Curriculum

of

Diploma Programme

in

Computer Science & Engineering

State Board of Technical Education (SBTE)
Bihar

Semester – IV
Teaching & Learning Scheme

Board of Study	Course Codes	CourseTitles	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				
	2418401	**Java Programming**	3	-	4	2	9	6
	2418402	**Theory of Computation**	2	1	-	2	5	4
	2418403	**Database Management System**	3	-	4	2	9	6
	2418404	**Computer Organization & Architecture**	2	1	-	2	5	4
	2418405	**Computer Troubleshooting and Maintenance**	3	-	4	2	9	6
	2400006	**Environmental Education and Sustainable Development** (Common for All Programmes)	1	-	1	1	3	2
	2400408	**Employability Skills Development** (Common for All Programmes)	1	-	-	-	1	1
	2400110	**Community/ Society Development** (Non-exam course) (AIML, AE, CSE, ELX (R), CHE, EE, ME, ME (Auto), MIE, FTS, CACDDM,)	1	-	-	-	1	1
Total			**16**	**2**	**13**	**11**	**42**	**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 - IV

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)	
	2418401	**Java Programming**	30	70	20	30	20	30	200
	2418402	**Theory of Computation**	30	70	20	30	-	-	150
	2418403	**Database Management System**	30	70	20	30	20	30	200
	2418404	**Computer Organization & Architecture**	30	70	20	30	-	-	150
	2418405	**Computer Troubleshooting and Maintenance**	30	70	20	30	20	30	200
	2400006	**Environmental Education and Sustainable Development** (Common for All Programmes)	15	-	10	-	10	15	50
	2400408	**Employability Skills Development** (Common for All Programmes)	25	-	-	-	-	-	25
	2400110	**Community/ Society Development** (Non-exam course) (AIML, AE, CSE, ELX (R), CHE, EE, ME, ME (Auto), MIE, FTS, CACDDM,)	25	-	-	-	-	-	25
Total			**215**	**350**	**110**	**150**	**70**	**105**	**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 : **2418401 (T2418401/P2418401/S2418401)**

B) Course Title : Java Programming

C) Pre- requisite Course(s) : ICT Tools

D) Rationale

Java is platform independent, open-source object-oriented programming language enriched with free and open source libraries. In current industrial scenario Java has the broad industry support and is prerequisite with many allied technologies like Advanced Java, Java Server Pages, and Android Application Development. Thus. Current industrial trends necessitate acquiring Java knowledge for Computer Engineering and Information Technology graduates. This course develops necessary skills in students to apply object-oriented programming techniques in Java so that students will be able to develop complete applications using core Java.

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 Implement Control structure in Java using concept of Class and Object

CO-2 Develop programs using Constructor and String Class

CO-3 Apply concept of inheritance for code reusability.

CO-4 Implement Exception Handling and multithreading

CO-5 Develop programs for handling I/O and file streams.

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	1	-	-		-	-	1		
CO-2	-	1	-	1	-	-	-		
CO-3	-	2	1	1	-	-	1		
CO-4	2	1	1	1	-	-	-		
CO-5	1	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				
Computer Science and Engineering	**2418401**	Java Programming	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)	
Computer Science and Engineering	**2418401**	Java Programming	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, Society connect, Indian Knowledge System (IKS) and others must be integrated appropriately.

J) Theory Session Outcomes (TSOs) and Units: T2418401

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1a. Write programs to create classes and objects for the given problem. *TSO 1b.* Explain the characteristics of the specified Java token. *TSO 1c.* Explain the function of the given operator with example. *TSO 1d.* Develop the programs using relevant control structure to solve the specified problem. *TSO 1e.* Explain the functioning of Java Virtual Machine *TSO 1f.* Explain the function of specified built-in Java Package	**Unit-1.0 Basic Syntactical Constructs in Java** 1.1 Java Features and the Java Programming Environment 1.2 Object-Oriented Paradigm:- - Objects & Classes - Data Abstraction - Data Encapsulation - Inheritance - Polymorphism 1.3 Benefits of OOP 1.4 Applications of OOP 1.5 Java History 1.6 Java Features - Simple small & Familiar - Compiled or Interpreted - Platform independent - Portable - Object Oriented - Robust & Secure - Distributed - Multithreaded & Interactive - Ease of development 1.7 Java vs C 1.8 Java vs C++ 1.9 Java Environment - Java Development Kit (JDK) - Java Development Tool 1.10 Application Programming Interface - Language Support Package - Utilities Package - Input/output Package - AWT Package - Applet Package - Networking Package 1.11 Simple Java Program - Class Declaration - Opening & Closing Braces - Main line - Output line - Creating object 1.12 Java Programming with multiple Statement - Application with two classes - Accessing class members	**CO-1**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
	- Use of Math Function - Comments 1.13 Java Program Structure - Document Section - Package statement - Import Statement - Interface Statements - Class Definitions - Main method Class 1.14 Java Tokens - Reserved Keywords - Identifiers - Literals - Operators - Separators 1.15 Compiling the Program - Java Compiler javac 1.16 Running the Program - Java Interpreter 1.17 Java Virtual Machine - Byte Code 1.18 Command line Arguments 1.19 Constant & Variables 1.20 Data Types - Primitive Data Types - Non Primitive Data Types 1.21 Operators:- - Arithmetic Operators - Relational Operators - Logical Operators - Increment & Decrement - Conditional Operators - Bit wise Operator - Dot Operators - Ternary Operator 1.22 Expressions:- - Operator precedence - Associativity 1.23 Decision Statement:- - if statement - if else statement - nested if else statement - if else if ladder - switch statement 1.24 Loop Statement:- - While statement - Do while statement - for statement - for-each statement. 1.25 Control Statement:- - Break	

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
	- Continue - Return statement	
TSO 2a. Use Constructors for the given programming problem. TSO 2b. Identify scope and lifetime of a variable in the given program code. TSO 2c. Describe the given visibility control with example. TSO 2d. Write the programs by implementing arrays to solve the given problem.	**Unit 2. Derived Syntactical Constructs in Java:** 2.1 Constructors - Default - Parameterized - Non parameterized 2.2 'this' keyword 2.3 Command line arguments - Varargs: variable-length arguments 2.4 Visibility Control: - Public - Private - Protected - Default 2.5 Arrays - Types of arrays - Declaration of Arrays - Creating an arrays - Initialization of Arrays 2.6 Strings - String classes - String Buffer	**CO-2**
TSO 3a. Apply the identified type of inheritance for the given programming problem. TSO 3b. Differentiate between overloading and overriding for the given example. TSO 3c. Develop program using the specified interface. TSO 3d. Create user defined package for the given problem.	**Unit 3. Inheritance, Interface and Package:** 3.1 Inheritance : Concept of Inheritance 3.2 Types of Inheritance - Single Inheritance - Multiple - Multilevel Inheritance - Hierarchical Inheritance 3.3 Interface - Defining interface - Extending interface - Implementing interface 3.4 Method overloading and overriding 3.5 Package - Define package - Types of package - Naming and creating packages - Accessing package - Using a Package - Import statement	**CO-3**
TSO 4a. Distinguish the errors and exceptions (if any) in the given example.	**Unit 4. Errors & Exception Handling, Multithreading:** 4.1 Errors - Introduction 4.2 Types of Errors	**CO-4**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 4b. Develop program for handling the given exception. TSO 4c. Crate threads to run the given multiple processes in the given program. TSO 4d. Explain the function of the specified phase in thread life cycle using the given example.	- Compile-Time Errors - Run-Time Errors 4.3 Exceptions - try and catch statement - nested try statement, - throws and Finally statement 4.4 Built-in exceptions 4.5 Multithreaded Programming - Creating a Thread: By extending to thread class - Creating a thread by implementing runnable Interface 4.6 Life cycle of thread: - New State - Runnable State - Running State - Blocked State - Dead State 4.7 Thread Methods: - wait () - sleep() - notify() - resume() - suspend() - stop()	
TSO 5a. Use I/O stream classes in a program to solve the given problem. TSO 5b. Write programs for reading and writing character streams to and from the given file. TSO 5c. Write programs for reading and writing bytes to and from the given file. TSO 5d. Write program to demonstrate use of primitive Data types with the specified stream.	**Unit 5. Managing Input/Output/Files in Java:** 5.1 Introduction and Concept of Streams. 5.2 Stream Classes. 5.3 Byte Stream Classes: - Input Stream Classes - Output Stream Classes. 5.4 Character Stream Classes: - Using Reader Stream classes - Using Writer Stream classes 5.5 Using File Class: - I/O Exceptions - Creation of Files - Reading/Writing Bytes - Handling Primitive Data types.	**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: P2418401

