perform various activities related to three learning domains (Cognitive, Psychomotor, and Affective) in classroom/ laboratory/ workshop/ field/ industry. The theory, practical experiences, and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following industry-oriented COs associated with the above-mentioned competency:

After completion of the course, the students will be able to-

- **CO-1** Enumerate the types and functions of operating systems.
- **CO-2** Explain the process and inter process communication.
- **CO-3** Analyze issues related to CPU scheduling and deadlocks.
- **CO-4** Illustrate the concept of Memory management and virtual memory.
- **CO-5** Illustrate the concept of File management.

F) Suggested Course Articulation Matrix (CAM):

Course Outcomes (COs)	Programme Outcomes(POs)							Programme SpecificOutcomes* (PSOs)	
	PO-1 Basic and Discipline Specific Knowledge	**PO-2** Problem Analysis	**PO-3** Design/ Developmentof Solutions	**PO-4** Engineering Tools	**PO-5** Engineering Practices for Society, Sustainability and Environment	**PO-6** Project Management	**PO-7** Life Long Learning	**PSO-1**	**PSO-2**
CO-1	2	-	-	-	-	-	1		
CO-2	1	2	-	-	-	-	2		
CO-3	1	2	3	-	1	-	2		
CO-4	2	2	1	-	-	-	1		
CO-5	2	2	1	-	1	2	2		

Legend: High (3), Medium (2), Low (1) and No mapping (-)

***** PSOs will be developed by respective programme coordinator at institute level. As per latest NBA guidelines, formulating PSOs is optional

G) Teaching & Learning Scheme:

Board of Study	Course Code	Course Title	Scheme of Study (Hours/Week)					
			Classroom Instruction (CI)		Lab Instruction (LI)	Notional Hours (TW+ SL)	Total Hours (CI+LI+TW+SL)	Total Credits (C)
			L	T				
Computer Science & Engineering	**2418302**	Operating System	02	01	-	02	05	04

Legend:

CI: Classroom Instruction (Includes different instructional/implementation strategies i.e. Lecture (L), Tutorial (T), Case method, Demonstrations, Video demonstration, Problem based learning etc. to deliver theoretical concepts)

LI: Laboratory Instruction (Includes experiments/practical performances /problem-based experiences in laboratory, workshop, field or other locations using different instructional/Implementation strategies)

Notional Hours: Hours of engagement by learners, other than the contact hours for ensuring learning.

TW: Term Work (includes assignments, seminars, micro projects, industrial visits, any other student activities etc.)

SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.

C: Credits = (1 x CI hours) + (0.5 x LI hours) + (0.5 x Notional hours)

Note: TW and SL have to be planned by the teacher and performed by the learner under the continuous guidance and feedback of teacher to ensure outcome of learning.

H) Assessment Scheme:

Board of Study	Course Code	Course Title	Assessment Scheme (Marks)						Total Marks (TA+TWA+LA)
			Theory Assessment (TA)		Term Work & Self-Learning Assessment (TWA)		Lab Assessment (LA)		
			Progressive Theory Assessment (PTA)	End Theory Assessment (ETA)	Internal	External	Progressive Lab Assessment (PLA)	End Laboratory Assessment (ELA)	
Computer Science & Engineering	**2418302**	Operating System	30	70	20	30	-	-	150

Legend:

PTA: Progressive Theory Assessment in class room (includes class test, mid-term test and quiz using online/offline modes)

PLA: Progressive Laboratory Assessment (includes process and product assessment using rating Scales and rubrics)

TWA: Term work & Self Learning Assessment (Includes assessment related to student performance in assignments, seminars, micro projects, industrial visits, self-learning, any other student activities etc.

Note:

- ETA & ELA are to be carried out at the end of the term/ semester.
- Term Work is to be done by the students under the guidance of internal faculty but its assessment will be done **internally (40%)** as well as **externally (60%)**. Assessment related to planning and execution of Term Work activities like assignment, micro project, seminar and self-learning is to be done by internal faculty (Internal Assessment) whereas assessment of output/product/ presentation related to these activities will be carried out by external faculty/expert (External Assessment). However, criteria of internal as well as external assessment may vary as per the requirement of respective course. For valid and reliable assessment, the internal faculty should prepare checklist & rubrics for these activities.

I) Course Curriculum Detailing: This course curriculum detailing depicts learning outcomes at course level and session level and their attainment by the students through Classroom Instruction (CI), Laboratory Instruction (LI), Term Work (TW) and Self Learning (SL). Students are expected to demonstrate the attainment of Theory Session Outcomes (TSOs) and Lab Session Outcomes (LSOs) leading to attainment of Course Outcomes (COs) upon the completion of the course. While curriculum detailing, NEP 2020 related reforms like Green skills, Sustainability, Multidisciplinary aspects, Society connect, Indian Knowledge System (IKS) and others must be integrated appropriately.

J) Theory Session Outcomes (TSOs) and Units: T2418302

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO.1.a Explain concepts and role as system software of an operating system. TSO.1.b Explain major functions of an operating system TSO.1.c Explain the different views of an operating. TSO.1.d Identify various types of operating systems and their characteristics. TSO.1.e. Explain the concept of system Calls.	**Unit-1: Operating System Concepts-** 1.1 Operating System – Concept, Components of OS, System Software 1.2 Functions of O.S : Program Management, Resource management, File Management, Device Management, Security and protection. 1.3Views of OS: User view, System View 1.4 Types of Operating Systems and their characteristics: Batch operating system, Multi Programming, Time Shared OS, Multiprocessing OS, Distributed OS, Real-time systems, Mobile OS. 1.5 Services of Operating System. 1.6 System Calls- Concept, types of system calls	**CO-1**
TSO.2.a Explain functions carried out in the given process state. TSO.2.b Justify the need of PCB with relevant example. TSO.2.c Explain the process of inter process communication with example. TSO.2.d Explain characteristics of the given multithreading model.	**Unit-2: Process Management** 2.1 Process-: process states, Process Control Block (PCB). 2.2 Process Scheduling- Scheduling Queues, Schedulers, Context switch. 2.3 Inter-process communication (IPC) : Introduction, shared memory system & message passing system. 2.4 Threads - Benefits, users and kernel threads, Multithreading Models - Many to One, One to One, Many to Many.	**CO-2**
TSO.3.a Justify the need and objective of job scheduling with relevant example. TSO.3.b Explain the procedure of allocation of CPU to a process with example. TSO.3.c Calculate turnaround time and average waiting time of the given scheduling algorithm. TSO.3.d Explain the given necessary condition leading to deadlock.	**Unit-3: CPU Scheduling and Algorithm** 3.1 Scheduling types – scheduling Objectives, CPU and I/O burst cycles, Pre-emptive, Non- Pre-emptive Scheduling, Scheduling criteria. 3.2 Types of Scheduling algorithms - First come first served (FCFS), Shortest Job First (SJF), Shortest Remaining Time(SRTN), Round Robin (RR) Priority scheduling, multilevel queue scheduling. 3.3 Deadlock - System Models, Necessary Conditions leading to Deadlocks, Deadlock Handling - Preventions, avoidance.	**CO-3**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO.4.a Justify the need of memory management. TSO.4.b Explain characteristic of the given memory management techniques. TSO.4.c Write algorithm for the given page replacement technique. TSO.4.d Calculate Page fault for the given page reference string.	**Unit- 4: Memory Management** 4.1 Basic Memory Management - Partitioning, Fixed and Variable, Free Space management Techniques - Bitmap, Linked List. 4.2 Virtual Memory – Introduction to Paging, Segmentation, Fragmentation, and Page fault. 4.3 Page Replacement Algorithms: FIFO, LRU, Optimal.	**CO-4**
TSO.5.a Explain the structure of the given file system with example. TSO.5.b Describe mechanism of the given file access method. TSO.5.c Explain procedure to create and access directories and assign the given files access permissions.	**Unit-5: File Management** 6.1 File – Concepts, Attributes, Operations, types and File System Structure. 6.2 Access Methods – Sequential, Direct, Swapping, File Allocation Methods- Contiguous, Linked, Indexed. 6.3 Directory structure-– Single level, two levels, tree-structured directory, Disk Organization and disk Structure- Physical structure, Logical structure.	**CO-5**

Note: One major TSO may require more than one Theory session/Period.

A) Course Code : **2418303(T2418303/S2418303)**

B) Course Title : Discrete Structure

C) Pre- requisite Course(s) : Applied Mathematics- B

D) Rationale :

Discrete structures are important in computer science because they provide a framework for modelling and solving real-world problems. By using discrete structures, complex problems can be broken down into simpler components which are easier to analyze and comprehend. This makes it possible to develop efficient algorithms for solving problems, as well as to design computer programs. In addition, discrete structure is essential for understanding computer science, as it provides the theoretical foundations for many areas such as cryptography, game theory, artificial intelligence, data structures, algorithms, and software engineering. Logic and proof techniques are essential tools for reasoning about the correctness of algorithms and programs. Sets and relations are used in databases and programming languages. Combinatorics is used to analyze the efficiency of algorithms, estimate the complexity of problems, and develop optimization strategies. Graph theory is used in a wide range of applications including computer networking, optimization, and scheduling. Fuzzy logic is a mathematical framework for dealing with uncertainty, vagueness, and imprecision in data. It is also increasingly important in many areas of computer science, including artificial intelligence and machine learning.