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 1.1. Install Java IDE.	1.	a) Setup a Java Programming development environment by using: • Command prompt(Classpath and path setup) • Any IDE (Eclipse, Jcreator etc.) b) Test the JDE Setup by implementing a small Program	CO-1

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 2.1. Develop program using decision-making Control Statements in Java.	2.	Write programs to demonstrate use of if statement, if else, nested if statement for decision making.	CO-1
LSO 4.1. Develop program in Java using loop statements to solve iterative problems.	3.	Write programs to demonstrate use of for loops and do-while loop.	CO-1
LSO 5.1. Implement constructor in Java	4.	Develop a program for implementation of- • Constructors • Parameterized constructor • Non parameterized constructor	CO-2
LSO 6.1. Use different functions available in java String Class	5.	Develop program demonstrating implementation of different String Class functions.	CO-2
LSO 7.1. Implement single and multidimensional array in Java	6.	Develop program for implementation of Arrays in Java	CO-2
LSO 8.1. Implement Overloading, Overriding concepts in Java	7.	Develop programs to demonstrate implementation of overloading and overriding concepts	CO-3
LSO 9.1. Implement concept of code reusability using inheritance in Java	8.	Develop programs for implementation of: • single inheritance • multiple Inheritance • multilevel Inheritance • hierarchical Inheritance	CO-3
LSO 10.1. Apply the concepts of Errors & Exceptional Handling	9.	Develop program to implement try-catch block	CO-4
LSO 11.1. implement concept of File handling using I/O stream	10.	Develop program to demonstration implementation of I/O stream classes	CO-5
LSO 12.1 implement concept of File handling using file stream	11.	Develop program to demonstrate implementation of file stream classes	CO-5

2. Self-learning topics:

- Various Java API
- Various Java Packages

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**.

	Course Evaluation Matrix						
	Theory Assessment (TA)**		**Term Work Assessment (TWA)**			**Lab Assessment (LA)#**	
COs	**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	25%	25%	25%	20%	20%	25%	25%
CO-2	20%	20%	20%	20%	20%	20%	20%
CO-3	20%	20%	20%	20%	20%	20%	20%
CO-4	20%	20%	20%	20%	20%	20%	20%
CO-5	15%	15%	15%	20%	20%	15%	15%
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 Basic Syntactical Constructs in Java	12	CO-1	18	6	4	8
Unit 2.0 Derived Syntactical Constructs in Java	8	CO-2	14	4	6	4
Unit 3.0 Inheritance, Interface and Package	10	CO-3	14	4	4	6
Unit 4.0 Errors & Exception Handling, Multithreading	10	CO-4	14	4	4	6
Unit 5.0 Managing Input/ Output/ Files in Java	8	CO-5	10	2	2	6
Total	**48**	**-**	**70**	**20**	**20**	**30**

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

O) Suggested Specification Table for Laboratory (Practical) Assessment:

S. No.	Laboratory Practical Titles	Relevant COs Number(s)	PLA /ELA		
			Performance		Viva-Voce (%)
			PRA* (%)	PDA** (%)	
1.	c) Setup a Java Programming development environment by using: • Command prompt(Classpath and path setup) • Any IDE (Eclipse, Jcreator etc.) d) Test the JDE Setup by implementing a small Program	CO-1	80	10	10
2.	Write programs to demonstrate use of if statement, if else, nested if statement for decision making.	CO-1	40	50	10
3.	Write programs to demonstrate use of for loops and do-while loop.	CO-1	40	50	10
4.	Develop a program for implementation of- • Constructors • Parameterized constructor • Non-parameterized constructor	CO-2	40	50	10
5.	Develop program demonstrating implementation of different String Class functions.	CO-2	40	50	10
6.	Develop program for implementation of Arrays in Java	CO-2	40	50	10
7.	Develop programs to demonstrate implementation of overloading and overriding concepts	CO-3	40	50	10
8.	Develop programs for implementation of: • single inheritance • multiple Inheritance • multilevel Inheritance • hierarchical Inheritance	CO-3	40	50	10
9.	Develop program to implement try-catch block	CO-4	50	40	10
10.	Develop program to demonstration implementation of I/O stream classes	CO-5	40	50	10
11.	Develop program to demonstrate implementation of file stream classes	CO-5	40	50	10

Legend:

PRA[*]: Process Assessment

PDA[**]: Product Assessment

A) Course Code : **2418402(T2418402/S2418402)**

B) Course Title : Theory of computation (AIML, CSE)

C) Pre- requisite Course(s) :

D) Rationale :

Theory of Computation is a broad field of study focused on creating more efficient algorithms and other computational processes. This course equips students with essential theoretical knowledge in areas such as automata theory, formal languages, computability, and complexity theory.

This course prepares students to comprehend the capabilities and limitations of computer systems, enabling them to develop critical thinking and problem-solving skills necessary in algorithm design and computational problem-solving within various engineering domains including artificial intelligence.

E) Course Outcomes (COs): After the completion of the course, teachers are expected to ensure the accomplishment of the 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 the classroom/ laboratory/ workshop/ field/ industry.

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

CO-1 Establish Relationship between grammar and language in the context of Theory of Computation.

CO-2 Apply finite automata, regular expressions, and regular grammar to recognize and generate regular languages.

CO-3 Analyze context-free languages and pushdown automata to determine their equivalence with context-free grammars.

CO-4 Evaluate Turing machines to solve problems considering their capabilities and limitations.

CO-5 Apply Rice's theorem to demonstrate undecidable problems about languages.

CO-6 Transforming one problem into another problem using reduction algorithm.

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	1	-	-	-	-	-	1		
CO-2	2	2	2	1	-	-	1		
CO-3	2	2	2	1	-	-	-		
CO-4	2	2	2	1	-	-	-		
CO-5	2	2	2	-	-	-	-		
CO-6	1	2	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	**2418402**	Theory of Computation	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)	
AIML	**2418402**	Theory of computation	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), 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, Society connect Indian Knowledge System (IKS) and others must be integrated appropriately.