E) Course Outcomes (COs): After the completion of the course, teachers are expected to ensure the accomplishment of following course outcomes by the learners. For this, the learners are expected to perform various activities related to three learning domains (Cognitive, Psychomotor and Affective) in classroom/laboratory/workshop/field/ industry.

After completion of the course, the students will be able to-

- **CO-1** Demonstrate proficiency in recognizing and applying various logic and proof techniques for engineering applications.
- **CO-2** Apply the concepts of set theory, relations and their application in modeling computer science engineering-based problems.
- **CO-3** Apply combinatorial principles to solve branch specific problems.
- **CO-4** Use graph theoretic principles to solve computer science engineering related problems.
- **CO-5** Solve computer science engineering-based problems using the basics of fuzzy set theory.

F) Suggested Course Articulation Matrix (CAM):

Course Outcomes (COs)	Programme Outcomes (POs)							Programme Specific Outcomes* (PSOs)	
	PO-1 Basic and Discipline Specific Knowledge	**PO-2** Problem Analysis	**PO-3** Design/ Development of Solutions	**PO-4** Engineering Tools	**PO-5** Engineering Practices for Society, Sustainability and Environment	**PO-6** Project Management	**PO-7** Life Long Learning	**PSO-1**	**PSO-2**
CO-1	3	2	2	1	1	-	1		
CO-2	3	1	-	-	-	-	-		
CO-3	2	2	-	-	1	-	1		
CO-4	3	2	2	-	1	1	1		
CO-5	2	1	1	-	-	-	-		

Legend: High (3), Medium (2), Low (1) and No mapping (-)

* PSOs will be developed by respective programme coordinator at institute level. As per latest NBA guidelines, formulating PSOs is optional

G) Teaching & Learning Scheme:

Board of Study	Course Code	Course Title	Scheme of Study (Hours/Week)					
			Classroom Instruction (CI)		Lab Instruction (LI)	Notional Hours (TW+ SL)	Total Hours (CI+LI+TW+SL)	Total Credits (C)
			L	T				
Computer Science Engineering	**2418303**	Discrete structure	02	01	-	02	05	04

Legend:

CI: Classroom Instruction (Includes different instructional/implementation strategies i.e. Lecture (L), Tutorial (T), Case method, Demonstrations, Video demonstration, Problem based learning etc. to deliver theoretical concepts)

LI: Laboratory Instruction (Includes experiments/practical performances /problem-based experiences in laboratory, workshop, field or other locations using different instructional/Implementation strategies)

Notional Hours: Hours of engagement by learners, other than the contact hours for ensuring learning.

TW: Term Work (includes assignments, seminars, micro projects, industrial visits, any other student activities etc.)

SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.

C: Credits = (1 x CI hours) + (0.5 x LI hours) + (0.5 x Notional hour

Note: TW and SL have to be planned by the teacher and performed by the learner under the continuous guidance and feedback of teacher to ensure outcome of learning.

H) Assessment Scheme:

Board of Study	Course Code	Course Title	Assessment Scheme (Marks)						Total Marks (TA+TWA+LA)
			Theory Assessment (TA)		Term Work & Self-Learning Assessment (TWA)		Lab Assessment (LA)		
			Progressive Theory Assessment (PTA)	End Theory Assessment (ETA)	Internal	External	Progressive Lab Assessment (PLA)	End Laboratory Assessment (ELA)	
Computer Science Engineering	**2418303**	Discrete structure	30	70	20	30	-	-	150

Legend:

PTA: Progressive Theory Assessment in class room (includes class test, mid-term test and quiz using online/offline modes)

PLA: Progressive Laboratory Assessment (includes process and product assessment using rating Scales and rubrics)

TWA Term work & self-learning Assessment (Includes assessment related to student performance in assignments, seminars, micro projects, industrial visits, self-learning, any other student activities etc.

Note:

- ETA & ELA are to be carried out at the end of the term/ semester.
- Term Work is to be done by the students under the guidance of internal faculty but its assessment will be done **internally (40%)** as well as **externally (60%)**. Assessment related to planning and execution of Term Work activities like assignment, micro project, and seminar and self-learning is to be done by internal faculty (Internal Assessment) whereas assessment of output/product/ presentation related to these activities will be carried out by external faculty/expert (External Assessment). However, criteria of internal as well as external assessment may vary as per the requirement of respective course. For valid and reliable assessment, the internal faculty should prepare checklist & rubrics for these activities.

I) Course Curriculum Detailing: This course curriculum detailing depicts learning outcomes at course level and session level and their attainment by the students through Classroom Instruction (CI), Laboratory Instruction (LI), Term Work (TW) and Self Learning (SL). Students are expected to demonstrate the attainment of Theory Session Outcomes (TSOs) and Lab Session Outcomes (LSOs) leading to attainment of Course Outcomes (COs) upon the completion of the course. While curriculum detailing, NEP 2020 related reforms like Green skills, Sustainability, Multidisciplinary aspects, Society connect, Indian Knowledge System (IKS) and others must be integrated appropriately.

J) Theory Session Outcomes (TSOs) and Units: T2418303

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1a. Identify difference between propositional and predicate logic. *TSO 1b.* Use logical equivalences concept to simply compound statements. *TSO 1c.* Express the given statement using "predicate logic". *TSO 1d.* Apply basic proof techniques to prove mathematical statements. *TSO 1e.* Use nested quantifiers to express complex statements. *TSO 1f.* Use the concept of Mathematical induction to prove the given statement.	**Unit-1.0 Logic and Proof Techniques** 1.1 Propositional logic: Connectives and Truth Tables, Tautologies and Contradictions, Logical Equivalences. 1.2 Predicate logic: Quantifiers, Nested Quantifiers, Inference rules for predicate logic. 1.3 Mathematical proofs: Basic proof techniques: Direct proofs, Proof by contrapositive, proof by contradiction and Proof by mathematical induction	**CO1**
TSO 2a. Use set theoretic operations to solve given problems. *TSO 2b.* Use De Morgan's Law to simplify expressions for applied problems. *TSO 2c.* Identify different types of relations. *TSO 2d.* Determine the Domain and range of a Relation *TSO 2e.* Use equivalence relations to solve given problems.	**Unit-2.0 Set Theory and Relation** 2.1 Set and subsets. 2.2 Operations on sets. 2.3 Venn diagrams and De Morgan's law. 2.4 Relations and their properties. 2.5 Equivalence relation.	**CO2**
TSO 3a. Apply fundamental counting principle to solve counting problems. *TSO 3b.* Differentiate between Permutations and Combinations on the basis of given applied problems and then solve. *TSO 3c.* Apply permutations and combinations to solve problems based on arranging letters in a word for practical applications. *TSO 3d.* Apply the pigeonhole principle to solve combinatorial problems. *TSO 3e.* Use binomial theorem to solve problems involving binomial coefficients and powers. *TSO 3f.* Solve counting problems using generating functions.	**Unit-3.0 Combinatorics** 3.1 Basics counting principles. 3.2 Permutations and Combinations. 3.3 Pigeonhole principle (without proof and its application. 3.4 Binomial theorem. 3.5 Generating functions.	**CO3**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 4a. Explain different types of graphs and calculate the degree of a vertex. *TSO 4b.* Identify the isomorphic graphs. *TSO 4c.* Define walks, paths, and cycles in graphs. *TSO 4d.* Apply Eulerian graphs and their applications to solve given problems. *TSO 4e.* Calculate the connectivity of a graph and identify its components.	**Unit- 4.0 Graph Theory** 4.1 Basic concepts and definition. 4.2 Types of Graph and degree of vertex. 4.3 Sub graph and Isomorphic Graphs. 4.4 Walks, Paths, Cycle. 4.5 Eulerian Graph (without proof) and its application. 4.6 Connectivity and Components	**CO4**
TSO 5a. Differentiate between classical set theory and fuzzy set theory. *TSO 5b.* Use the concept of membership functions and degrees of membership to applied problems. *TSO 5c.* Define the concept of fuzzy propositions and truth values.	**Unit-5.0 Introduction to Fuzzy Set Theory** 5.1 Basics of Fuzzy set theory. 5.2 Membership functions and degrees of membership. 5.3 Fuzzy set theoretic operations. 5.4 Fuzzy propositions and truth values.	**CO5**

Note: One major TSO may require more than one Theory session/Period.