J) Theory Session Outcomes (TSOs) and Units: T2418402

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1a. Explain the fundamental concepts of alphabets, languages, and grammar, and their role in the theory of computation. TSO 1b. Apply productions and derivations to generate and manipulate strings within a given language. TSO 1c. Compare different types of languages based on their grammatical structures and complexities. TSO 1d. Explain the significance and implications of the Chomsky Hierarchy in classifying languages and understanding their computational power.	**Unit-1.0 Introduction to Theory of Computation** 1.1 Alphabet, Languages, and Grammar: - Introduction to Alphabet: Set of Symbols or Characters -Exploration of the languages: Sets of strings -Introduction to grammar: Set of rules, - Structure, and formation of valid strings. Productions and derivations 1.2 Productions and Derivations: - Concept of production in the grammar. - Explanation of the derivation to generate a string. 1.3 Chomsky Hierarchy of Languages: - Introduction to Chomsky Hierarchy. - Levels of formal language category: Type-3 (Regular), Type-2 (Context-Free), Type-1 (Context-Sensitive), and Type-0 (Unrestricted).	CO-1
TSO 2a. Explain the concept of regular expressions and their role in defining regular languages. TSO 2b. Compare nondeterministic finite automata (NFA) with DFA and establish their equivalence. TSO 2c. Explain the relationship between regular grammars and finite automata and demonstrate their equivalence. TSO 2d. Prove laws related to properties of regular languages, such as closure under union, intersection, and complementation. TSO 2e. Apply the pumping lemma for regular languages to prove the non-regularity of specific languages. TSO 2f. Compare the characteristics of Mealy and Moore Machine.	**Unit-2.0 Regular languages and** Finite **Automata** 2.1 Regular expressions and languages., Deterministic finite automata (DFA) and equivalence with regular expressions. 2.2 Nondeterministic finite automata (NFA) and equivalence with DFA, 2.3 Regular grammars and equivalence with finite automata, Properties of regular languages, Pumping lemma for regular languages, 2.4 Minimization of finite automata. 2.5 Mealy and Moore machine	CO-2
TSO 3a. Explain the concept of context-free grammars (CFG) and their role in defining context-free languages (CFL). TSO 3b. Converts a Chomsky normal form into Greibach Normal Form TSO 3c. Establish the equivalence between nondeterministic pushdown automata (NPDA) and context-free grammars.	**Unit 3.** Context-**Free Languages and Pushdown Automata** 3.1 Context-free grammars (CFG) and languages (CFL) 3.2 Chomsky and Greibach normal forms 3.3 Nondeterministic pushdown automata (NPDA)	CO-3

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 3d. Apply the pumping lemma for context-free languages to prove the non-context-freeness of specific languages.	and equivalence with CFG 3.4 Parse trees and ambiguity in CFG 3.5 Pumping lemma for context-free languages 3.6 Deterministic pushdown automata and closure properties of CFLs.	
TSO 4a. Explain the concept of context-sensitive grammars (CSG) and their role in defining context-sensitive languages (CSL). TSO 4b. Analyze equivalence between linear bounded automata (LBA) and context-sensitive grammars. TSO 4c. Explain the basic model of Turing machines (TM) and their computational capabilities with example. TSO 4d. Differentiate between Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages. TSO 4e. Establish equivalence of different turning machine variants. TSO 4f. Explain the concept of unrestricted grammars and establish their equivalence with Turing machines. TSO 4g. Explain the role of Turing machines as enumerators in the computation of languages.	**Unit 4. Context-Sensitive Languages and Turing Machines** 4.1 Context-sensitive grammars (CSG) and languages 4.2 Linear bounded automata (LBA) and equivalence with context-sensitive grammar 4.3 The basic model for Turing machines (TMs) 4.4 Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties 4.5 Variants of Turing machines, including nondeterministic TMs and equivalence with deterministic TMs 4.6 Unrestricted grammars and equivalence with Turing machines 4.7 TMs as enumerators.	CO-4
TSO 5a. Explain the Church-Turing thesis and its significance in defining the concept of computability. TSO 5b. Establish reduction techniques between different languages. TSO 5c. Prove the Rice's theorem of undecidable problems about languages. TSO 5d. Differentiate between the complexity classes P, NP, NP-complete, and NP Hard. TSO 5e. Apply reduction techniques and complexity analysis to classify problems within the P, NP, NP-complete, and NP Hard classes.	**Unit 5. Undecidability** 5.1 Church-Turing thesis and universal Turing machine 5.2 Diagonalization languages and reduction between languages, Reduction Theorem, Properties of Reduction 5.3 Rice's theorem and undecidable problems about languages 5.4 P, NP, NP-complete, and NP Hard class of problems	CO-5 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: (Not Applicable)

L) Suggested Term Work and Self Learning: S2418402 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. Regular Expression Matcher: Create a program that can match regular expressions against input strings. You can implement this using the Thompson's construction algorithm for NFA from regex.

2. Pushdown Automaton Visualizer: Develop a graphical tool that visualizes the operation of a pushdown automaton (PDA) on input strings. This can help users understand how PDAs process input.
3. Context-Free Grammar Parser: Create a program that parses and validates context-free grammars. Users can input a grammar, and the program should check if it's in the correct format.
4. Comparative Analysis of Complexity Classes: Create an informative chart comparing the complexity classes P, NP, NP-complete, and NP Hard. Include a detailed explanation of each class, examples of problems belonging to each class, and their relationship with each other. Highlight the significance of these complexity classes in understanding the computational difficulty of problems.

b. Other Activities:

1. Seminar Topics:

- Automata Theory and Applications: Explore the basics of automata theory, its types (finite automata, pushdown automata, Turing machines), and practical applications in computer science.
- Complexity Theory: Investigate the different complexity classes (P, NP, NP-complete, PSPACE, etc.), their relationships, and their implications in solving computational problems efficiently.
- Formal Languages and Grammars: Discuss formal language theory, context-free grammars, regular expressions, and their relevance in programming languages and compiler design.
- Decidability and Undecidability: Explore the concept of decidability, undecidability, and examples of undecidable problems in computer science.
- Reduction Techniques in Computational Complexity: Investigate various reduction techniques, such as polynomial-time reductions and Cook-Levin reductions, and their applications in proving NP-completeness.

2. Self-learning topics:

- Non-standard computational models like quantum Turing machines, cellular automata.
- Relationship between computational complexity theory and cryptography.
- Theory of Computation in Machine Learning.

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**.

	Course Evaluation Matrix						
	Theory Assessment (TA)**		**Term Work Assessment (TWA)**			**Lab Assessment (LA)$^{\#}$**	
COs	**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	20%	20%	20%	-	15%	-	-
CO-2	20%	20%	20%	20%	20%	-	-
CO-3	20%	20%	20%	20%	20%	-	-
CO-4	20%	20%	20%	20%	20%	-	-
CO-5	15%	15%	15%	20%	20%	-	-
CO-6	5%	5%	5%	20%	05%	-	-
Total Marks	**30**	**70**	**20**	**20**	**10**	**-**	**-**
			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 Introduction	9	CO-1	14	4	6	4
Unit-2.0 Regular languages and Finite Automata	10	CO-2	14	4	4	6
Unit 3.0 Context-free languages and pushdown automata	11	CO-3	14	4	4	6
Unit 4.0 Context-sensitive languages and Turing machines	10	CO-4	14	4	6	4
Unit 5.0 Undecidability	8	CO-5 CO-6	14	4	4	6
Total	**48**	**-**	**70**	**20**	**24**	**26**

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): (Not Applicable)

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, Field Trips, 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 Session, Video Clippings, Use of Open Educational Resources (OER), MOOCs etc.

Q) List of Major Laboratory Equipment, Tools and Software: (Not Applicable)

R) Suggested Learning Resources:

(a) Books :

A) Course Code : 2418403(T2418403/P2418403/S2418403)

B) Course Title : Database Management System (AIML, CSE)

C) Pre- requisite Course(s) : ICT Tools

D) Rationale :

Database Management System (DBMS) is a vital components of information systems for development of any software application. Any software application deals with large data set which has to be properly organized to provide necessary input to the application. The DBMS focuses on structures and principles necessary to design and implement a database management system.

In this course Students will learn the approach and process of good database designs. Student will also learn to use Structured Query Language to create and manipulate database appropriately to serve the requirement of given software application.

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 Illustrate the fundamental concepts of Database, Database System and Database Management System.

CO-2 Use the concepts of E-R Modeling, Keys and constraints to design a database

CO-3 Normalize/De-normalize the database to optimize its performance

CO-4 Use Structured Query Language (SQL) for database manipulation

CO-5 Create and use schema objects such as View, Index, Synonyms and Sequence to optimize database performance.

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	1	-	-		-	-	1		
CO-2	-	1	-	1	-	-	-		
CO-3	-	2	2	1	1	-	1		
CO-4	2	1	2	1	1	-	-		
CO-5	1	1	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				
AIML	**2418403**	Database Management System	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)	
	2418403	Database Management System	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, Society connect, Indian Knowledge System (IKS) and others must be integrated appropriately.

J) Theory Session Outcomes (TSOs) and Units: T2418403

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1a. Illustrate concept of Database Management System. *TSO 1b.* State the importance of DBMS over file processing. *TSO 1c.* Describe the overall structure of the given DBMS *TSO 1d.* Explain the characteristics of Relational database model *TSO 1e.* Explain characteristics of given Database systems.	**Unit-1.0 Overview of the Database Management System** 1.1 Database- Concept of database, Need of Database, Advantage of database, Application of Database, Traditional Database 1.2 Database Management System, File Processing System, Advantages of DBMS over file processing system, Characteristic of Database 1.3 Relational Data Model- Domain, Attributes, Tuples and Relations 1.4 Types of Database System-Centralized Database System, Parallel Database System, Client / Server Database System, Distributed Database System	**CO-1**
TSO 2a. Describe the given term related to RDBMS *TSO 2b.* Describe the given components of E-R diagram. *TSO 2c.* Explain the purpose of given type of Key in DBMS *TSO 2d.* Apply given Integrity Constraint on database. *TSO 2e.* Convert given E-R diagram into Table	**Unit 2. Relational Database Management System (RDBMS):** 2.1 Introduction to RDBMS, RDBMS terminology. Relational Model (Instances, Schema). 2.2 E-R model concept- Notation for E-R diagram, Component of E-R diagram, Strong Entity set, Weak Entity set, Types of Attributes, E-R design Issues 2.3 Keys in DBMS- Primary key, Candidate key, Foreign key, Super Keys, Alternate Keys 2.4 Integrity Constraints- Domain Constraint, Entity Integrity Constraint, Referential Integrity Constraint, Key Constraint 2.5 Conversion of E-R diagram into Table	**CO-2**
TSO 3a. Find Functional Dependencies in a relation for good database design. *TSO 3b.* Describe closure properties in database. *TSO 3c.* Normalization the given database from one normal form to other normal form. *TSO 3d.* De-normalize database for optimizing its performance	**Unit 3 Relational Database Design:** 3.1 Functional Dependency, Closures of a Set of Functional Dependencies. 3.2 Normalization & Normal forms- 1NF, 2NF, 3NF, BCNF 3.3 Denormalization: Process, benefits and draw back.	**CO-3**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 4a. Describe the use of given relational algebra operator with example. *TSO 4b.* Explain the given join operation on tables with example. *TSO 4c.* Write SQL queries for adding, deleting and updating table data. *TSO 4d.* Write queries by implementing given aggregate functions on data. *TSO 4e.* Write SQL queries to display the data in sorted order. *TSO 4f.* Combine the data as per given criteria *TSO 4g.* Write SQL Statement to join two relations *TSO 4h.* Write Correlated and Nested Query *TSO 4i.* Write SQL to control the database transaction	**Unit 4. Relational Algebra & SQL:** 4.1 Concept of relational algebra 4.2 Types of relational operations- Select operation, Project operations, Union operations, Set Intersection operations, Set Difference operations, Cartesian operations, Rename operations 4.3 Join operations- Natural Join, Outer Join, Equi Join 4.4 Overview of SQL- SQL Commands - Data Definition language(DDL), Data Manipulation Language(DML), Data Control Language(DCL) - Select, Insert, Update, Delete SQL Statements 4.5 SQL Clauses- Group by, Having, Order by clause 4.6 Aggregate functions- Max, Min, Sum, Count, Avg 4.7 SQL Join Operations, Inner join, Left join, Right join, Full join 4.8 SQL Queries - Correlated and Nested Query 4.9 TCL: Commit, save point, rollback, set transaction	**CO-4**
TSO 5a. Write SQL statement to create and manipulate view *TSO 5b.* Write SQL statement to create and manipulate sequence. *TSO 5c.* Write SQL to create and drop Index in table. *TSO 5d.* Write SQL to create and drop synonyms in database	**Unit 5. Other Schema Objects** 5.1 Views: Concept of View, The Create and update Views, Views and Joins, Views and Sub queries, Dropping Views. 5.2 Sequences: Concept and need of Sequence Creating Sequences, Altering Sequences, Dropping Sequences. 5.3 Indexes: need of index, index Types, creating of an Index: Simple Unique, and Composite Index, Dropping Indexes 5.4 Synonyms: Creating Synonyms, Dropping Synonyms.	**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: P2418403

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 1.1. Install and configure Database product.	1.	Database installation (such as MySQL, MariaDB)	CO-4
LSO 2.1. Write and execute DDL command to create a database for the identified problem	2.	Design table structure	CO-4
LSO 3.1. Apply given integrity constraint on table	3.	Apply integrity constraints	CO-4, CO-2
LSO 4.1. Write and execute DML commands to insert, delete and update data	4.	Use DML commands.	CO-4
LSO 5.1. Write and execute queries using relational algebraic operations.	5.	Apply relational algebraic operations	CO-4
LSO 6.1. Use given aggregate function in SQL Query	6.	Write statements to demonstrate the use of aggregate functions	CO-4
LSO 7.1. Implement different join operations using queries	7.	Perform join operations	CO-4

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 8.1. Write and execute Correlated and Nested Query for given problem	8.	Write Correlated and Nested Query	CO-4
LSO 9.1. Perform Transaction control operations on a database	9.	Write TCL Queries	CO-4
LSO 11.1. implement concept of view to optimize database handling	10.	Implement Views to perform following operations: a. Create views. b. Insert, modify and delete records through views. c. Delete the views.	CO-5
LSO 12.1. implement Synonyms for Database usage simplification and flexibility *LSO 12.2.* Create and use index for efficient access of ordered records. *LSO 12.3.* Create and use sequences to avoid possible concurrency	11.	Create Indexes, Sequences, and Synonyms in SQL.	CO-5

L) **Suggested Sessional Work and Self Learning: S2418403** 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.

- Teachers are suggested to identify various systems for computerization and students can be assigned to make E-R diagram (which can later be converted in to tables) to design suitable database.
- Normalize the given table to successive Normal form

b. **Micro Projects:**

1. Identify various Entity and attributes and relation between them for Library Management System.
2. Draw ER Diagram for Hospital Management System Database.
3. Identify a system, create database and Normalize the database consecutively up to 3NF
4. Prepare a presentation on by taking an example

c. **Other Activities:**

1. Seminar Topics: -

- Importance of Database Management System.
- various vulnerabilities in database management systems
- Database recovery techniques
- Concurrency control in database

2. **Self-learning topics:**

- Hierarchical and network Database System
- Transaction management: ACID properties

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**.