K) Suggested Laboratory (Practical)/ Tutorials and Outcomes:

Outcomes	S. No.	Laboratory (Practical)/ Tutorials Titles	Relevant COs Number(s)
LSO 1.1. Verify the statement "An object is either in motion or it is not in motion." is a Tautology, prove it? *LSO 1.2.* Use the concept of "proof by contradiction", to establish the uniqueness of the solution. *LSO 1.3.* Verify the correctness of the circuit design of the given Boolean equation using predicate logic. *LSO 1.4.* Verify the correctness of the given algorithm using "contrapositive". *LSO 1.5.* Apply principle of mathematical induction to prove given mathematical statements.	1.	• Analysis of moving object using "Tautology". • Uniqueness of solutions by contradiction. • Verification of circuit design correctness using predicate logic. • Verification using "contra-positiveness". • Application of Mathematical induction.	**CO1**
LSO 2.1. Represent different operations on given sets by Venn Diagram. *LSO 2.2.* Prove De Morgan's law geometrically and interpret the result. *LSO 2.3.* Apply Equivalence relations for equivalence partioning to check validity.	2.	• Operations on set. • Geometrical interpretation of De Morgan's Law. • Applications of Equivalence relation.	**CO2**
LSO 3.1. Apply counting techniques to count the number of possible outcomes in given algorithms or programs. *LSO 3.2.* Count the number of possible routes for a delivery driver. *LSO 3.3.* Count the number of possible configurations of a network. *LSO 3.4.* Find duplicate entries in a database using	3.	• Applications of counting techniques. • Applications of Pigeonhole Principle. • Binomial theorem and its applications. • Applications of generating functions.	**CO3**

Outcomes	S. No.	Laboratory (Practical)/ Tutorials Titles	Relevant COs Number(s)
Pigeonhole Principle. *LSO 3.5.* Use Pigeonhole Principle to find the minimum number of items required to guarantee that at least two of them share a certain property. *LSO 3.6.* Write programs in a programming language to implement the binomial theorem for computing binomial coefficients by expanding binomial expressions. *LSO 3.7.* Compute generating functions using Mathematica or Python.			
LSO 4.1. Use graph theory to Schedule the tasks such that the overall time to complete all the tasks is minimized. *LSO 4.2.* Use Graph theory to allocate the resources such that all constraints are satisfied and the overall cost is minimized. *LSO 4.3.* Find the shortest path between two routers, considering the cost of each link. *LSO 4.4.* Use graph theory algorithm in Internet Routing.	4.	• Applications of graph theory for minimization problems. • Applications of graph theory for shortest path problems. • Graph theory and algorithm.	**CO4**
LSO 5.1. Use fuzzy set theory to Predict final grade of the student using input variables such as attendance, homework scores and exam score. *LSO 5.2.* Create membership functions and define fuzzy sets using built-in functions or libraries.	5.	• Applications of fuzzy set theory. • Applications of membership functions	**CO5**

A) Course Code : 2418304 (T2418304/P2418304/S2418304)

B) Course Title : Digital Electronics and Microprocessor

C) Pre- requisite Course(s) : Fundamentals of Electrical and Electronics Engineering

D) Rationale :

Currently, most of the state-of-art electronic equipment like mobiles, computers, ATM, TV, music system, air conditioners, automobiles are embedded with digital circuits; and in fact, microprocessor is called as the heart of a computer. The ICs used in any electronic equipment needs continuous monitoring for their proper upkeep. For this work, knowledge and skills related with logic gates, combinational circuits, sequential circuits, data converters and memory are a must for diploma engineers. This course is meant to provide the basic skills to use and solve the application problems based on digital integrated circuits and microprocessor. In addition, this course will enable the students to inculcate assembly language programming concepts and also help to develop hardware related projects.

E) Course Outcomes (COs): After the completion of the course, teachers are expected to ensure the accomplishment of following course outcomes by the learners. For this, the learners are expected to perform various activities related to three learning domains (Cognitive, Psychomotor and Affective) in classroom/ laboratory/ workshop/ field/ industry.

After completion of the course, the students will be able to-

- **CO-1** Minimize the Boolean expressions and implement it using logic gates.
- **CO-2** Test simple combinational and sequential circuits.
- **CO-3** Use data converters and memory in digital electronic systems.
- **CO-4** Develop simple assembly language programs for various operations using instruction set of 8085 microprocessor.
- **CO-5** Interface the memory and I/O devices to 8085 microprocessor.

F) Suggested Course Articulation Matrix (CAM):

Course Outcomes (COs)	Programme Outcomes (POs)							Programme Specific Outcomes* (PSOs)	
	PO-1 Basic and Discipline Specific Knowledge	**PO-2** Problem Analysis	**PO-3** Design/ Development of Solutions	**PO-4** Engineering Tools	**PO-5** Engineering Practices for Society, Sustainability and Environment	**PO-6** Project Management	**PO-7** Life Long Learning	**PSO-1**	**PSO-2**
CO-1	3	1	-	-	-	1	1		
CO-2	3	-	2	1	-	1	1		
CO-3	3	-	2	-	-	1	1		
CO-4	3	2	3	1	-	1	-		
CO-5	3	-	2	-	-	1	2		

Legend: High (3), Medium (2), Low (1) and No mapping (-)

***** PSOs will be developed by respective programme coordinator at institute level. As per latest NBA guidelines, formulating PSOs is optional.

G) Teaching & Learning Scheme:

Board of Study	Course Code	Course Title	Teaching & Learning Scheme (Hours/Week)					
			Classroom Instruction (CI)		Lab Instruction (LI)	Notional Hours (TW+ SL)	Total Hours (CI+LI+TW+SL)	Total Credits (C)
			L	T				
Electronics Engineering	**2418304**	Digital Electronics and Microprocessor	03	-	04	02	09	06

Legend:

CI: Classroom Instruction (Includes different instructional/implementation strategies i.e. Lecture (L), Tutorial (T), Case method, Demonstrations, Video demonstration, Problem based learning etc. to deliver theoretical concepts)

LI: Laboratory Instruction (Includes experiments/practical performances /problem-based experiences in laboratory, workshop, field or other locations using different instructional/Implementation strategies)

Notional Hours: Hours of engagement by learners, other than the contact hours for ensuring learning.

TW: Term Work (includes assignments, seminars, micro projects, industrial visits, any other student activities etc.)

SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.

C: Credits = (1 x CI hours) + (0.5 x LI hours) + (0.5 x Notional hours)

Note: TW and SL have to be planned by the teacher and performed by the learner under the continuous guidance and feedback of teacher to ensure outcome of learning.

H) Assessment Scheme:

Board of Study	Course Code	Course Title	Assessment Scheme (Marks)						
			Theory Assessment (TA)		Term Work & Self Learning Assessment (TWA)		Lab Assessment (LA)		Total Marks (TA+TWA+LA)
			Progressive Theory Assessment (PTA)	End Theory Assessment (ETA)	Internal	External	Progressive Lab Assessment(PLA)	End Laboratory Assessment (ELA)	
Electronics Engineering	**2418304**	Digital Electronics and Microprocessor	30	70	20	30	20	30	200

Legend:

PTA: Progressive Theory Assessment in class room (includes class test, mid-term test and quiz using online/offline modes)PLA: Progressive Laboratory Assessment (includes process and product assessment using rating Scales and rubrics)

TWA: Term work & Self Learning Assessment (Includes assessment related to student performance in assignments, seminars, micro projects, industrial visits, self learning, any other student activities etc.

Note:

- ETA & ELA are to be carried out at the end of the term/ semester.
- Term Work is to be done by the students under the guidance of internal faculty but its assessment will be done **internally (40%)** as well as **externally (60%)**. Assessment related to planning and execution of Term Work activities like assignment, micro project, seminar and self-learning is to be done by internal faculty (Internal Assessment) whereas assessment of output/product/ presentation related to these activities will be carried out by external faculty/expert (External Assessment). However criteria of internal as well as external assessment may vary as per the requirement of respective course. For valid and reliable assessment, the internal faculty should prepare checklist & rubrics for these activities.

I) **Course Curriculum Detailing:** This course curriculum detailing depicts learning outcomes at course level and session level and their attainment by the students through Classroom Instruction (CI), Laboratory Instruction (LI), Term Work (TW) and Self Learning (SL). Students are expected to demonstrate the attainment of Theory Session Outcomes (TSOs) and Lab Session Outcomes (LSOs) leading to attainment of Course Outcomes (COs) upon the completion of the course. While curriculum detailing, NEP 2020 related reforms like Green skills, Sustainability, Multidisciplinary aspects, Society connect, Indian Knowledge System (IKS) and others must be integrated appropriately.

J) Theory Session Outcomes (TSOs) and Units: T2418304

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1a. Explain the given number system. *TSO 1b.* Convert a given number in any number system into another specified number system. *TSO 1c.* Perform the specific arithmetic operation with respect to given number(s) in a given number system. *TSO 1d.* Determine 1's and 2's complement of given binary number. *TSO 1e.* Represent negative number in 1's and 2's complement. *TSO 1f.* Use 1's and 2's complement for subtraction. *TSO 1g.* Minimize the given Boolean expression using Boolean algebra and K-map. *TSO 1h.* Realize the logical expression using logic gates.	**Unit-1.0 Number Systems, Boolean Algebra and Logic Gates** 1.1 Different number systems: • Binary, Octal, Decimal, Hexadecimal • Conversion from one number system to another number systems. 1.2 Arithmetic operation of Binary, Octal, Hexadecimal number systems. 1.3 Complements: 1's and 2's complement. 1.4 Data Representation: • Representation of negative number in 1's and 2's complement • Subtraction using 1's and 2's complement 1.5 Boolean Algebra: • Rules and laws of Boolean Algebra • De-Morgan's Theorem 1.6 Standard Boolean Representation: • Sum of Product (SOP) • Product of Sum (POS) 1.7 Minimization: • Karnaugh's Map (K-map) up to three variables • Simplification of Boolean expressions using Boolean laws and K-map. 1.8 Logic Gates and applications: • AND, OR, NOT, Buffer, NAND, NOR, XOR, XNOR (Symbol, Truth table, Logic expression and its applications) 1.9 Implementation of Boolean expressions using basic gates	**CO1**
TSO 2a. Develop simple arithmetic circuits using logic gates. *TSO 2b.* Implement multiplexer and de-multiplexer using logic gates. *TSO 2c.* Use encoder and decoder in digital circuits. *TSO 2d.* Differentiate combinational and sequential circuits. *TSO 2e.* Explain the ripple counter for up/down sequence with block diagram.	**Unit-2.0 Combinational and Sequential Logic Circuits** 2.1 Arithmetic Circuits: • Half Adder and Full Adder • Half Subtractor and Full Subtractor 2.2 Multiplexer: • 2 to 1 MUX • 4 to 1 MUX • Applications	**CO1, CO2**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 2f. Differentiate synchronous and asynchronous counter. *TSO 2g.* Explain the ring counter with block diagram	2.3 De-multiplexer: • 1 to 2 DEMUX • 1 to 4 DEMUX • Applications 2.4 Encoder and Decoder 2.5 Flip-Flops : SR, JK, T, D, and JK, Master Slave JK flip-flop 2.6 Shift Registers: • Serial In Serial Out • Serial In Parallel Out • Parallel In Serial Out • Parallel In Parallel Out 2.7 Counters: • Modulus of counter • Asynchronous Counter: Ripple up/down counter • Synchronous Counter: Ring Counter	
TSO 3a. Calculate the output voltage of given Op-amp circuit. *TSO 3b.* Explain the DAC and ADC. *TSO 3c.* Compare various type of memory in terms of its functionality. *TSO 3d.* List the memory chip.	**Unit-3.0 Data Converters and Memory Devices** 3.1 Data Converters: • Op-Amp: Introduction (Inverting and Non inverting) • Digital to analog and Analog to digital converter: Uses 3.2 Random Access Memory: Introduction and its types 3.3 Read Only Memory: Introduction and its types	**CO3**
TSO 4a. Interpret the general-purpose microprocessor. *TSO 4b.* Explain the architecture of 8085 microprocessor with block diagram. *TSO 4c.* Explain various types of interrupts. *TSO 4d.* Classify the different types of instruction used in 8085. *TSO 4e.* Differentiate addressing modes of 8085 microprocessor. *TSO 4f.* Differentiate addressing modes of 8085 microprocessor. *TSO 4g.* Use various types of instruction to write simple Assembly Language Program.	**Unit-4.0 Basics, Instruction Set and Programming of 8085 Microprocessor** 4.1 Basics of Microprocessor: • Evolution of Microprocessors • Architecture and Pin diagram of 8085 • Timing Diagram and Memory Organization • Interrupts 4.2 Instruction Set: • Data Transfer Instructions • Control instructions • Arithmetic instructions • Logical instructions • Branching instructions 4.3 Different types of Addressing Modes: • Immediate Addressing Mode • Register Addressing Mode • Direct Addressing Mode • Indirect Addressing Mode • Indexed Addressing Mode 4.4 Assembly Language Programming	**CO4**
TSO 5a. Interface Intel PPI 8255 with 8085.	**Unit-5.0 Interfacing with 8085 Microprocessor:**	**CO4, CO5**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 5b. Interface various memory chips with 8085 microprocessors. *TSO 5c.* Explain the operation of interfacing chips. *TSO 5d.* Differentiate between the serial and parallel communication modes of 8085 microprocessor.	5.1 Programmable Peripheral Interface (PPI)- Intel 8255 (Generation of I/O Ports) 5.2 Programmable Interval timers (Intel 8253/8254) 5.3 Overview of Memory chips and their interfaces 5.4 Overview of other interfacing chips (Name and Application(s) only)	

Note: One major TSO may require more than one theory session/period.

K) Suggested Laboratory (Practical) Session Outcomes (LSOs) and List of Practical: P2418304

Practical/Lab Session Outcomes(LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 1.1 List the IC number of different types of logic gates. *LSO 1.2* Verify the truth table of identified logic gate IC.	1.	Test the functionality of logic gates using ICs.	CO1
LSO 2.1 Build the circuit on breadboard for making AND gate using NOR gate. *LSO 2.2* Verify the truth table of the developed AND gate. *LSO 2.3* Build the circuit on breadboard similarly for other gates using NOR gate. *LSO 2.4* Verify the truth table of the developed gate.	2.	Implement logic gates using universal NAND gate IC only.	CO1
LSO 3.1 Build the circuit on breadboard for making AND gate using NOR gate. *LSO 3.2* Verify the truth table of the developed AND gate. *LSO 3.3* Build the circuit on breadboard similarly for other gates using NOR gate. *LSO 3.4* Verify the truth table of the developed gate.	3.	Implement logic gates using universal NOR gate IC only.	CO1
LSO 4.1 Build the circuit of Half Adder using basic gates on breadboard. *LSO 4.2* Test the functionality of Half Adder. *LSO 4.3* Build the circuit of Half Subtractor on breadboard. *LSO 4.4* Test the functionality of Half Subtractor.	4.	Implement Half Adder and Half Subtractor using basic gates.	CO2
LSO 5.1 Build the circuit of Full Adder using basic gates on breadboard. *LSO 5.2* Check the result of binary addition on the developed circuit.	5.	Implement Full Adder using basic gates.	CO2
LSO 6.1 Build the circuit of Full Subtractor using NOR gate on breadboard. *LSO 6.2* Check the result of binary subtraction on the developed circuit.	6.	Implement Full Subtractor using basic gates.	CO2
LSO 7.1 Build the circuit connection of multiplexer on trainer kit. *LSO 7.2* Test whether the particular input line is available at output for given data select line.	7.	Test the functionality of multiplexer on trainer kit.	CO2
LSO 8.1 Build the circuit connection of De-multiplexer. *LSO 8.2* Test whether the given data available at input is distributed correctly to output for	8.	Test the functionality of de-multiplexer on trainer kit.	CO2

Practical/Lab Session Outcomes(LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
given data select line.			
LSO 9.1 Build the circuit of SR flip-flop using NAND gate on breadboard. *LSO 9.2* Verify the characteristic table of SR flip-flop.	9.	Verify the function of SR flip-flop using NAND gate.	CO2
LSO 10.1 Build the circuit of SR flip-flop using NOR gate on breadboard. *LSO 10.2* Verify the characteristic table of SR flip-flop.	10.	Verify the function of SR flip-flop using NOR gate.	CO2
LSO 11.1 Construct the circuit diagram of D flip-flop on breadboard. *LSO 11.2* Test the functionality of D flip-flop.	11.	Test the functionality of D flip-flop using IC 7476.	CO2
LSO 12.1 Construct the circuit diagram of T flip-flop on breadboard. *LSO 12.2* Test the functionality of T flip-flop.	12.	Test the functionality of T flip-flop using IC 7476.	CO2
LSO 13.1 List the IC number of DAC. *LSO 13.2* Test its functionality.	13.	Test the functionality of DAC using IC.	CO3
LSO 14.1 List the IC number of ADC. *LSO 14.2* Test its functionality.	14.	Test the functionality of ADC using IC.	CO3
LSO 15.1 Examine the 8085 Trainer kit. *LSO 15.2* Identify the various components in 8085 Trainer Kit.	15.	Test and verify the features of 8085 Trainer Kit.	CO4, CO5
LSO 16.1 Write an assembly language program based on Data transfer Instructions & Arithmetic Instructions. *LSO 16.2* Test the results by executing the assembly language program.	16.	Write and execute an ALP for 8085 to add two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations.	CO4
LSO 17.1 Write an assembly language program based on Data transfer Instructions & Arithmetic Instructions. *LSO 17.2* Test the results by executing the assembly language program.	17.	Write and execute an ALP for 8085 to Subtract two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations.	CO4
LSO 18.1. Develop an assembly language program to interface 7 segment display with 8051 Microcontroller *LSO 18.2.* Test the results by executing the assembly language program.	18.	Develop a program to interface 7 segment display with 8051.	CO5

7. Develop and Execute an 8085 Assembly language programme to alternatively blink LEDs connected on 8255 port at an interval of 0.1 second. Build the circuit.

c. Other Activities:

1. Seminar Topics:
   - Biometric voting machine
   - Night vision technology
   - Digital locker
   - Barcodes Reader

2. Visits: Visit nearby radio station/industry/ electronic shops. Prepare report of visit with special comments of digital electronics component/batch production/mass production and cost of component.