	Course Evaluation Matrix						
	Theory Assessment (TA)**		**Term Work Assessment (TWA)**			**Lab Assessment (LA)$^{\#}$**	
COs	**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%	10%	-	-	-	-
CO-2	25%	25%	30%	25%	-	-	-
CO-3	15%	15%	30%	25%	33%	-	-
CO-4	25%	25%	15%	25%	33%	60%	60%
CO-5	20%	20%	15%	25%	34%	40%	40%
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. An Overview of the Database Management System	9	CO-1	11	5	6	0
Unit 2. Relational Database Management System (RDBMS)	10	CO-2	17	3	6	8
Unit 3 Relational Database Design	9	CO-3	10	4	2	4
Unit 4. Relational Algebra & SQL	12	CO-4	18	4	6	8
Unit 5. Other Schema Objects	8	CO-5	14	4	4	6
Total	**48**	**-**	**70**	**20**	**24**	**26**

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.	Database installation (such as MySQL, MariaDB)	CO-4	80	20	-
2.	Design table structure	CO-4	40	60	10
3.	Apply integrity constraints	CO-4, CO-2	40	60	10
4.	Use DML commands.	CO-4	40	60	10
5.	Apply relational algebraic operations	CO-4	40	60	10
6.	Write statements to demonstrate the use of aggregate functions	CO-4	40	50	10
7.	Perform join operations	CO-4	40	50	10
8.	Write Correlated and Nested Query	CO-4	40	50	10
9.	Write TCL Queries	CO-4	80	10	10
10.	Implement Views to perform following operations: a. Create views. b. Insert, modify and delete records through views. c. Delete the views.	CO-5	40	50	10
11.	Create Indexes, Sequences, and Synonyms in SQL.	CO-5	30	60	10

Legend:

PRA*: Process Assessment
PDA**: Product Assessment

A) Course Code : **2418404(T2418404/S2418404)**

B) Course Title : Computer Organization and Architecture

C) Pre- requisite Course(s) : Fundamentals of IT System, Digital System

D) Rationale :

Computer architectures represent the means of interconnectivity for a computer's hardware components as well as the mode of data transfer and processing exhibited. Different computer architecture configurations have been developed to speed up the movement of data, allowing for increased data processing. The basic architecture has the CPU at the core with a main memory and input/output system on either side of the CPU. Computer Organization lets you know how exactly each instruction is executed at the micro level. For the study of embedded systems/ processor design, these concepts are very important, as they form the basis of design strategy.

This paper enables students to acquire basic knowledge internal architecture of computer and understand the functioning of arithmetic, logic and memory unit of computer. Students can utilize the concept in building compilers, developing efficient programs and optimize program behavior.

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 Elaborate the basic architecture of Central processing Unit (CPU)

CO-2 Analyze the operations of Arithmetical and Logical Unit (ALU) by applying various arithmetic operation principles.

CO-3 Analyze communication paradigm of I/O devices and standard I/O interfaces

CO-4 Analyze the performance of various classes of Memories

CO-5 Calculate the performance of CPU in Pipelined based architecture.

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** Proble m Analysis	**PO-3** Design/ Developmen tof 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	-	-	-	-	-	-		
CO-2	2	2	1	1	-	-	1		
CO-3	2	2	2	1	-	-	-		
CO-4	2	1	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 Organization and Architecture	**2418404**	Computer Organization and Architecture	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	**2418404**	Computer Organization and Architecture	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), 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, Society connect Indian Knowledge System (IKS) and others must be integrated appropriately.

J) Theory Session Outcomes (TSOs) and Units: T2418404

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1.a. Define the concept of "Stored Program Organization" in computer architecture. TSO 1.b. Explain the different types of buses, including Address, Data, and Memory buses. TSO 1.c. Explain the various addressing modes used in computer instruction sets. TSO 1.d. Explain the three stages of the "Instruction Cycle" (Fetch, Decode, Execute) and their significance. TSO 1.e. Differentiate between "Maskable" and "Non-Maskable" interrupts, and explain their roles in computer systems. TSO 1.f. Describe the process of "Register, Bus, and Memory transfer" within a CPU and its importance in data processing. TSO 1.g. Implement a "Subroutine call and return" in a simple program.	**Unit 1.0: Basic Computer Organization and Design:** 1.1 Instruction Codes 1.1.1 Stored Program Organization 1.1.2 Instruction Set 1.1.3 Instruction Cycle (Fetch, Decode, Execute) 1.1.4 Register Reference Instruction 1.2 Memory Reference Instructions 1.3 Interrupt 1.3.1 Hardware and Software 1.3.2 Maskable and Non-Maskable 1.3.3 Input and Output 1.4 Types of buses (Address, Data, Memory) and bus arbitration. 1.5 Register, Bus, and memory transfer. 1.6 Processor organization, general registers organization (Control Word), stack organization (Register Stack, Memory Stack) 1.7 Addressing Modes (Direct, Indirect, Register Direct, Immediate etc.) 1.8 Program Control 1.8.1 Conditional Branch Instruction 1.8.2 Subroutine call and return	**CO-1**
TSO 2.a. Define "Signed Magnitude" and explain how addition and subtraction are performed using this representation. TSO 2.b. List the steps involved in the hardware implementation of addition and subtraction operations. TSO 2.c. Illustrate the multiplication algorithm for signed operands, emphasizing the difference between signed and unsigned multiplication. TSO 2.d. Explain "Booth's Algorithm" and "Array Multiplier" as methods for efficient multiplication and explain when to use each. TSO 2.e. Describe the hardware implementation of division algorithms for signed magnitude	**Unit 2: Arithmetic and logic unit:** 2.1 Addition and Subtraction 2.1.1 Addition and Subtraction with Signed Magnitude 2.1.2 Hardware Implementation 2.1.3 Addition and Subtraction with Signed 2's Complement Data 2.2 Multiplication Algorithm 2.2.1 Signed operand multiplication 2.2.2 Booths algorithm and array multiplier. 2.3 Division Algorithms 2.3.1 Hardware Implementation for signed Magnitude data 2.3.3 Divide Overflow	**CO-2**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
data, including handling divide overflow situations.	2.4 Decimal Arithmetic Unit 2.5.1 Floating point arithmetic operation 2.5.2 Addition and Subtraction 2.5 IEEE Standard for Fixed Point and Floating Point Numbers	
TSO 3.a. Explain with example of an "I/O Interface" and its role in connecting peripheral devices to a computer. TSO 3.b. Explain the key differences between an "I/O Bus" and a "Memory Bus." TSO 3.c. Explain the concept of "Asynchronous Data Transfer" and its significance in computer communication. TSO 3.d. Explain the process of "Handshaking" in asynchronous communication and how it helps in reliable data transfer.	**Unit 3: Input-Output Organization** 3.1 Input Output Interface 3.1.1 I/O Bus and Interface Module 3.1.2 I/O versus Memory Bus 3.1.3 Example of I/O Interface 3.2 Asynchronous Data Transfer 3.2.1 Asynchronous Communication Interface 3.2.2 Handshaking 3.3 Modes of data transfer 3.3.1 Programmed I/O 3.3.2 Interrupt Initiated I/O 3.3.3 Direct Memory Access 3.3.3.1DMA Controller 3.3.3.2 DMA Transfer 3.4 Serial Communication 3.4.1 Character Oriented Protocol 3.4.2 Bit Oriented Protocol 3.5 Parallel Communication 3.6 Serial vs Parallel Communication	**CO-3**
TSO 4.a Explain the concept of "Memory Hierarchy" and explain its role in computer architecture. TSO 4.b. Explain the hardware organization of "Associative Memory" and how it enables fast data retrieval. TSO 4.c. Explain the principles of "Cache Memory" and differentiate between "Associative Mapping," "Direct Mapping," and "Set Associative Mapping." TSO 4.d. Analyze the concept and design issues associated with different cache memory mappings TSO 4.e. Evaluate impact of Cache Memory Design on system performance.	**Unit 4. Memory Organization:** 4.1 Basic concept and Memory Hierarchy 4.2 Types of Memory 4.2.1 Read Only Memory 4.2.2 Random Access Memory 4.2.3 Cache Memory 4.2.4 Register Memory 4.2.5 Auxiliary Memory(Magnetic Disk, Magnetic Tape) 4.3 Associative Memory 4.3.1 Hardware Organization 4.3.2 Match Logic 4.3.3 Read and Write Operation 4.4 Cache Memory 4.4.1 Associative Mapping 4.4.2 Direct Mapping 4.4.3 Set Associative Mapping 4.4.4 : Concept and design issues & performance 4.4.5 Calculate Hit Ratio, Miss Ratio and Performance 4.5 2D & 2 1/2D memory organization. 4.6 Virtual memory 4.7.1Address Space and Memory Space 4.7.2 Implementation of Virtual Memory 4.7.3 Page Replacement(FIFO,LRU)	**CO-4**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 5.a Explain the concept of "Pipelining" in the context of computer architecture. TSO 5.b. List the stages of a typical pipeline, including "Fetch," "Decode," "Execute," "Memory," and "Write Back." TSO 5.c Explain the concept of "Data Hazards" and "Structural Hazards" in the context of pipelining. TSO 5.d. Explain the concept of access time in the context of CPU performance, including its definition, importance, and factors affecting it.	**Unit 5. Pipelining and Vector Processing:** 5.1 Introduction to Pipelining 5.1.1 Definition of Pipelining 5.1.2 Need for Pipelining in modern processor 5.2 Pipeline Stages (Fetch, Decode, Execute, Memory, Write Back) 5.3 Pipelining Hazards (Structure Hazards, Data Hazards) 5.4 Performance Metrics of CPU (Access Time, EMAT (Effective Memory Access Time), Scale Factor	**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: (Not Applicable)