3. Self- learning topics:
   - PCB design technique
   - Key board encoder
   - 2-bit comparator
   - Carry look ahead adder
   - Self-complimentary code like 2421, 3321

M) Suggested Course Evaluation Matrix: The course teacher has to decide and use appropriate assessment strategy and its weightage in theory, laboratory and Term Work for ensuring CO attainment. The response/performance of each student in each of these designed activities is to be used to calculate **CO attainment**.

COs	Course Evaluation Matrix						
	Theory Assessment (TA)**		Term Work Assessment (TWA)			Lab Assessment (LA)#	
	Progressive Theory Assessment (PTA) Class/Mid Sem Test	End Theory Assessment (ETA)	Term Work & Self-Learning Assessment			Progressive Lab Assessment (PLA)	End Laboratory Assessment (ELA)
			Assignments	Micro Projects	Other Activities*		
CO-1	15%	10%	15%	-	20%	15%	20%
CO-2	25%	20%	25%	25%	20%	30%	20%
CO-3	15%	30%	15%	25%	20%	20%	20%
CO-4	30%	25%	30%	25%	20%	25%	20%
CO-5	15%	15%	15%	25%	20%	10%	20%
Total Marks	**30**	**70**	**20**	**20**	**10**	**20**	**30**
			50				

Legend:

*: Other Activities include self- learning, seminar, visits, surveys, product development, software development etc.

**: Mentioned under point- (N)

#: Mentioned under point-(O)

Note:

- The percentage given are approximate
- In case of Micro Projects and End Laboratory Assessment (ELA), the achieved marks will be equally divided in all those COs mapped with total experiments.
- For CO attainment calculation indirect assessment tools like course exit survey need to be used which comprises of questions related to achievement of each COs.

N) Suggested Specification Table for End Semester Theory Assessment: Specification table represents the reflection of sample representation of assessment of cognitive domain of full course.

Unit Title and Number	Total Classroom Instruction (Cl) Hours	Relevant COs Number(s)	Total Marks	ETA (Marks)		
				Remember (R)	Understanding (U)	Application & above (A)
Unit-1.0 Number Systems, Boolean Algebra and Logic Gates	8	CO1	13	4	4	5
Unit-2.0 Combinational and Sequential Logic Circuits	10	CO1, CO2	16	4	7	5
Unit-3.0 Data Converters and Memory Devices	8	CO3	12	4	4	4
Unit-4.0 Basics, Instruction Set and Programming of 8085 Microprocessor	14	CO4	18	4	8	6
Unit-5.0 Interfacing with 8085 Microprocessor	8	CO4, CO5	11	3	4	4
Total	**48**	**-**	**70**	**20**	**26**	**24**

Note: Similar table can also be used to design class/mid-term/ internal question paper for progressive assessment.

O) Suggested Assessment Table for Laboratory (Practical):

S. No.	Laboratory Practical Titles	Relevant COs Number (s)	PLA/ELA		
			Performance		Viva-Voce (%)
			PRA* (%)	PDA** (%)	
1.	Test the functionality of given logic gates using ICs.	CO1	30	60	10
2.	Implement logic gates using universal NAND gate IC.	CO1	40	50	10
3.	Implement logic gates using universal NOR gate IC.	CO1	40	50	10
4.	Implement Half Adder and Half Subtractor using basic gates.	CO2	30	60	10
5.	Implement Full Adder using basic gates.	CO2	40	50	10
6.	Implement Full Subtractor using basic gate.	CO2	40	50	10
7.	Test the functionality of multiplexer on trainer kit.	CO2	20	70	10
8.	Test the functionality of de-multiplexer on trainer kit.	CO2	40	50	10
9.	Verify the function of SR flip-flop using NAND gate.	CO2	20	70	10
10.	Verify the function of SR flip-flop using NOR gate.	CO2	40	50	10
11.	Test the functionality of D flip-flop using IC 7476.	CO2	40	50	10
12.	Test the functionality of T flip-flop using IC 7476.	CO2	40	50	10

S. No.	Laboratory Practical Titles	Relevant COs Number (s)	PLA/ELA		
			Performance		Viva-Voce (%)
			PRA* (%)	PDA** (%)	
13.	Test the functionality of DAC using IC.	CO3	30	60	10
14.	Test the functionality of ADC using IC.	CO3	30	60	10
15.	Test and verify the features of 8085 Trainer Kit.	CO4, CO5	30	60	10
16.	Write and execute an ALP for 8085 to add two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations.	CO4	40	50	10
17.	Write and execute an ALP for 8085 to Subtract two 8-bit Nos. which is stored at two different memory locations and store the result (with carry & without carry cases) at another memory locations.	CO4	40	50	10
18.	Develop a program to interface 7 segment display with 8051.	CO5	40	50	10

Legend :
PRA*: Process Assessment
PDA**: Product Assessment

Note: This table can be used for both end semester as well as progressive assessment of practical. Rubrics need to beprepared by the course teacher for each experiment/practical to assess the student performance.

P) Suggested Instructional/Implementation Strategies: Different Instructional/ Implementation Strategies may be appropriately selected, as per the requirement of the content/outcome. Some of them are Improved Lecture, Tutorial, Case Method, Group Discussion, Industrial visits, Industrial Training, Portfolio Based Learning, Role Play, Live Demonstrations in Classrooms, Lab, Field, Information and Communications Technology (ICT) Based Teaching Learning, Blended or flipped mode, Brainstorming, Expert Sessions, Video Clippings, Use of Open Educational Resources (OER), MOOCs etc.

Q) List of Major Laboratory Equipment, Tools and Software:

S. No.	Name of Equipment, Tools, and Software	Broad Specifications	Relevant Experiment/Practical Number
1.	Oscilloscope	Dual Channel 20MHz	All
2.	Function generator	100MHz Function & Arbitrary Generator, 500MSa/s-DG4102	All
3.	Digital IC Trainer Kits	Power Supply: +5V, +/- 12V Display Type: 2 Digit BCD to Decimal Display	All
4.	Logic Gates ICs	Two input and 3-Input	1 to 6
5.	Bread Board	MB 102 Breadboard with Power Supply Module, Jumper Wires, Battery Clip,830 & 400 tie-Points	All
6.	Digital Multimeter	DM-86 Digital Multimeter AC Frequency Response: 40-400Hz Low Battery Display: Approx. < 7.5V	All

S. No.	Name of Equipment, Tools, and Software	Broad Specifications	Relevant Experiment/Practical Number
7.	IC Tester	• Package: Digital ICs of 14, 16, 18,20,24,28 & 40 pins dual in line. • Range: Tristate, Open Collector & Bidirectional TTL/CMOS ICs. • Method: Truth table comparison. • Sockets: 20 and 40 pin ZIF. • Keyboard: 24 feather touch keys. • Display: 16 digit 0.5" Seven segment LED display. • Voltage: 230 volts + 10% 50Hz, AC.	All
8.	Microprocessor Trainer Kit	Single board systems with 8K RAM, ROM memory with battery backup, 16X4,16 X2, LCD display, PC keyboard interfacing facility, Hex keypad facility, single user cross c-compiler, RS- 232, USB, interfacing facility with built in power supply.	15,16, 17,18
9.	Keyboard Trainer Board	Keyboard 4*4 trainer board	Term work
10.	7-segment LED Display	7-segment LED Display: -0.56 in 1-digit, common anode/common cathode	18
11.	Display Trainer Board	LCD trainer board	Term work
12.	Trainer Boards for DAC & ADC	DAC (0808) trainer board, ADC (0808) trainer board	13, 14

R) Suggested Learning Resources:

(a) Books:

S. No.	Titles	Author(s)	Publisher and Edition with ISBN
1.	Digital principles & Applications	Albert Paul Malvino & Donald P. Leach	McGraw Hill Education; Eighth edition, ISBN: 978- 9339203405
2.	Digital Electronics, Principles and Applications	Roger L. Tokheim	McGraw-Hill Education (ISE Editions); International 2nd revised edition, ISBN: 978-0071167963
3.	Digital Electronics – An Introduction to Theory and Practice	William H. Gothmann	Prentice Hall India Learning Private Limited; 2nd edition ISBN: 978-8120303485
4.	Fundamentals of Logic Design	Charles H. Roth & Larry L. Kinney	Jaco Publishing House; First edition, ISBN: 978-8172247744
5.	Digital Electronics	R. Anand	Khanna Publications, New Delhi, (Edition 2018), ISBN: 978-93-82609445
6.	8085 Microprocessor	Ramesh S. Gaonkar	5th Edition, Prentice Hall ISBN: 0130195707
7.	Fundamentals of Microprocessor & Microcontroller	B. Ram	Dhanpat Rai & Sons Pub., 3rd edition, 2008, ISBN: 978-8189928605

A) Course Code : **2418305(T2418305, P2418305,S2418305)**

B) Course Title : Python Programming

(CE, CSE, AIML, ME, ME (Auto)., ELX, ELX (R), MIE, FTS, CRE, CHE, TE, CACDDM, GT, RE)

C) Pre- requisite Course(s) :

D) Rationale :

Python programming has emerged as a popular programming language across wide range of application segments from Scientific to Machine Learning to mobile app development, and so on. Python is a high-level general-purpose programming language.