L) **Suggested Sessional Work and Self Learning: S2418404** 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.

1. Simulate a simple instruction set architecture (ISA) including basic instructions like addition, subtraction, conditional branches, and subroutine calls.
2. Explore different types of memory and their access times.
3. Design and implement a simple assembly program that demonstrates the execution of conditional branch instructions
4. Give a comparative study of 8085,8086 and NSC micro-processors

b. **Micro Projects:**

1. Prepare a market survey report on the application of different types of digital system.
2. Prepare a comparison chart on the technical specification and application of different types of memory, PLDs and CPLDS.
3. Develop a visual representation of memory hierarchy. Include various memory types such as cache memory, RAM, and auxiliary memory (e.g., magnetic disk).

c. **Seminar Topics:**

1. Computer Architecture for Edge Computing
2. RISC vs CISC Architecture
3. Windows vs Linux Architecture
4. Microprocessor 8085/8086 architecture

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**			
			Assignments	**Micro Projects**	**Other Activities***	
CO-1	15%	15%	15%	20%	20%	**NOT APPLICABLE**
CO-2	25%	25%	25%	20%	20%	
CO-3	25%	25%	25%	20%	20%	
CO-4	20%	20%	20%	20%	20%	
CO-5	15%	15%	15%	20%	20%	
Total Marks	**30**	**70**	**20**	**20**	**10**	
			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 Basic Computer Organization and Design	6	CO-1	11	5	3	3
Unit-2.0 Arithmetic and logic unit	12	CO-2	17	4	6	7
Unit-3.0 Input-Output Organization	14	CO-3	17	4	8	6
Unit-4.0 Memory Organization	8	CO-4	14	4	6	4
Unit-5.0 Pipelining and Vector Processing	8	CO-5	11	3	4	4
Total	**48**	**-**	**70**	**20**	**27**	**23**

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): (Not Applicable)**

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, Field Trips, 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 Session, Video Clippings, Use of Open Educational Resources (OER), MOOCs etc.

Q) List of Major Laboratory Equipment, Tools and Software: (Not Applicable)

R) Suggested Learning Resources:

(a) Books:

A) **Course Code** : **2418405 (T2418405/P2418405/S2418405)**

B) **Course Title** : Computer Troubleshooting & Maintenance

C) **Pre- requisite Course(s)** :

D) **Rationale** :

Computer Troubleshooting & Maintenance is frequently required for smooth functioning of computer system. The Objective of this subject is to make the students capable to understand the functioning of hardware parts and develop skills in diagnosing the faults and troubleshoots the computer system. This course will be helpful for students to get employment in the computer maintenance industry as well as self-employment.

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 Troubleshoot the computer system.

CO-2 Troubleshoot peripheral devices.

CO-3 Troubleshoot operating system, security threats and driver-related problems.

CO-4 Troubleshoot network and internet connectivity.

CO-5 Recover data and optimize system performance.

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	1	-	-	-	-	-	1		
CO-2	2	2	1	1	-	-	1		
CO-3	2	2	1	1	-	-	-		
CO-4	2	3	1	1	-	-	-		
CO-5	2	3	1	1	-	-	-		
CO-6	2	3	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 & Engg.	**2418405**	Computer Troubleshooting & Maintenance	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)	
Computer Science & Engg.	**2418405**	Computer Troubleshooting & Maintenance	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: T2418405