Because code is automatically compiled to byte code and executed, Python is suitableuse as a scripting language, Web application implementation language, etc.
In Python there are multiple levels of organizational structure: functions, classes, modules, and packages. These assist in organizing code. An excellentand large example is the Python standard library.

The Object-oriented Python provides a consistent way to use objects: in Python it is easy to implement new object types (called classesin object-oriented programming).
This introductory course to learn basic Python programming features which can be used as building blocks to develop different kind of applications using Python 3.

E) Course Outcomes (COs): After the completion of the course, teachers are expected to ensure the accomplishment of following course outcomes by the learners. For this, the learners are expected to perform various activities related to three learning domains (Cognitive, Psychomotor and Affective) in classroom/ laboratory/ workshop/ field/ industry.

After completion of the course, the students will be able to-

CO-1 Use various data types and operators in formation of expressions.
CO-2 Write and execute programs using control statements.
CO-3 Perform relevant operations on Sequence data types
CO-4 Create functions in modules
CO-5 Use numpy in writing python programs
CO-6 Handle data files and exceptions.

F) Suggested Course Articulation Matrix (CAM):

Course Outcomes (COs)	Programme Outcomes(POs)							Programme Specific Outcomes* (PSOs)	
	PO-1 Basic and Discipline Specific Knowledge	**PO-2** Problem Analysis	**PO-3** Design/ Developmentof Solutions	**PO-4** Engineering Tools	**PO-5** Engineering Practices for Society, Sustainability and Environment	**PO-6** Project Management	**PO-7** Life Long Learning	**PSO-1**	**PSO-2**
CO-1	1	-	1	-	-	-	-		
CO-2	1	2	2	1	-	1	-		
CO-3	1	2	2	1	-	1	-		
CO-4	1	2	2	1	-	1	2		
CO-5	1	2	2	1	-	1	-		
CO-6	1	2	2	1	-	1	1		

Legend: High (3), Medium (2), Low (1) and No mapping (-)

* PSOs will be developed by respective programme coordinator at institute level. As per latest NBA guidelines, formulating PSOs is optional

G) Teaching & Learning Scheme:

Board of Study	Course Code	Course Title	Scheme of Study (Hours/Week)					
			Classroom Instruction (CI)		Lab Instruction (LI)	Notional Hours (TW+ SL)	Total Hours (CI+LI+TW+SL)	Total Credits (C)
			L	T				
	2418305	Python programming	03	-	04	02	09	06

Note: Prefix will be added to Course Code if applicable (T for theory Paper, P for Practical Paper and S for Term work)

Legend:

CI: Classroom Instruction (Includes different instructional/implementation strategies i.e. Lecture (L), Tutorial (T), Case method, Demonstrations, Video demonstration, Problem based learning etc. to deliver theoretical concepts)

LI: Laboratory Instruction (Includes experiments/practical performances /problem-based experiences in laboratory, workshop, field or other locations using different instructional/Implementation strategies)

Notional Hours: Hours of engagement by learners, other than the contact hours for ensuring learning.

TW: Term Work (includes assignments, seminars, micro projects, industrial visits, any other student activities etc.)

SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.

C: Credits = (1 x CI hours) + (0.5 x LI hours) + (0.5 x Notional hours)

Note: TW and SL have to be planned by the teacher and performed by the learner under the continuous guidance and feedback of teacher to ensure outcome of learning.

H) Assessment Scheme:

Board of Study	Course Code	Course Title	Assessment Scheme (Marks)						Total Marks (TA+TWA+LA)
			Theory Assessment (TA)		Term Work & Self-Learning Assessment (TWA)		Lab Assessment (LA)		
			Progressive Theory Assessment (PTA)	End Theory Assessment (ETA)	Internal	External	Progressive Lab Assessment (PLA)	End Laboratory Assessment (ELA)	
	2418305	Python programming	30	70	20	30	20	30	200

Note: Prefix will be added to Course Code if applicable (T for theory Paper, P for Practical Paper and S for Term work)

Legend:

PTA: Progressive Theory Assessment in class room (includes class test, mid-term test and quiz using online/offline modes)

PLA: Progressive Laboratory Assessment (includes process and product assessment using rating Scales and rubrics)

TWA: Term work & Self Learning Assessment (Includes assessment related to student performance in assignments, seminars, micro projects, industrial visits, self-learning, any other student activities etc.

Note:

- ETA & ELA are to be carried out at the end of the term/ semester.
- Term Work is to be done by the students under the guidance of internal faculty but its assessment will be done **internally (40%)** as well as **externally (60%)**. Assessment related to planning and execution of Term Work activities like assignment, micro project, seminar and self-learning is to be done by internal faculty (Internal Assessment) whereas assessment of output/product/ presentation related to these activities will be carried out by external faculty/expert (External Assessment). However, criteria of internal as well as external assessment may vary as per the requirement of respective course. For valid and reliable assessment, the internal faculty should prepare checklist & rubrics for these activities.

I) Course Curriculum Detailing: This course curriculum detailing depicts learning outcomes at course level and session level and their attainment by the students through Classroom Instruction (CI), Laboratory Instruction (LI), Term Work (TW) and Self Learning (SL). Students are expected to demonstrate the attainment of Theory Session Outcomes (TSOs) and Lab Session Outcomes (LSOs) leading to attainment of Course Outcomes (COs) upon the completion of the course. While curriculum detailing, NEP 2020 related reforms like green skills, Sustainability, Multidisciplinary aspects, Society connect, Indian Knowledge System (IKS) and others must be integrated appropriately.

J) Theory Session Outcomes (TSOs) and Units: T2418305

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1a. Differentiate between Procedure Oriented P and Object Oriented Programming approach with example. *TSO 1b.* Use the concept of Lvalue and Rvalue *TSO 1c.* Write python program using various data types and operators	Unit 1: Fundamentals of Python Programming Syntax 1.1 Introduction to Python Character Set, Python Tokens, Variables, Lvalue and Rvalue Concepts, and the Use of Comments. 1.2Overview of Data Types: • Number Types: Integer, Floating Point, Complex • Boolean Type • Sequence Types: String, List, Tuple • None Type • Mapping Type: Dictionary • Distinction between Mutable and Immutable Data Types 1.3 Understanding Operators: • Arithmetic Operators • Relational Operators • Logical Operators • Assignment Operator • Augmented Assignment Operators • Expressions and Statements • Type Conversion and Input/Output Mechanisms • Precedence of Operators • Expression Evaluation	**CO-1**
TSO 2a. Write Python program using decision making statements *TSO 2b.* Write Python program using loop structure to solve iterative problems	**Unit-2.0 Conditional and Iterative statements** 2.1 Conditional statements: • simple if statement • if- else statemen • if-elif-else statement 2.2 Iterative statements: • while loop • for loop • range function • break and continue statements • nested loops	**CO-2**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 3a. Perform various operations on string using string operators and methods *TSO 3b.* Perform various operations on List using list operators and methods *TSO 3c.* Perform various operations on tuples using tuples operators and methods *TSO 3d.* Perform various operations on set using set methods *TSO 3e.* Perform various operations on dictionary using dictionary methods	**Unit-3.0 String, List, Tuples, set and Dictionary** 3.1 String: • Indexing • string operations (concatenation, repetition, membership & slicing) • traversing a string using loops • built-in functions. 3.2 Lists: • Introduction • Indexing in list • list operations: concatenation, repetition, membership & slicing, traversing a list, built- in list functions, linear search on list of numbers and counting the frequency of elements in a list 3.3 Tuples: Creating, initializing, accessing elements, tuple assignment, performing operations on tuples, tuple methods and built-in functions, nested tuples 3.4 Set: Creating set, traversing, adding, removing data in set, performing set operations like join, Union intersection, difference 3.5 Dictionary: accessing items in a dictionary using keys, mutability of dictionary: adding a new item, modifying an existing item, built-in dictionary functions.	**CO-3**
TSO 4a. Create and use user defined functions to implement modular programming approach *TSO 4b.* Differentiate variable scope with example. *TSO 4c.* Import and use Python modules, libraries	**Unit-4.0 Python Functions, Modules and packages** 4.1 Functions: types of function (built- in functions, functions defined in module, user defined functions), creating user defined function, arguments and parameters, default parameters, positional parameters, Lambda functions, returning value, scope of a variable: global scope, local scope 4.2 Modules and Packages: Importing module using 'import' Regular Expressions, Exception Handling, PyPI Python Package Index, Pip Python package manager, Importing Libraries and Functions	**CO-4**
TSO 5a. Write simple Python programs using numpy *TSO 5b.* Use Numpy array in python program *TSO 5c.* Use Numpy to solve linear algebra problem.	**Unit-5.0 Numpy** 5.1 Introduction to NumPy 5.2 Installation of NumPy 5.3 NumPy Arrays: • Understanding the NumPy array	**CO-5**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
	• The fundamental data structure in NumPy. • Creation of arrays using different methods: np.array(), np.zeros(), np.ones(), etc. • Exploring array attributes like shape, size, and dimensions. 5.4 Array Indexing and Slicing: • Accessing elements and subarrays in NumPy arrays using indexing and slicing. • Demonstration of the difference between one-dimensional and multi-dimensional array indexing. 5.5 Array Operations: • Performing element-wise operations on NumPy arrays. • Exploring universal functions (ufuncs) for mathematical operations. 5.6 Linear Algebra with NumPy: • Introduction to linear algebra operations using NumPy. • Matrix multiplication, determinant, inverse, and solving linear equations. 5.7 File input and output with Numpy 5.8 Broadcasting in Numpy	
TSO 6a. Explain different types of Exceptions in python *TSO 6b.* Write Python programs for exception handling in Python *TSO 6c.* Differentiate different modes of file opening. *TSO 6d.* Perform read, Write, Append operations in files	**Unit 6: Exception and File Handling in Python** 6.1 Exception Handling: syntax errors, exceptions, need of exception handling, user-defined exceptions, raising exceptions, handling exceptions, catching exceptions, Try - except - else clause, Try - finally clause, recovering and continuing with finally, built-in exception classes. 6.2 File Handling: text file and binary file, file types, open and close files, reading and writing text files, reading and writing binary files, file access modes	**CO-6**

Note: One major TSO may require more than one Theory session/Period.

K) Suggested Laboratory (Practical) Session Outcomes (LSOs) and List of Practical: P2418305

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 1.1. Write, execute and debug simple Python program using Integrated Development and Learning Environment (IDLE) *LSO 1.2.* Write and execute simple 'C' program using variables, arithmetic expressions.	1.	a) Download and Install IDLE. Write and execute Python program to- b) Calculate the Area of a Triangle where its three sides a, b, c are given. s=(a+b+c)/2, Area=square root of s(s-a)(s-b)(s-c) (write program without using function) c) Swap Two Variables d) Solve quadratic equation for real numbers.	**CO-1**
LSO 2.1. Write and execute python programs using conditional statements. *LSO 2.2.* Write and execute python programs using various types of Loop statements	2.	Write and execute Python program to- a) Check if a Number is Positive, Negative or zero. b) Check whether the given year is a Leap Year. c) Print all Prime Numbers in an Interval. d) Display the multiplication Table based on the given input. e) Print the Fibonacci sequence. f) Find the Factorial of a Number.	**CO-2**
LSO 3.1. Write and execute Python program to perform various operations on string using string operators and methods	3.	Write and execute Python program to- a) Check whether the string is Palindrome b) Reverse words in a given String in Python c) identify in a strings the name, position and counting of vowels. d) Count the Number of matching characters in a pair of string (set) e) Python program for removing i-th character from a string	**CO-2, CO-3**
LSO 4.1. Write and execute Python program to perform various operations on List using List operators and methods	4.	Write and execute Python program to- a) find largest number in a given list without using max(). b) find the common numbers from two lists. c) create a list of even numbers and another list of odd numbers from a given list. d) To find number of occurrences of given number without using built-in methods.	**CO-2, CO-3**
LSO 5.1. Write and execute Python program to perform various operations on Tuple using Tuple operators and methods.	5.	Write and execute Python program to- a) find the index of an item of a tuple. b) find the length of a tuple. c) to reverse a tuple. d) Write a Python program to sort a list of tuple by its float element. Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')] Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]	**CO-2, CO-3**

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 6.1. Write and execute Python program to perform various operations on sets using set methods.	6.	Write and execute Python program to- a) create an intersection of sets. b) create a union of sets. c) create set difference. d) check if two given sets have no elements in common.	**CO-2, CO-3**
LSO 7.1. Write and execute Python program to perform various operations on Dictionary using Dictionary methods	7.	Write and execute Python program to- a) Write a Python script to concatenate two dictionaries to create a new one b) Write a Python script to merge two Python dictionaries. c) Write a Python program to combine two dictionary adding values for common keys. d1 = {'a': 100, 'b': 200, 'c':300} d2 = {'a': 300, 'b': 200, 'd':400} Sample output: d({'a': 400, 'b': 400, 'd': 400, 'c': 300})	**CO-2, CO-3**
LSO 8.1. Write and execute Python program to create user defined functions and call them.	8.	Write and execute Python program to- a) Write a Python function for reversing a string and call it. b) Write a Python function for calculating compound interest and call it. c) Write a Python function for calculating the factorial of a number and call it to calculate !n/(!r)*!(n-r)) where symbol "**!** " stands for factorial.	**CO-2, CO-4**
LSO 9.1. Write and execute Python program to define a numpy array. *LSO 9.2.* Develop and execute Python program Using various types of Numpy operation.	9.	a) Write a python program to create a Numpy array filled with all zeros b) Write a python program to check whether a Numpy array contains a specified row c) Write a python program to Remove rows in Numpy array that contains non-numeric values d) Write a python program to Find the number of occurrences of a sequence in a NumPy array e) Write a python program to Find the most frequent value in a NumPy array f) Write a python program to Combine a one and a two-dimensional NumPy Array g) Write a python program to Flatten a Matrix in Python using NumPy h) Write a python program to Interchange two axes of an array	**CO-2, CO-5**
LSO 10.1. Develop and execute Python program to handle various type of exceptions. *LSO 10.2.* Develop and execute Python program to perform file operations.	10.	a) Using exception handling feature such as try…except, try finally- write minimum three programs to handle following types of exceptions. i. Type Error ii. Name Error	**CO-6, CO-1, CO-2,**

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
		iii. Index Error iv. Key Error v. Value Error vi. IO Error vii. Zero Division Error b) Write Python program to demonstrate file operations.	

Note: in addition to above listed practical, students are suggested to practice all the examples covered by the teacher during theory sessions.

L) Suggested Term Work and Self Learning: **S2418305** Some sample suggested assignments, micro project and other activities are mentioned here for reference.

a. **Assignments:** Questions/Problems/Numerical/Exercises to be provided by the course teacher in line with the targeted COs.

b. Micro Projects:

1. Crete a shop billing system
2. Create income tax calculation system.
3. Develop number guessing game (random integer will be selected by the system and the user has to guess that integer in the minimum number of guesses. Maximum 5 guess allowed.)
4. Assign numbers to alphabet a-z as (1-26). User will input a word. System will convert in to a number by adding all the individual alphabet of that word.
5. Design a basic calculator program that performs arithmetic operations like addition, subtraction, multiplication, and division based on user input.
6. Any other micro-projects suggested by subject faculty on similar line.

(Students may use file and sequence data types to develop above listed applications)

c. Other Activities:

1. Seminar Topics:

- Tkinter widgets in python
- Python date/time module and its applications
- wxPython and its applications

M) Suggested Course Evaluation Matrix: The course teacher has to decide and use appropriate assessment strategy and its weightage in theory, laboratory and Term Work for ensuring CO attainment. The response/performance of each student in each of these designed activities is to be used to calculate **CO attainment**.

COs	Course Evaluation Matrix						
	Theory Assessment (TA)**		Term Work Assessment (TWA)			Lab Assessment (LA)#	
	Progressive Theory Assessment (PTA) Class/Mid Sem Test	End Theory Assessment (ETA)	Term Work & Self Learning Assessment			Progressive Lab Assessment (PLA)	End Laboratory Assessment (ELA)
			Assignments	Micro Projects	Other Activities*		
CO-1	10%	10%	15%	16%	16%	10%	16%

CO-2	15%	15%	15%	16%	16%	15%	16%
CO-3	25%	25%	20%	18%	18%	25%	18%
CO-4	15%	15%	15%	16%	16%	15%	16%
CO-5	25%	25%	25%	18%	18%	25%	18%
CO-6	10%	10%	10%	16%	16%	10%	16%
Total Marks	**30**	**70**	**20**	**20**	**10**	**20**	**30**
			50				

Legend:

*: Other Activities include self- learning, seminar, visits, surveys, product development, software development etc.

**: Mentioned under point- (N)

#: Mentioned under point-(O)

Note:

- The percentage given are approximate
- In case of Micro Projects and End Laboratory Assessment (ELA), the achieved marks will be equally divided in all those COs mapped with total experiments.
- For CO attainment calculation indirect assessment tools like course exit survey need to be used which comprises of questions related to achievement of each COs.

N) Suggested Specification Table for End Semester Theory Assessment: Specification table represents the reflection of sample representation of assessment of cognitive domain of full course.