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
TSO 1a. Identify the hardware components found in different computer system. *TSO 1b.* Identify typical connections and necessary components of motherboard. *TSO 1c.* Recognize the common features of processors *TSO 1d.* Explain the architecture of multiple core processors *TSO 1e.* Describe the role of BIOS in a computer system *TSO 1f.* Configuring date and time, boot device priority, boot settings, and password security with BIOS settings. *TSO 1g.* Recognize the various memory module shapes, including DIP, ZIP, SIPP, SIMM, DIMM, and RIMM. *TSO 1h.* Explain the working of different types of memory modules (Dynamic RAM (DRAM), Synchronous DRAM (SDRAM), DDR SDRAM, SLDRAM, DRDRAM, Fast Page Mode (FPM) DRAM, and Extended Data Out (EDO)) DRAM. *TSO 1i.* Explain the working principles of an SMPS.	**Unit-1 Internal components of the PC** 1.1 **Computers**: Desktop Computers, Tablet, Laptop, Mainframe, Supercomputer. 1.2 **Features description**: Hardware components of Desktop Systems, Laptops, and Tablets. 1.3 **Motherboard:** Definition, parts and connections, and functioning block diagram 1.4 **Processor:** Common Features, Types of Processors (Microprocessor, Microcontroller etc.). Basic Structure of CPU, Different levels of cache, system bus, clock speed. a. Multiple Core Processors: Description, two core processor architecture and multi-core processor architecture 1.5 **BIOS:** Basic Input Output System Services, Bios Interaction, date and time, Boot device priority, boot setting configuration, password security. 1.6 **System Memory**: definition, memory sizes, speeds, and shapes (DIP, ZIP, SIPP, SIMM, DIMM, RIMM), Memory modules (Dynamic RAM, SDRAM, DDR SDRAM, SLDRAM, DRDRAM, Fast Page Mode (FPM) DRAM, Extended Data Out (EDO) DRAM). 1.7 **Power Supply**: Purpose and Features of SMPS, Working of SMPS Fault finding in power supply Uninterrupted Power Supply: Characteristics of UPS, Types of UPS, online and offline Preventive Maintenance of Power Supply	**CO-1**
TSO 2a. Explain keyboard and mouse operation and their interfaces. *TSO 2b.* Explain the working principles of scanners. *TSO 2c.* Explain the interfaces, components, and working principles of Ink-jet and LaserJet printers. *TSO 2d.* Explain various display technologies. *TSO 2e.* Identify components of graphics cards, including accelerated video cards. *TSO 2f.* Explain various hard disk interface technologies. *TSO 2g.* Explain various disk geometry parameters. *TSO 2h.* Describe disk performance characteristics, including hard disk controller. Differentiate different types of pen drives.	**Unit 2: Input Device, Output device, and Storage Devices** 2.1 **Keyboard**: Keyboard operation, Keyboard Type Keyboard interfaces 2.2 **Mouse:** Types, Operation, Interfaces 2.3 **Scanner:** Scanner Types, Image quality measurement, Working 2.4 **Printer:** Printer interface, Types of printers Ink-jet Printer: Parts, working Principle LaserJet Printer: Parts, working Principle. 2.5 **Monitor:** Video Basics (CRT parameters), VGA monitors Digital Display Technology- Thin Displays, Liquid Crystal Displays, Plasma Displays, Light Emitting Displays Graphics **Cards**: Components of a card, Accelerated Video cards, CGA,	**CO-2**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
	EGA, VGA 2.6 **Hard Disk Drive:** Disk Basics Hard Disk Interfaces: EIDE, Serial ATA, SCSI, USB, and IEEE 1394 (Firewire), RAID **Solid State Drive:** Types of SSD, Interface,Working Principal **Disk Geometry:** Heads, Tracks, Sectors, Cylinders, Cluster, Landing zone, MBR, Zone bit recording Disk performance Characteristics: Seeks and Latency, Data Transfer Rate Explain the working of hard disk controller **Hard Disk Controller:** Functional Blocks, HDC Functions **Pend Drive:** Types of pen drive, working principle of Pen drive.	
TSO 3a. Successfully install and set up an operating system. *TSO 3b.* Navigate and utilize the command-line interface efficiently. *TSO 3c.* Manage files and directories within the file system. *TSO 3d.* Create and manage user accounts with a focus on security best practices. *TSO 3e.* Establish network connectivity and troubleshoot network-related issues. *TSO 3f.* Perform software updates and stay up to date with the latest security patches. *TSO 3g.* Customize and personalize the operating system to suit specific preferences. *TSO 3h.* Effectively troubleshoot common operating system scenarios. *TSO 3i.* Install and configure device drivers for various hardware devices. *TSO 3j.* Install and manage plug and play device drivers efficiently.	**Unit 3: Software's, Malware, Security, and Drivers** 3.1 **Operating System** Installation and Setup, Command-Line Usage, File System Management, User Account and Security, Networking and Connectivity, Software Updates, Customization and Personalization, Troubleshooting Scenarios 3.2 Software Applications: Microsoft Word, Google Docs, Microsoft Excel, Google Sheets, Google Slides, Adobe Photoshop, Adobe Dreamweaver, WordPress 3.3 **Malware and Security** Basics definition of Malware, Viruses, and security threats. Methods of detecting and removing malware, viruses, and other security threats and protecting the system. 3.4 **Device Driver Software** Definition of device driver, Need and importance of driver, Installing and configuration of driver for various devices including plug and play types of drivers.	**CO-3**
TSO 4a. Explain the role of switches and routers in building and managing wired computer networks. *TSO 4b.* Explain the connectivity of switches using uplink ports and regular ports. *TSO 4c.* Explain the use of wireless router in the wireless network. *TSO 4d.* List different configuration-parameters in router. *TSO 4e.* Explain various types of internet	**Unit 4: Network and Internet Connectivity** 4.1 **Introduction to Networking Devices** Switches, Routers, Types of Switches, Connectivity of switch and up linking using uplink port and using regular port, Types of Routers, Connectivity of Router and Access point, Basic router-parameter-configuration 4.2 **Internet Connectivity** Types of internet connectivity, Broadband, Leased Line connection	**CO-4**

Major Theory Session Outcomes (TSOs)	Units	Relevant COs Number(s)
connections *TSO 4f.* Explain need and importance of hardware and software-based firewalls *TSO 4g.* Explain the role of proxy servers in internet connectivity	4.3 **Firewalls and Security:** Need for firewalls and their role in network security Types of Firewalls- hardware and software-based firewalls, Firewall Configuration and Management, Firewalls and Virtual Private Networks (VPNs), Packet Filtering, Proxy Servers	
TSO 5a. Identify common causes of data loss, including hardware failure, human error, and software issues. *TSO 5b.* Explain various data recovery methods, such as data retrieval from backups and data recovery software. *TSO 5c.* Differentiate between file recovery and disk recovery techniques. *TSO 5d.* Explore data backup strategies, including full, incremental, and differential backups. *TSO 5e.* Create and implement data backup plans to safeguard critical data. *TSO 5f.* Use data recovery software to retrieve lost or deleted files. *TSO 5g.* Define system performance metrics and their significance in evaluating a computer's performance. *TSO 5h.* Implement registry cleaning and optimization techniques to enhance system performance. *TSO 5i.* Optimize system performance by managing background tasks and processes efficiently.	**Unit 5: Data Recovery and System Performance Optimization** 5.1 **Data Loss Causes**: Human Error, Viruses & Malware, Power Outages, Hardware failure, Natural Disasters, Software Corruption, Migration errors 5.2 **Data Recovery Methods:** File Restore, Volume, Restore, Local virtualization, Cloud virtualization 5.3 **Data Backup Strategies: -** Types of Backups: Full backup, Incremental backup, Differential backup Direct-to-cloud backup, cloud-to-cloud backup, and SaaS backup 5.4 **System Performance Optimization:** System Performance Metrics, Registry Cleaning and Optimization Overheating and Cooling Solutions Hardware Upgrades for Performance Optimize the performance of system, Managing background tasks, and upgrading hardware components.	**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: P2418405

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 1.1. Identify the various hardware components of a desktop computer *LSO 1.2.* Set up different options in BIOS *LSO 1.3.* Identify different types of memory module LSO 1.4. Troubleshoot a PC's power supply using a multimeter.	1.	1.1 Identify the various hardware components of a desktop computer 1.2 Configure the following in BIOS setting i. power-saving settings ii. Adjust fan speeds and temperature thresholds iii. Enabling or disabling the display of POST error messages iv. Set a BIOS password for security v. Alter Booting 1.3 Identify different types of memory module 1.4 Troubleshoot a PC's power supply using a multimeter.	**CO-1**

Practical/Lab Session Outcomes (LSOs)	S. No.	Laboratory Experiment/Practical Titles	Relevant COs Number(s)
LSO 2.1. Troubleshoot common problems with peripherals like printers, scanners, monitors, keyboards, and mouse. *LSO 2.2.* Apply various memory testing tools for memory tests. *LSO 2.3.* Troubleshoot the internal components of a computer like CPU, RAM, hard drive, motherboard, and graphics card. *LSO 2.4.* Apply common tools like multimeters for testing power supplies and cables. *LSO 2.5.* Troubleshoot SMPS problem, cable connection problem, port problem, and interface problem.	2.	2.1 Troubleshoot common printer problem such as paper jams, driver problems, and connectivity. 2.2 Troubleshoot monitor problems by adjusting the display settings, checking cable connections, and replacing faulty components if necessary. 2.3 Identify overheating, faulty RAM, and CPU socket problems. 2.4 Reset the CPU and RAM modules, apply thermal paste, and test for stability. **2.5** Troubleshoot the given type of hard drive for loose or damaged cables, as well as signs of overheating or physical damage.	**CO-2**
LSO.3.1 Troubleshoot common problems with the operating system. LSO.3.2 Apply appropriate solutions for operating system recovery and repair options. LSO.3.3 Remove malware, viruses, and other security threats. LSO.3.4 Install antivirus and anti-malware for system security. LSO.3.5 Troubleshoot different types of conflicts in drivers.	3.	3.1 Fix errors in operating system like "Missing Operating System," "BOOTMGR is missing," and "NTLDR is missing. 3.2 Troubleshoot common problems with the operating system, like startup errors, system crashes, and performance problems. 3.3 Use recovery tools like Windows Startup Repair or Linux's GRUB to fix booting problems. 3.4 Install the given software and ensure proper functioning. 3.5 Use appropriate antivirus to remove malware. 3.6 Install drivers for various hardware components, ensuring compatibility and addressing any installation problems.	**CO-3**
LSO 4.1. Install layer-2 networking devices. *LSO 4.2.* Troubleshoot internet connectivity issue. *LSO 4.3.* Troubleshoot problems with different networking devices. *LSO 4.4.* Configure and manage firewalls for network security and threat protection.	4.	4.1 Install appropriate layer-2 networking device in the given situation. 4.2 Simulate the functioning of the router using packet tracer/Wireshark simulator. 4.3 Troubleshoot the internet connectivity problems. 4.4 Configure wireless access points. 4.5 Configure firewalls, enable Windows Defender, and update security settings.	**CO-4**
LSO 5.1. Apply tools and procedures to recover data from corrupted or damaged storage media. *LSO 5.2.* Set up data backup systems to protect critical data. LSO 5.3. Optimize system performance.	5.	5.1 Use data recovery software to retrieve deleted or corrupted files from the given storage device. 5.2 Recover data from a USB flash drive with corrupted or deleted files. 5.3 Set up a backup system (e.g., using backup software or cloud storage) to regularly back up the data. 5.4 Manage background processes to improve system performance by reducing startup programs or limiting running services.	**CO-5**