Unit Title and Number	Total Classroom Instruction (CI) Hours	Relevant COs Number(s)	Total Marks	ETA (Marks)		
				Remember (R)	Understanding (U)	Application & above (A)
Unit-1.0 Basics of Python Programming syntax	4	CO-1	7	3	2	2
Unit-2.0 Conditional and Iterative statements	6	CO-2	10	3	3	4
Unit-3.0 3.0 String, List, Tuples, set and Dictionary	12	CO-3	18	5	3	10
Unit-4.0 Python Functions, Modules and packages	7	CO-4	10	3	3	4
Unit-5.0 Numpy	12	CO-5	18	4	5	9
Unit-6.0 Exception and File Handling in Python	7	CO-6	7	2	2	3
Total	**48**	**-**	**70**	**20**	**18**	**32**

Note: Similar table can also be used to design class/mid-term/ internal question paper for progressive assessment.

O) Suggested Assessment Table for Laboratory (Practical):

S. No.	Laboratory Practical Titles	Relevant COs Number(s)	PLA/ELA		
			Performance		Viva-Voce (%)
			PRA* (%)	PDA** (%)	
1.	Write and execute Python program to- a) Calculate the Area of a Triangle where its three sides a,b,c are given. s=(a+b+c)/2, Area=square root of s(s-a)(s-b)(s-c) (write program without using function) b) Swap Two Variables c) Solve quadratic equation for real numbers.	**CO-1**	40	50	10
2.	Write and execute Python program to- a) Check if a Number is Positive, Negative or zero. b) Check whether the given year is a Leap Year.	**CO-2**	40	50	10

S. No.	Laboratory Practical Titles	Relevant COs Number(s)	PLA/ELA		
			Performance		Viva-Voce (%)
			PRA* (%)	PDA** (%)	
	c) Print all Prime Numbers in an Interval. d) Display the multiplication Table based on the given input. e) Print the Fibonacci sequence. f) Find the Factorial of a Number.				
3.	Write and execute Python program to- a) Check whether the string is Palindrome b) Reverse words in a given String in Python c) identify in a strings the name, position and counting of vowels. d) Count the Number of matching characters in a pair of string (set) e) Python program for removing i-th character from a string	**CO-2, CO3**	40	50	10
4.	Write and execute Python program to- a) find largest number in a given list without using max(). b) find the common numbers from two lists. c) create a list of even numbers and another list of odd numbers from a given list. d) To find number of occurrences of given number without using built-in methods.	**CO-2, CO-3**	40	50	10
5.	Write and execute Python program to- a) find the index of an item of a tuple. b) find the length of a tuple. c) to reverse a tuple. d) Write a Python program to sort a list of tuple by its float element. Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')] Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]	**CO-2, CO-3**	40	50	10
6.	Write and execute Python program to- a) create an intersection of sets. b) create a union of sets. c) create set difference. d) check if two given sets have no elements in common.	**CO-2, CO-3**	40	50	10
7.	Write and execute Python program to- a) Write a Python script to concatenate two dictionaries to create a new one b) Write a Python script to merge two Python dictionaries. c) Write a Python program to combine two dictionary adding values for common keys. d1 = {'a': 100, 'b': 200, 'c':300} d2 = {'a': 300, 'b': 200, 'd':400} Sample output: d({'a': 400, 'b': 400, 'd': 400, 'c': 300})	**CO-2, CO-3**	40	50	10
8.	Write and execute Python program to- a) Write a Python function for reversing a string and call it. b) Write a Python function for calculating compound interest and call it.	**CO-2, CO-4**	40	50	10

S. No.	Laboratory Practical Titles	Relevant COs Number(s)	PLA/ELA		
			Performance		Viva-Voce (%)
			PRA* (%)	PDA** (%)	
	c) Write a Python function for calculating the factorial of a number and call it to calculate !n/(!r)*!(n-r)) where symbol "**!** " stands for factorial.				
9.	a) Write a python program to create a Numpy array filled with all zeros b) Write a python program to check whether a Numpy array contains a specified row c) Write a python program to Remove rows in Numpy array that contains non-numeric values d) Write a python program to Find the number of occurrences of a sequence in a NumPy array e) Write a python program to Find the most frequent value in a NumPy array f) Write a python program to Combine a one and a two-dimensional NumPy Array g) Write a python program to Flatten a Matrix in Python using NumPy Write a python program to Interchange two axes of an array	**CO-2, CO-5**	40	50	10
h)	Using exception handling feature such as try…except, try finally-write minimum three programs to handle following types of exceptions. viii. TypeError ix. NameError x. IndexError xi. KeyError xii. ValueError xiii. IOError xiv. ZeroDivisionError	**CO-2, CO-6**	40	50	10
i)	Write and execute Python program to- a) Calculate the Area of a Triangle where its three sides a,b,c are given. s=(a+b+c)/2, Area=square root of s(s-a)(s-b)(s-c) (write program without using function) b) Swap Two Variables c) Solve quadratic equation for real numbers.	**CO-1**	40	50	10

A) Course Code : 2418306(P2418306/S2418306)

B) Course Title : Summer Internship -I (Common For all Programmes)

C) Pre- requisite Course(s) :

D) Rationale :

Diploma students are required to give exposure of their own diploma programme related industrial hardware, software and practices, just after completing one semester, so that they can correlate this industrial exposure with the concept being taught in the branch specific specialized engineering courses in forthcoming semesters. Mentors/Coordinators/ Teachers need to map the academic contents of the programme of study with the activities of this industrial exposure and are advised to follow the 'Whole to Part' approach to make the students aware about the potential industry's expected outcomes & setup ('Whole') from the diploma programme – and then teaching the related concepts ('Part') of the same in subsequent semesters. In this way before actually being exposed to academic input specific to diploma programmes, the students need to be sent to the nearby/local industries and also may be advised to explore information related to their programme of study using different sources related to potential employment opportunities of both wage and self-employment, job function, job position, nearby relevant industries and so on.

The summer internship will provide the direction to the students and also help in mind mapping to plan their futuristic course of action, after passing the diploma. This would also bridge the gap between their virtual imagination about the outcome of the programme and real happenings related to the diploma programme.

E) Course Outcomes (COs): After the completion of the course, teachers are expected to ensure the accomplishment of following course outcomes by the learners. For this, the learners are expected to perform various activities related to three learning domains (Cognitive, Psychomotor and Affective) in classroom/laboratory/workshop/field/ industry.

After completion of the course, the students will be able to-

- **CO-1** Comprehend the practices of identified industry or world of work related to diploma engineering programme of study.
- **CO-2** Map real equipment, processes, product, management, operations etc. to the course of study through various glimpses of input, process and output in different type of industries.
- **CO-3** Identify the probable enterprises /startups for futuristic planning and self-growth.
- **CO-4** Identify the probable job function and job position in their relevant programme of study.

F) Suggested Course Articulation Matrix (CAM):

Course Outcomes (COs)	Programme Outcomes (POs)							Programme Specific Outcomes* (PSOs)	
	PO-1 Basic and Discipline Specific Knowledge	**PO-2** Problem Analysis	**PO-3** Design/ Development of Solutions	**PO-4** Engineering Tools	**PO-5** Engineering Practices for Society, Sustainability and Environment	**PO-6** Project Management	**PO-7** Life Long Learning	**PSO-1**	**PSO-2**
CO-1	3	-	-	1	-	-	1		
CO-2	3	-	-	1	-	-	1		
CO-3	3	-	-	-	1	-	2		
CO-4	3	-	-	-	1	-	2		

Legend: High (3), Medium (2), Low (1) and No mapping (-)

* PSOs will be developed by respective programme coordinator at institute level. As per latest NBA guidelines, formulating PSOs is optional

G) Teaching & Learning Scheme:

Board of Study	Course Code	Course Title	Scheme of Study (Hours/Week)					
			Classroom Instruction (CI)		Lab Instruction (LI)	Notional Hours (TW+ SL)	Total Hours (CI+LI+TW+SL)	Total Credits (C)
			L	T				
	2418306	Summer Internship -I	-	-	02	02	04	02

Legend:

CI: Classroom Instruction (Includes different instructional/implementation strategies i.e. Lecture (L), Tutorial (T), Case method, Demonstrations, Video demonstration, Problem based learning etc. to deliver theoretical concepts)

LI: Laboratory Instruction (Includes experiments/practical performances /problem-based experiences in laboratory, workshop, field or other locations using different instructional/Implementation strategies)

Notional Hours: Hours of engagement by learners, other than the contact hours for ensuring learning.

TW: Term Work (includes assignments, seminars, micro projects, industrial visits, any other student activities etc.)

SL: Self Learning, MOOCs, spoken tutorials, online educational resources etc.

C: Credits = (1 x CI hours) + (0.5 x LI hours) + (0.5 x Notional hours)

Note: TW and SL have to be planned by the teacher and performed by the learner under the continuous guidance and feedback of teacher to ensure outcome of learning.

H) Assessment Scheme:

Board of Study	Course Code	Course Title	Assessment Scheme (Marks)						Total Marks (TA+TWA+LA)
			Theory Assessment (TA)		Term Work & Self-Learning Assessment (TWA)		Lab Assessment (LA)		
			Progressive Theory Assessment (PTA)	End Theory Assessment (ETA)	Internal	External	Progressive Lab Assessment (PLA)	End Laboratory Assessment (ELA)	
	2418306	Summer Internship -I	-	-	10	15	10	15	50