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: S2418405** 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:

i. **Project Description:** In this micro project on computer troubleshooting and maintenance, students will work on a project that involves ensuring the proper functionality and performance of a classroom computer lab. The project will include the following steps:

ii. **Initial Assessment**: Students will conduct an initial assessment of the computers in the classroom lab. This includes checking for any hardware or software issues, verifying that all computers are up to date, and assessing the overall performance of the machines.

iii. **Issue Identification**: Students will identify and document any issues or problems found during the assessment. This could include hardware malfunctions, software errors, connectivity issues, or any other anomalies.

iv. **Troubleshooting**: Based on the identified issues, students will plan and execute troubleshooting procedures. This may involve diagnosing hardware problems, resolving software conflicts, or addressing network connectivity issues.

v. **Maintenance Tasks**: Students will perform routine maintenance tasks on the computers, such as cleaning out dust, updating software, and optimizing system performance. They will also check for security updates and ensure that antivirus software is up to date.

vi. **Documentation**: Throughout the project, students will maintain detailed documentation of the issues identified, the troubleshooting steps taken, and the maintenance tasks performed.

vii. **Recommendations**: Students will provide recommendations for long-term maintenance and potential upgrades to enhance the lab's performance and reliability.

viii. **Testing and Validation**: After troubleshooting and maintenance, students will validate the effectiveness of their efforts by conducting tests on the computers to ensure they are running smoothly.

ix. **Final Report**: Students will compile their findings, actions taken, and recommendations into a final report to present to the class or instructors. The report should include a summary of the current state of the lab, the improvements made, and suggestions for ongoing maintenance.

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

c. Other Activities:

1. Identify and label various hardware components within a computer
2. Installing various operating systems, such as Windows, Linux, or macOS, and troubleshoot common installation issues like driver conflicts or disk partitioning errors.
3. Use antivirus and anti-malware tools to remove the threats while minimizing data loss.
4. Practice of cleaning unnecessary files, defragmenting the hard drives, and optimize system performance.
5. Maintenance of a computer lab, including cleaning, routine hardware checks, and ensuring that all software and systems are up to date.

d. **Self-learning topics**:

1. Learning how to diagnose and fix common OS-related problems, such as boot issues, system crashes, and software conflicts.

2. List common hardware failures like overheating, power supply issues, and hard drive failures.
3. Learn how to update, roll back, or reinstall device drivers to resolve hardware-related issues.
4. Exploring the BIOS/UEFI settings to troubleshoot boot problems, configure hardware, and update firmware.
5. Learning how to manage and troubleshoot storage devices, including disk partitioning, formatting, and file system errors.
6. List the tools and techniques for providing remote support and troubleshooting for remote users or clients.
7. Exploring how to upgrade hardware components like RAM, hard drives, graphics cards, and power supplies, including compatibility considerations.

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**.

	Course Evaluation Matrix						
	Theory Assessment (TA)**		**Term Work Assessment (TWA)**			**Lab Assessment (LA)#**	
COs	**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%	15%	10%	100%	100%	20%	20%
CO-2	10%	25%	10%			20%	20%
CO-3	10%	20%	15%			20%	20%
CO-4	25%	20%	25%			20%	20%
CO-5	25%	20%	20%			20%	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 (CI) Hours**	**Relevant COs Number(s)**	**Total Marks**	**ETA (Marks)**		
				Remember (R)	**Understanding (U)**	**Application & above (A)**
Unit 1.0 Internal Components of the PC	7	**CO-1**	10	4	3	3
Unit 2.0 Input Device, Output Device, and Storage Devices	11	**CO-2**	18	4	5	11
Unit3.0 Software's Malware, Security, and Drivers	10	**CO-3**	14	4	4	6

Unit 4.0 Network and Internet Connectivity	10	**CO-4**	14	5	3	7
Unit 5.0 Data Recovery and System Performance Optimization	10	**CO-5**	14	3	6	5
Total	**48**	**-**	**70**	**20**	**21**	**29**

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

O) Suggested AssessmentTable for Laboratory (Practical):

S. No.	**Laboratory Practical Titles**	**Relevant COs Number(s)**	**PLA/ELA**		
			Performance		**Viva-Voce (%)**
			PRA* (%)	**PDA** (%)**	
1.	1.1 Identify the various hardware components of a desktop computer 1.2 Configure the following in BIOS setting i. power-saving settings ii. Adjust fan speeds and temperature thresholds iii. Enabling or disabling the display of POST error messages iv. Set a BIOS password for security v. Alter Booting 1.3 Identify different types of memory module 1.4 A PC's power supply using a multimeter.	**CO-1**	40	50	10
2.	2.1 Troubleshoot common printer problem such as paper jams, driver problems, and connectivity. 2.2 Troubleshoot monitor problems by adjusting the display settings, checking cable connections, and replacing faulty components if necessary. 2.3 Identify overheating, faulty RAM, and CPU socket problems. 2.4 Reset the CPU and RAM modules, apply thermal paste, and test for stability. 2.5 Troubleshoot the given type of hard drive for loose or damaged cables, as well as signs of overheating or physical damage.	**CO-2**	40	50	10
3.	3.1 Fix errors in operating system like "Missing Operating System," "BOOTMGR is missing," and "NTLDR is missing. 3.2 Troubleshoot common problems with the operating system, like startup errors, system crashes, and performance problems. 3.3 Use recovery tools like Windows Startup Repair or Linux's GRUB to fix booting problems. 3.4 Install the given software and ensure proper functioning. 3.5 Use appropriate antivirus to remove malware. 3.6 Install drivers for various hardware components, ensuring compatibility and addressing any installation problems.	**CO3**	40	50	10
4.	4.1 Install appropriate layer-2 networking device in the given situation. 4.2 Simulate the functioning of the router using packet tracer/Wireshark simulator. 4.3 Troubleshoot the internet connectivity problems. 4.4 Configure wireless access points.	**CO-4**	40	50	10

S. No.	Laboratory Practical Titles	Relevant COs Number(s)	PLA/ELA		
			Performance		Viva-Voce (%)
			PRA* (%)	PDA** (%)	
	4.5 Configure firewalls, enable Windows Defender, and update security settings.				
5.	5.1 Use data recovery software to retrieve deleted or corrupted files from the given storage device. 5.2 Recover data from a USB flash drive with corrupted or deleted files. 5.3 Set up a backup system (e.g., using backup software or cloud storage) to regularly back up the data. 5.4 Manage background processes to improve system performance by reducing startup programs or limiting running services.	**CO-5**	40	50	10

Legend: