B.E. COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LERANING) DEGREE PROGRAMME

SCHEDULING OF COURSES
&

CURRICULUM AND DETAILED SYLLABI

FOR

FIRST SEMESTER TO SECOND SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2024-25 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

Thiagarajar College of Engineering, Madurai - 625015
Credit Distribution for B.E.CSE (AIML) Programme – 2024 – 2028 Batch

S.No	Category of Courses	Credits
		Regular Admission
A	**Foundation Courses(FC)**	**54 - 66**
	Humanities and Social Sciences including Management Courses (HSMC)	09 - 12
	Basic Science Courses (BSC)	24 - 27
	Engineering Science Courses (ESC)	21 - 27
B	**Professional Core Courses (PCC)**	**55**
C	**Professional Elective Courses (PEC)**	**24 - 39**
	Programme Specific Electives (PSE)	15 – 24
	ProgrammeElectives for Expanded Scope (PEES)	9 - 15
D	**Open Elective Courses (OEC)**	**6 - 12**
	Interdisciplinary Elective (IE)	3 - 6
	Basic Science Elective (BSE)	3 - 6
E	**Project work**	**12**
F	Internship and Mandatory Audit as per the Regulatory Authorities	
	Minimum credits to be earned for the award of the Degree	**160 (from A to E) and the successful completion of Mandatory Courses**

Thiagarajar College of Engineering, Madurai-625015
Department of Computer science and Engineering
B.E. Computer Science and Engineering (Artificial Intelligence and Machine Learning) Programme
Scheduling of Courses – for those join in the year 2024 – 2025

Sem	Theory / Theory cum Practical / Practical										Credit
	1	2	3	4	5	6	7	8	9	Audit Courses (Mandatory Non- credit}	
I	22MA110 Calculus for Engineers (BS-4)	23PH120 Physics (BS-3)	22CH130 Chemistry (BS-3)	22EG140 Technical English (HSS-2)	24AM150 Engineering Exploration (ES-2)	24AM160 Problem Solving and Programming (PC-3) TCP	22EG170 English Laboratory (HSS-1)	22PH180 Physics Laboratory (BS-1)	22CH190 Chemistry Laboratory (BS-1)		20
II	24AM210 Linear Algebra (BS-4)	22CS320 Theory and Design of Programming Languages (ES-3)	24AM230 Engineering Science Course# (ES -3)	24AM240 Introduction to Computer Systems (PC-3)	24AM250 Object Oriented Programming (PC -3)	22CS261 Engineering Graphics and Extended Reality (ES-3)	24AM270 Object Oriented Programming Lab (PC-1)			22CHAA0 Environmental Science	20
III	24AM310 Probability and Statistics (BS-4)	24AM320 Cognitive Science (ES-3)	24AM330 Artificial Intelligence (PC-3)	22CS340 Data Structures and Algorithms (PC-3)		22CS360 Operating Systems (PC-4) TCP	22CS370 Data Structures Lab (PC-1)	24AM380 Artificial Intelligence Lab (PC-1)	22ES390 Design Thinking (ES-3)		22
IV	24AM410 Discrete Mathematics (BS-4)	22CS420 Design and Analysis of Algorithms (PC-3)	22CS430 Data Communication and Networks (PC-3)	24AM440 Machine Learning (PC-3)	24AM450 Database Systems (PC-3) TCP	22EG660 Professional Communication (HSS-2)	24AM470 Machine Learning Lab (PC-1)	22CS480 Algorithms Lab (PC-1)	22CS490 Project Management (HSS-3)	Audit Course 2	23
V	24AM510 Modelling and Optimization (ES-3)	22CS520 Theory of Computation (ES-3)	24AM530 Deep Learning for NLP (PC-3)	24AMPX0 Programme Elective (PE - 3)	22XXGX0 Interdisciplinary Elective (OE-3)	24AM560 Autonomous Agents (PC-3)	22CS570 Network Programming Lab (PC-1)	24AM580 Autonomous Agents Lab (PC-1)	22CS590 Project – I (P-3)		23
VI	24AM610 Social and Ethical aspects of AI (HSS-3)	24AM620 Deep Learning for Computer Vision (PC-3)	24AM630 Human-AI Interaction (PC-3)	24AMPX0 Programme Elective (PE - 3)	22XXFX0 Basic Science Elective (OE-3)	24AMPX0 Programme Elective (PE - 3)	24AM670 Deep Learning Lab (PC-1)		22CS690 Project - II (P-3)		22
VII	24AM710 Behavioural and Emotional Intelligence (ES-2)	24AM720 Reinforcement Learning (PC-3)	24AMPX0 Programme Elective (PE - 3)	24AMPX0 Programme Elective (PE - 3)	24AMPX0 Programme Elective (PE - 3)	24AMPX0 Programme Elective (PE - 3)	24AM770 Reinforcement Learning Lab (PC-1)		22CS790 Project - III (P-3)		21
VIII	24AMPX0 Programme Elective (PE - 3)	24AMPX0 Programme Elective (PE - 3)							22CS890 Project - IV (P-3)		9

Passed in Board of Studies meeting on 27.04.2024

Approved in 67th Academic Council meeting on 25.05.2024

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI- 625 015
(A Govt. Aided, ISO 9001:2008 certified Autonomous Institution affiliated to Anna University)
Department of Computer Science and Engineering

B.E. Computer Science and Engineering (Artificial Intelligence and Machine Learning) Programme

Categorization of Courses

List of Humanities and Social Sciences including Management Courses (9-12)

- 22EG140 - Technical English (2)
- 22EG170 - English Laboratory (1)
- 22EG660 - Professional Communication (2)
- 22CS490 - Project Management (3)
- 24AM610 - Social and Ethical aspects of AI (3)

List of Basic Science Courses (24-27)

- 22MA110 - Calculus for Engineers (4)
- 24AM210 - Linear Algebra (4)
- 24AM310 - Probability and Statistics (4)
- 24AM410 - Discrete Mathematics (4)
- 23PH120 - Physics (3)
- 22CH130 - Chemistry (3)
- 22PH180 - Physics Laboratory (1)
- 22CH190 - Chemistry Laboratory (1)

List of Engineering Science Courses (21-27)

- 24AM150 - Engineering Exploration (2)
- 22CS320 – Theory and Design of Programming Languages (3)
- 24AM230 – Engineering Science Course[#] (3)
- 22CS261 - Engineering Graphics and Extended Reality (3)
- 24AM320 - Cognitive Science (3)
- 22ES390 - Design Thinking (3)
- 24AM510 – Modelling and Optimization (3)
- 22CS520 - Theory of Computation (3)
- 24AM710 - Behavioural and Emotional Intelligence (2)

List of Core Courses (55)

- 24AM160 - Problem solving and Programming (3) (TCP)
- 24AM240 - Introduction to Computer Systems (3)
- 24AM250 - Object Oriented Programming (3)
- 24AM270 - Object Oriented Programming Lab (1)
- 22CS340 - Data Structures and Algorithms (3)
- 24AM330 - Artificial Intelligence (3)
- 22CS360 - Operating Systems (4) (TCP)
- 22CS370 - Data Structures Lab (1)
- 24AM380 - Artificial Intelligence Lab (1)
- 22CS420 - Design and Analysis of Algorithms (3)
- 22CS430 – Data Communication and Networks (3)
- 24AM440 - Machine Learning (3)
- 24AM450 - Database Systems (3) (TCP)
- 24AM470 - Machine Learning Lab (1)
- 22CS480 - Algorithms Lab (1)

- 24AM530 - Deep Learning for NLP (3)
- 24AM560 - Autonomous Agents (3)
- 22CS570 - Network Programming Lab (1)
- 24AM580 - Autonomous Agents Lab (1)
- 24AM620 - Deep Learning for Computer Vision (3)
- 24AM630 - Human-AI Interaction (3)
- 24AM670 - Deep Learning Lab (1)
- 24AM720 - Reinforcement Learning (3)
- 24AM770 - Reinforcement Learning Lab (1)

Programme Elective Courses (24 – 39)

- Programme Electives (27)

Open Elective Courses (OEC): (6 – 12)

- Interdisciplinary Elective (3)
- Basic Science Elective (3)

Project (12)

- 22CS590 - Project – I (3)
- 22CS690 - Project - II (3)
- 22CS790 - Project – III (3)
- 22CS890 - Project – IV (3)

The course will be passed in next academic council meeting

B.E. COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE and MACHINE LEARNING) DEGREE PROGRAMME

CURRICULUM AND SYLLABI

FOR

FIRST SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2024 - 25 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
BECSE(AIML) Degree Programme

COURSES OF STUDY
(For the candidates admitted from 2024-25 onwards)

FIRST SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			credits
			L	T	P	
THEORY						
22MA110	Calculus for Engineers	BSC	3	1	-	4
23PH120	Physics	BSC	3	-	-	3
22CH130	Chemistry	BSC	3	-	-	3
22EG140	Technical English	HSMC	2	-	-	2
24AM150	Engineering Exploration	ESC	1	1	-	2
THEORY CUM PRACTICAL						
24AM160	Problem Solving and Programming	PC	2	-	2	3
PRACTICAL						
22EG170	English Laboratory	HSMC	-	-	2	1
22PH180	Physics Laboratory	BSC	-	-	2	1
22CH190	Chemistry Laboratory	BSC	-	-	2	1
Total			**14**	**2**	**8**	**20**

BSC : Basic Science Courses
HSMC : Humanities and Social Sciences including Management Courses
ESC : Engineering Science Courses

L : Lecture
T : Tutorial
P : Practical

Note:
1 Hour Lecture is equivalent to 1 credit
1 Hour Tutorial is equivalent to 1 credit
2 Hours Practical is equivalent to 1 credit

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
BECSE(AIML) Degree Programme

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2024-25 onwards)

FIRST SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	22MA110	Calculus for Engineers	3	40	60	100	27	50
2	23PH120	Physics	3	40	60	100	27	50
3	22CH130	Chemistry	3	40	60	100	27	50
4	22EG140	Technical English	3	40	60	100	27	50
5	24AM150	Engineering Exploration	3	40	60	100	27	50
THEORY CUM PRACTICAL								
6	24AM160	Problem Solving and Programming	3 (Terminal Exam Type : Theory)	50	50	100	22.5	50
PRACTICAL								
7	22EG170	English Laboratory	3	60	40	100	18	50
8	22PH180	Physics Laboratory	3	60	40	100	18	50
9	22CH190	Chemistry Laboratory	3	60	40	100	18	50

* CA evaluation pattern will differ from course to course and for different tests. This will have to be declared in advance to students. The department will put a process in place to ensure that the actual test paper follow the declared pattern.

** Terminal Examination will be conducted for maximum marks of 100 and subsequently be reduced to 50 marks for the award of terminal examination marks

24AM150 ENGINEERING EXPLORATION

Category	L	T	P	Credit
ES	1	1	-	2

Preamble

The course Engineering Exploration provides an introduction to the engineering field. It is designed to help the student to learn about engineering and how it affects our everyday lives. On the successful completion of the course, students will explain how engineering is different from science and technology and design basic combinational digital logic circuits.

Prerequisite

- NIL

Course Outcomes

On the successful completion of the course, students will be able to

CO	Course Outcome	TCE Proficiency Scale	Expected Proficiency	Expected Attainment
CO1	Explain technological & engineering development, change and impacts of engineering	TPS2	70	70
CO2	Complete initial steps (Define a problem, list criteria and constraints, brainstorm potential solutions and document the ideas) in engineering design process	TPS3	70	70
CO3	Communicate possible solutions through drawings, modelling and Testing Final output	TPS3	70	70
CO4	Explain the working principle of semiconductor devices and its applications	TPS2	70	70
CO5	Perform simplification of boolean logic functions by applying the theorems and postulates of Boolean algebra and Karnaugh map and Perform code conversion and signed number arithmetic	TPS3	70	70
CO6	Design combinational logic circuits for given specifications and implement them using logic gates	TPS3	70	70

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1.	M	L	-	-	-	-	-	-	L	L	-	L	L	-	L
CO2.	S	M	L	-	-	L	L	L	L	L	-	L	M	L	L
CO3	S	M	L	-	-	L	L	L	L	L	-	L	M	L	L
CO4	M	L	-	-	-	-	-	-	L	L	-	L	L	-	L
CO5	S	M	L	-	-	-	-	L	M	M	-	L	M	-	L
CO6	S	M	L	-	-	-	-	L	M	M	-	L	M	-	L

S- Strong; M-Medium; L-Low

Assessment Pattern

CO	Worksheet-1			Worksheet-2			CAT			Terminal (Theory)
TPS Scale	1	2	3	1	2	3	1	2	3	
CO1		10						5		10
CO2		5	10						10	20
CO3		5	20						10	20
CO4								5		10
CO5						25			10	20
CO6						25			10	20

Syllabus

What is Engineering: Engineering Requirement, Knowledge within Engineering disciplines, Engineering advancements **Engineering Design:** Problem definition, idea generation through brainstorming and researching, solution creation through evaluating and communicating, text/analysis, final solution and design improvement. **Defining problems and Brainstorming:** Researching design, sketching problem solving Communicating solution: Dimensioning orthographic drawing, perspective drawing **Modeling and Testing final output:** Product evaluation, reverse engineering, final project report.

Electronics Engineering: History and overview- Diodes ,Bipolar Junction Transistors - Integrated circuits -types-Applications -Storage cell using Transistors

Computer Engineering: Binary Number System :Binary Number system and its conversion to other number System-Complements, Signed Binary Numbers and Arithmetic- Boolean Algebra Theorems and Properties of Boolean Algebra-Digital Logic Gates and Logic Operations-Simplification of logic functions using Karnaugh Map Method.

Combinational Logic Circuits: Design and Implementation-Binary Adder, Subtractor and, decoder and encoder- Multiplexers and Demultiplexers

Text Books

1. Ryan A.Brown, Joshua W.Brown and Michael Berkihiser: “Engineering Fundamentals: Design, Principles, and Careers”, Goodheart-Willcox Publisher, Second Edition, 2014.
2. Malvino, A. P., & Bates, D. J. Electronic principles. McGraw-Hill/Higher Education,Ninth edition, 2020
3. M. Morris Mano, Micheal D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog, Pearson Education; Sixth edition,2018.

Course Contents and Lecture Schedule

Module	Topic	No. of Periods
1	**What is Engineering**	
1.1	Engineering Requirement, Knowledge within Engineering disciplines	1
1.2	Engineering advancements	1
2	**Engineering Design**	
2.1	Problem definition, idea generation through brainstorming and researching	1
2.2	solution creation through evaluating and communicating	1
2.3	text/analysis,	1
2.4	final solution and design improvement.	1
3	**Defining problems and Brainstorming**	
3.1	Researching design	1
3.2	sketching problem solving Communicating solution	1
3.3	Dimensioning orthographic drawing	1
3.4	perspective drawing	1
4	**Modeling and Testing final output**	
4.1	Product evaluation, reverse engineering	1
4.2	final project report.	1
5	**Electronics Engineering**	
5.1	History and overview	1
5.2	Diodes ,Bipolar Junction Transistors	**1**
5.3	Integrated circuits -types-Applications	1
5.4	Storage cell using Transistors	1
6	**Computer Engineering**:	
6.1	Binary Number System :Binary Number system and its conversion to other number System	1
6.2	number System-Complements	**1**
6.3	Signed Binary Numbers and Arithmetic	1
6.4	Boolean Algebra Theorems and Properties of Boolean Algebra	1
6.5	Digital Logic Gates and Logic Operations- Simplification of logic functions using Karnaugh Map	1
6.6	Combinational Logic Circuits: Design and Implementation-	1
6.7	Binary Adder, Subtractor decoder and encoder-	1
6.8	Multiplexers and Demultiplexers	1

Course Designer(s):

1. Dr.C.Senthil Kumar, Associate Professor/CSE cskcse@tce.edu

24AM160	PROBLEM SOLVING AND PROGRAMMING	Category	L	T	P	Credit
		PC	2	0	2	3

Preamble

The objective of this course is to learn problem solving methodologies and aspects of C programming. Problem solving in programming helps the students to gain more knowledge over coding and programming. These problem solving skills also help them to develop more skills in a person and build a promising career.

Prerequisite

- NIL

Course Outcomes

On the successful completion of the course, students will be able to

COs	Course Outcomes	TCE Proficiency Scale	Expected Proficiency in %	Expected Attainment Level %
CO1	Develop simple algorithms for arithmetic and logical problems.	TPS 2	B	85
CO2	Translate the algorithms to programs & execution	TPS 3	B	80
CO3	Implement conditional branching, iteration and recursion	TPS 3	B	80
CO4	Decompose a problem into functions and synthesize a complete program using divide and conquer approach	TPS 3	B	80
CO5	Implement arrays, pointers and structures to develop algorithms and programs	TPS 3	B	80
CO6	Design and Develop Solutions to problems using dynamic memory allocation and files	TPS 3	B	80

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C(	M	L	-	-	L	-	-	L	L	L	L	L	L	-	L
CO2.	S	M	L	-	L	-	-	L	L	L	L	L	M	-	L
CO3.	S	M	L	-	L	-	-	L	L	L	L	L	M	-	L
CO4.	S	M	L	-	L	-	-	L	L	L	L	L	M	-	L
CO5.	S	M	L	-	L	-	-	L	L	L	L	L	M	-	L
CO6.	S	M	L	-	L	L	L	L	L	M	L	L	M	L	L

S- Strong; M-Medium; L-Low

Assessment Pattern

CO	CAT1						CAT2						Assignment 1						Assignment 2						Terminal(Theory)					
TPS Scale	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**
CO1	10	10												20											5	5	-			
CO2		10	30												40											5	10			
CO3		10	30												40											5	15			
CO4							10	10	20												30				5	5	10			
CO5								10	20												30					5	15			
CO6								10	20												40					5	10			

***** Terminal examination should cover all Course Outcomes in the appropriate TPS Scale level.

Syllabus

Introduction to Programming: Introduction to components of a computer system- Idea of Algorithm - Representation of Algorithm: Flowchart - From algorithms to programs; source code, variables and memory locations- Syntax and Logical Errors in compilation-debugging and testing-object and executable code.

Control Structures: Arithmetic expressions and precedence- Conditional Branching and Loops- Arrays: Arrays (1-D, 2-D), Character arrays and Strings

Basic Algorithms: Searching- Basic Sorting Algorithms (Bubble and Selection)- Functions using built in libraries- Parameter passing in functions- call by value- Passing arrays to functions: idea of call by reference

Recursion: Recursion-a different way of solving problems -Example programs: Factorial, Fibonacci series, Towers of Hanoi. Structure: Structures, defining structures and Array of Structures. **Pointers**: Idea of pointers- Defining pointers, Use of Pointers in self-referential structures, notion of linked list (no implementation).

Dynamic Memory Allocation: Allocation of a Block of memory: malloc, allocating multiple blocks of memory: calloc, memory leak, releasing the used space: free, Altering the size of a block: realloc. Concept of Link list.

File Processing: Defining and Opening a file, closing a file, input/output operations on files, error handling during I/O operations, Command Line Arguments.

Text Book

1. Jeri R. Hanly, and Elliot B. Koffman, Problem Solving and Program Design in C, Pearson Education India, 2015.
2. Byron Gottfried, Programming with C, McGraw Hill International Edition, Fourth Edition, 2018.

Reference Books & web resources

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language 2e, Pearson Education India, 2015.
2. Behrouz A. Forouzan and Richard F. Gilberg, Computer Science: A Structured Programming Approach Using C, Course Technology Inc, Third Edition , 2006.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Periods
1	**Introduction to Programming**	
1.1	Introduction to components of a computer system- Idea of Algorithm	1
1.2	Representation of Algorithm: Flowchart - From algorithms to programs	2
1.3	Source code, variables and memory locations- Syntax and Logical Errors in compilation- debugging and testing- object and executable code	1
2	**Control Structures**	
2.1	Arithmetic expressions and precedence	1
2.2	Conditional branching and loops	2
2.3	Arrays: Arrays (1-D, 2-D), Character arrays and Strings	2
3	**Basic Algorithms**	
3.1	Searching- Basic Sorting Algorithms (Bubble and Selection)	2
3.2	Functions: Functions using built in libraries- Parameter passing in functions- call by value- Passing arrays to functions: idea of call by reference	2
4	**Recursion**	
4.1	Recursion-a different way of solving problems - Example programs: Factorial, Fibonacci series, Towers of Hanoi	2
4.2	Structure: Structures, Defining structures and Array of Structures.	2
4.3	**Pointers**: Idea of pointers- Defining pointers, Use of Pointers in self-referential structures, notion of linked list (no implementation)	2
5	**Dynamic Memory Allocation**	
5.1	Allocation of a Block of memory: malloc, allocating multiple blocks of memory: calloc, memory leak - releasing the used space: free, Altering the size of a block: realloc. Concept of Link list.	2
6	**File Processing**	
6.1	Defining and Opening a file, closing a file, input/output operations on files,	1
6.2	Error handling during I/O operations, Command Line Arguments.	2
	Total	24

List of Experiments for Laboratory

1. Write a C program to generate the first n terms of the Fibonacci sequence.
2. Write a C program to generate prime numbers from 1 to n.
3. Write a C program to check whether given number is Armstrong Number or not.
4. Write a C program to perform arithmetic operations using switch statement.
5. Write a C program to find factorial of a given integer using recursive and non-recursive function.
6. Write C program to find GCD of two integers by using recursive function.
7. Write a C program to Sort the Array in an Ascending Order
8. Write a C program to find whether given matrix is symmetric or not.
9. Write a C program to perform matrix manipulation.
10. Develop a C Program for user defined Data Type: Structure
11. Illustrate C pre-processor directives through programming.
12. Simple Application Development

Course Designer(s):

1. Dr.M.P.Ramkumar,
   Associate Professor, CSE Department,
   ramkumar@tce.edu

B.E. COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE and MACHINE LEARNING) DEGREE PROGRAMME

CURRICULUM AND SYLLABI

FOR

SECOND SEMESTER

FOR THE STUDENTS ADMITTED IN THE

ACADEMIC YEAR 2024 - 25 ONWARDS

THIAGARAJAR COLLEGE OF ENGINEERING
(A Government Aided Autonomous Institution affiliated to Anna University)
MADURAI – 625 015, TAMILNADU

Phone: 0452 – 2482240, 41
Fax: 0452 2483427
Web: www.tce.edu

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
BECSE(AIML) Degree Programme

COURSES OF STUDY
(For the candidates admitted from 2024-25 onwards)

SECOND SEMESTER

Course Code	Name of the Course	Category	No. of Hours / Week			credits
			L	T	P	
THEORY						
24AM210	Linear Algebra	BSC	3	1	-	4
22CS320	Theory and Design of Programming Languages	ESC	3	-	-	3
24AM230	Engineering Science Course#	ESC	3	-	-	3
24AM240	Introduction to Computer Systems	PC	3	-	-	3
24AM250	Object Oriented Programming	PC	3	-	-	3
THEORY CUM PRACTICAL						
22CS261	Engineering Graphics and Extended Reality	ESC	2	-	2	3
PRACTICAL						
24AM270	Object Oriented Programming Lab	PC	-	-	2	1
AUDIT COURSE						
22CHAA0	Environmental Science	AC				-
Total			**17**	**1**	**4**	**20**

BSC : Basic Science Courses
ESC : Engineering Science Courses
PC : Professional Core Courses

L : Lecture
T : Tutorial
P : Practical

Note:
1 Hour Lecture is equivalent to 1 credit
1 Hour Tutorial is equivalent to 1 credit
2 Hours Practical is equivalent to 1 credit

The course will be passed in next academic council meeting

THIAGARAJAR COLLEGE OF ENGINEERING: MADURAI – 625 015
BECSE(AIML) Degree Programme

SCHEME OF EXAMINATIONS
(For the candidates admitted from 2024-25 onwards)

SECOND SEMESTER

S.No.	Course Code	Name of the Course	Duration of Terminal Exam. in Hrs.	Marks			Minimum Marks for Pass	
				Continuous Assessment *	Terminal Exam **	Max. Marks	Terminal Exam	Total
THEORY								
1	24AM210	Linear Algebra	3	40	60	100	27	50
2	22CS320	Theory and Design of Programming Languages	3	40	60	100	27	50
3	24AM230	Engineering Science Course #	3	40	60	100	27	50
4	24AM240	Introduction to Computer Systems	3	40	60	100	27	50
5	24AM250	Object Oriented Programming	3	40	60	100	27	50
THEORY CUM PRACTICAL								
6	22CS261	Engineering Graphics and Extended Reality	3 (Terminal Exam Type : Theory)	50	50	100	22.5	50
PRACTICAL								
7	24AM270	Object Oriented Programming Lab	3	60	40	100	18	50

* CA evaluation pattern will differ from course to course and for different tests. This will have to be declared in advance to students. The department will put a process in place to ensure that the actual test paper follow the declared pattern.

** Terminal Examination will be conducted for maximum marks of 100 and subsequently be reduced to 50 marks for the award of terminal examination marks

The course will be passed in next academic council meeting

		Category	L	T	P	Credit
24AM210	**LINEAR ALGEBRA**	BS	3	1	0	4

Preamble

Linear algebra and Matrices are essential tools for most algorithms in artificial intelligence and machine learning. This course introduces the idea of solving systems of linear and non-linear equations. Students learn to use the fundamental notion of vectors, vector space, linear independent, spanning, and basis. Also, it provides about linear transformations, the matrix of linear transformation, Eigen values, and Eigen vectors. They can perform approximations and orthogonal projections and construct an orthonormal basis for vector spaces. Moreover, this course demonstrates the various techniques for decomposition of matrices and principal component analysis.

Prerequisite

- Nil

Course Outcomes

On the successful completion of the course, students will be able to

		TCE Proficiency Scale	Expected Proficiency in %	Expected Attainment Level %
CO1	Solve linear and non-linear systems of equations using direct and iterative methods.	TPS3	B	60
CO2	Demonstrate vector space and subspace, Identify the basis, row space, column space, null space, and dimension of a vector space.	TPS3	B	60
CO3	Compute an orthonormal basis of an inner product space for a given basis	TPS3	B	60
CO4	Apply matrix algebra techniques to transform a matrix into a diagonalizable matrix	TPS3	B	60
CO5	Decompose a matrix using LU, Singular Value Decomposition, and QR factorization method	TPS3	B	60
CO6	Perform dimensionality reduction on the given data using Principal Component Analysis (PCA)	TPS3	B	60

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO	S	M	L		M			L	M	L		M	M	L	L
CO2.	S	M	L		M			L	M	L		M	M	L	L
CO3.	S	M	L		M			L	M	L		M	M	L	L
CO4	S	M	L		M			L	M	L		M	M	L	L
CO5	S	M	L		M			L	M	L		M	M	L	L
CO6	S	M	L		M			L	M	L		M	M	L	L

S- Strong; M-Medium; L-Low

Assessment Pattern

CO	Assessment 1		Assessment 2		Terminal (%)
	CAT 1 (%)	Assignment 1 (%)	CAT 2 (%)	Assignment 2 (%)	

TPS	1 10	2 30	3 60	1 -	2 -	3 100	1 10	2 30	3 60	1 -	2 -	3 100	1 -	2 30	3 70
CO1	-	10	17	-	-	25	-	-	-	-	-	-		3	8
CO2	7	10	20	-	-	38	-	-	-	-	-	-		6	12
CO3	3	10	23	-	-	37	-	-	-	-	-	-		6	12
CO4	-	-	-	-	-	-	10	20	20	-	-	50		12	14
CO5	-	-	-	-	-	-	-	5	20	-	-	25		-	12
CO6	-	-	-	-	-	-	-	5	20	-	-	25		3	12
TOTAL	10	30	60	-	-	100	10	30	60	-	-	100	-	30	70

Syllabus

System of linear Equations: System of linear equation - Row Echelon forms - Gauss Elimination. Nonlinear systems: System of Non-Linear Equations- Fixed point iteration- Newton Raphson Method-Gauss Jordan Method.

Vector Spaces: Definition and Examples- Subspaces, Linear Independence, Basis and Dimension, Row space and Column spaces.

Orthogonality: The scalar Product in R^n - Orthogonal Subspaces- Inner Product spaces, Orthonormal sets - Gram-Schmidt process.

Linear Transformations: Linear Transformations-Definitions and Examples - Matrix Representations of Linear Transformations -Similarity. Eigen systems: Eigen values of Matrices and Eigen vectors of Matrices–Diagonalization of Matrices – Quadratic Forms.

Decomposition and PCA: LU Decomposition, QR-Factorization- Singular Value Decomposition – Generalized Inverse (Pseudo Inverse)- Principal Component Analysis.

Text Books

1. Steven J. Leon., “Linear Algebra with Application” Ninth Edition, Pearson, 2015.
2. Steven C. Chapra & Raymond P. Canale., Numerical Methods for Engineers. 7th Edition, McGraw Hill Publications. 2015.
3. Gilbert Strang., “Introduction to Linear Algebra” 5th Edition, 2016

Reference Books & web resources

1. David C. Lay., “Linear Algebra and Its Applications” 5th Edition, 2015.
2. Richard Bronson., “Schaum’s Outline of Matrix Operations” Second Edition, McGraw Hill Publications. 2015.

Course Contents and Lecture Schedule

Module No.	Topic	No. of. Lectures
1	**LINEAR AND NONLINEAR SYSTEMS**	
1.1	System of linear equations, Row Echelon form	1
1.2	Gauss Elimination	1
1.3	System of Non-Linear Equations, Fixed point iteration	1
	Tutorial	1
1.4	Newton Raphson Method, Gauss Jordan Method	1
	Tutorial	1
	Program Using Python	
		(6)
2	**VECTOR SPACES**	
2.1	Vector space-Definition and Examples	2
2.2	Subspaces	1
	Tutorial	1

2.3	Linearly Independence	1
2.4	Basis and Dimension	1
	Tutorial	1
2.5	Row space and Column space	1
	Tutorial	1
	Program Using Python	
		(9)
3	**ORTHOGONALITY**	
3.1	The scalar Product in $\mathbf{R^n}$	1
3.2	Orthogonal Subspaces	1
	Tutorial	1
3.3	Inner Product spaces	1
3.4	Orthonormal sets	1
	Tutorial	1
3.5	Gram-Schmidt process	2
	Tutorial	1
	Program Using Python	
		(9)
4	**LINEAR TRANSFORMATION & EIGEN SYSTEMS**	
4.1	Linear Transformation- Definition and Examples	1
4.2	Matrix Representations of Linear Transformations	2
	Tutorial	1
4.3	Similarity Transformation	1
4.4	Eigenvalues of Matrices and Eigen vectors of Matrices	2
	Tutorial	1
4.5	Diagonalization of Matrices	2
4.6	Quadratic Forms	1
	Tutorial	1
	Program Using Python	
		(12)
5	**MATRIX DECOMPOSITION AND PCA**	
5.1	LU Decomposition	2
5.2	QR factorisation	1
	Tutorial	1
5.3	The Single Value Decomposition (SVD) of Matrices	2
	Tutorial	1
5.6	Pseudo Inverse	2
5.7	Principal Component Analysis	2
	Tutorial	1
	Program Using Python	
		(12)
	Total	**48**

Course Designer(s):

1. Dr. P. Victor pvmat@tce.edu
2. Dr. P. Krishnapriya pkamat@tce.edu
3. Dr. M. Sundar msrmat@tce.edu

24AM240	INTRODUCTION TO COMPUTER SYSTEMS	Category	L	T	P	Credit
		PC	3	0	0	3

Preamble

This course provides an overview of computer systems, focusing on their architecture, components, operation, and interaction. Topics include hardware components, software systems, data representation, computer organization, Real-world applications and significance in modern computing

Prerequisite

-Nil-

Course Outcomes

On the successful completion of the course, students will be able to

CO	Course Outcomes	TCE Proficiency Scale	Expected Proficiency in %	Expected Attainment Level %
CO1	Comprehend the basis operations that happen in a CPU	TPS 2	B	80
CO2	Sequence the data path and control path implementation	TPS 3	B	75
CO3	Customise code to support pipelining and improve performance	TPS 3	B	75
CO4	Apply knowledge of data representation and digital logic	.TPS 3	B	75
CO5	Categorize the memory hierarchy design and I/O design	TPS 3	B	75
CO6	Relate applications and algorithms for computing enhancements	TPS 3	B	75

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	P
C(	M	L						L	L	L		L	L		L
CO2.	S	M	L					L	L	L		L	M		L
CO3	S	M	L					L	L	L		L	M		L
CO4	S	M	L					L	L	L		L	M		L
CO5	S	M	L					L	L	L		L	M		L
CO .	S	M	L		L	L		L	L	L		L	M	L	L

S- Strong; M-Medium; L-Low

Assessment Pattern

CO	CAT1						CAT2						Assignment 1						Assignment 2						Terminal					
TPS Scale	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**
CO1	10	10												20											5	5				
CO2		20	20												40											5	10			
CO3		20	20												40											5	10			
CO4							5	15	20												40					10	20			
CO5							10	10	20												40					5	10			
CO6							5	5	10												20					5	10			

Syllabus

Computer System Fundamentals

Motivation, IAS Computer Structure and Operation, Functional Units, Basic concepts

Computer Function and Interconnection

Top level view of components and functions, Instruction cycle and program execution, Interrupts and instruction cycles, Multiple interrupts, Interconnection structures, Bus interconnection, Multiple buses, Synchronous and asynchronous bus timings

Pipelining: Basic Concepts, Data Hazards, Instruction hazards, Influence on Instruction Sets, Data path and Control Considerations, Superscalar Operation, Performance Considerations.

Memory Design

Characteristics and hierarchy of memory, Cache memory principles and operation, Cache design and mapping functions, replacement algorithms, main memory, DRAM and SRAM, Types of ROMs, Module organization, Introduction to magnetic disks

I/O Systems

I/O transfer and disk performance, Interrupt driven and DMA transfers, Performance estimation and trade-offs in design.

Computer Arithmetic

Arithmetic and Logic Unit, Integer multiplication of unsigned and signed numbers, Booth's algorithm, Division of unsigned binary, Floating point arithmetic

GPU Design Principles

Need and opportunities, CPU, GPU, GPU Architecture – Overview of hardware and software technology, AI co processors, Spectral functions, Case studies on different architecture, evaluation with performance metrics, sustainability, energy efficiency

Text Book

1. William Stallings, Computer Organization and Architecture Designing for Performance, Eleventh edition, Prentice Hall, 2013.
2. Andrew S Tanenbaum and Todd Austin, Structured Computer Organization, Sixth edition, Pearson, 2013.
3. Carl Hamacher, Computer Organization and Embedded Systems, Sixth edition, McGrawHill, 2012.
4. Computer Organization and Design" by David A. Patterson and John L. Hennessy
5. Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig.

Reference Books & web resources

1. DodiyaTripti, Computer Organisation and Advanced Microprocessors, First edition, Cengage Learning India,2012.
2. Barry B.Brey, The Intel Microprocessors Architecture Programming and Interfacing, Eighth edition, Pearson Prentice Hall, 2009.
3. N.Senthil Kumar, M.Saravanan and S. Jeevananthan, Microprocessors and Microcontrollers, First edition, Oxford University Press, 2010.
4. Charles H. Roth, Jr., “Fundamentals of Logic Design”, Jaico Publishing House, Mumbai,Fourth Edition 1992.
5. David A. Patterson and John L. Hennessy, “Computer Organization and Design: The Hardware/Software Interface”, Second Edition.
6. https://www.spectroscopyonline.com/view/an-interview-with-ai-about-its-potential-role-in-vibrational-and-atomic-spectroscopy

Course Contents and Lecture Schedule

Module No.	Topic	No. of Periods	CO mapping
	Introduction to the course and course outcomes		
1	**Computer System Fundamentals (5)**		
1.1	Organisation and system architecture	1	CO1
1.2	Structures and functions	1	CO1
1.3	History and challenges evolution of computer and technology	1	CO1
1.4	IAS computer design and working	1	CO1
1.5	Measures of performance	1	CO1
2	**Computer Function and Interconnection (6)**		
2.1	Computer components and functions	1	CO2
2.2	Interconnection structures	1	CO2
2.3	Instruction cycle and program execution	1	CO2
2.4	Multiple buses	1	CO2
2.5	Synchronous and asynchronous bus timings	1	CO2
2.6	Interrupts and handling mechanism	1	CO2
3	**Pipelining (4)**		
3.1	Basic Concepts of pipelining	1	CO3
3.2	Hazards and its impact	1	CO3
3.3	Data path and Control Considerations	1	CO3
3.4	Superscalar Operations and performance Considerations	1	CO3
4	**Computer Arithmetic (5)**		

Module No.	Topic	No. of Periods	CO mapping
4.1	Arithmetic and Logic Unit	1	CO4
4.2	Integer multiplication of unsigned and signed numbers	1	CO4
4.3	Booth's algorithm	1	CO4
4.4	Division of unsigned binary	1	CO4
4.5	Floating point arithmetic	1	CO4
5	**MEMORY DESIGN (6)**		
5.1	Memory system overview and layout	1	CO5
5.2	Characteristics and hierarchy of memory	1	CO5
5.3	Cache memory principles and operation	1	CO5
5.4	Cache design its mapping functions and replacement algorithms	1	CO5
5.5	main memory, DRAM and SRAM	1	CO5
5.6	, Introduction to magnetic disks	1	CO5
6	**I/O Systems (5)**		
6.1	I/O transfer and disk performance	1	CO5
6.2	Programmed I/O	1	CO5
6.3	Interrupt driven I/O	1	CO5
6.4	DMA transfers	1	CO5
6.5	Performance estimation and trade-offs in design.	1	CO5
7	**GPU Design Principles (5)**		
7.1	Need and opportunities of CPU and GPU	1	CO6
7.2	GPU Architecture – Overview of hardware and software	1	CO6
7.3	Field programmable gates, AI co processors,	1	CO6
7.4	AI for Spectral functions, ML framework-Pytorch	1	
7.5	Case studies on different sustainable, energy efficiency system architecture and evaluation with performance metrics	1	CO6
	Total	36	

Course Designer(s):

1. K.NarasimhaMallikarjunan Associate Professor, CSE arjunkambaraj@tce.edu

24AM250	OBJECT ORIENTED PROGRAMMING	Category	L	T	P	Credit
		PC	3	0	0	3

Preamble

This syllabus is intended for BECSE(AIML) students and enables them to learn Object Oriented Programming and the design of computer solutions in a precise manner. The syllabus emphasizes OOP concepts, Functions, Polymorphism, Inheritance, I/O and Patterns. The intention is to provide sufficient depth in these topics to enable students to apply Object Oriented approach to programming. The modules in the syllabus reflect solving general problems via programming solutions. Thus, modules collectively focus on programming concepts, strategies and techniques; and the application of these toward the development of programming solutions.

Prerequisite

Programming fundamentals

Course Outcomes

Upon the successful completion of the course, students will be able to

CO	Course Outcome	TCE Proficiency Scale	Expected Proficiency in %	Expected Attainment Level %
CO1	Construct Object oriented programs using methods and passing arrays, objects, and array of objects to them.	TPS3	B	80
CO2	Demonstrate Compile-time and Run-time polymorphism using object oriented programs.	TPS3	B	80
CO3	Illustrate the relationships between objects using inheritance hierarchies and aggregation	TPS3	B	80
CO4	Develop Object Oriented programs to handle exceptions	TPS3	B	80
CO5	Develop Object Oriented programs to handle data using Java collections, Files and Object Serialization	TPS3	B	80
CO6	Develop Object Oriented programs to demonstrate event driven programming, concurrent programming.	TPS3	B	80

Mapping with Programme Outcomes

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L	L	L	-	-	L	L	L	L	L	M	L	L
CO2	S	M	L	L	L	-	-	L	L	L	L	L	M	L	L
CO3	S	M	L	L	L	-	-	L	L	L	L	L	M	L	L
CO4	S	M	L	L	L	-	-	L	L	L	L	L	M	L	L
CO5	S	M	L	L	L	-	-	L	L	L	L	L	M	L	L
CO6	S	M	L	L	L	-	-	L	L	L	L	L	M	L	L

S- Strong; M-Medium; L-Low

Assessment Pattern

CO	CAT1						CAT2						Assignment		Terminal					
													1	2						
TPS Scale	1	2	3	4	5	6	1	2	3	4	5	6	3	3	1	2	3	4	5	6
CO1	10		10										20		2	4	10			
CO2	10	10	10										40		2	4	10			
CO3	10	20	20										40		4	4	10			
CO4							10		10					30	4	4	10			
CO5							10	10	10					30	4	4	10			
CO6							10	20	20					40	4		10			

***** Terminal examination should cover all Course Outcomes in the appropriate TPS Scale level.

Syllabus

Basics of Object Oriented Programming Object oriented programming and its benefits - Object oriented programming concepts: Encapsulation, Information hiding and Abstraction – Generalization/Specialization and Polymorphism - Object oriented design: finding the Classes and their Responsibilities – Object oriented programming language: Java and Python

Classes and Objects Instance fields and Methods-Constructors–Passing Arguments to a Method – Returning Value from a Method – Method overloading –Constructor overloading-Passing Arrays as Arguments to Methods – Passing Objects to Methods- Returning Objects from Methods

Class collaborations and Polymorphism Object Oriented Design: Class Collaborations – Aggregation –Composition –Chains of Inheritance – Overriding Super class methods – Abstract Classes and Abstract Methods – Interfaces

I/O Handling and Exception Handling – Binary files – Random-Access files- Object serialization – Exception handling

Collection Framework – Introduction to collections, Array List, Vector, Hash table, Stack, List, Tuple, Set, and Dictionary,

Event-Driven Programming Concurrent Programming Network programming – Text-related GUI components – other GUI components – Handling mouse events and button events – Thread life cycle and methods – Runnable interface – Thread Synchronization

Text Book

1. Herbert Schildt: “Java: The Complete Reference”, Twelfth Edition, McGraw-Hill,

2021.
2. Tony Gaddis, “Starting Out with Java: From Control Structures through Objects”, Sixth edition, Pearson Education Limited, 2016.
3. Bart Baesens, Aimee Backiel, SeppevandenBroucke, “Beginning Java Programming:The Object-Oriented Approach”, John Wiley & Sons, 2015.
4. Allen B. Downey, “Think Python: How to Think like a Computer Scientist”, 2nd Edition, O’Reilly Publishers, 2016. Grady Booch, Robert Maksimchuk, Michael Engel, Bobbi Young, Jim Conallen, Kelli Houston“ Object Oriented Analysis and Design with Applications”, Third Edition, 2012.

Course Contents and Lecture Schedule

Module No.	Topic	No. of Hours
1	**Basics of Object oriented Programming (6)**	
1.1	Object oriented programming and its benefits	1
1.2	Object oriented programming concepts: Encapsulation, Information hiding and Abstraction, Generalization/Specialization and Polymorphism	1
1.3	Object oriented design: finding the Classes and their Responsibilities	2
1.4	Object oriented programming language: Java	1
1.5	Object oriented programming language: Python	1
2	**Classes and Objects (5)**	
2.1	Instance fields and Methods-Constructors	1
2.2	Passing Arguments to a Method – Returning Value from a Method - Method overloading	1
2.3	Constructor overloading	1
2.4	Passing Arrays As Arguments to Methods	1
2.5	Passing Objects to Methods, Returning Objects from Methods	1
3	**Class collaborations and Polymorphism (6)**	
3.1	Object oriented Design: Class Collaborations	1
3.2	Aggregation –Composition	1
3.3	Chains of Inheritance	1
3.4	Overriding Super class methods	1
3.5	Abstract Classes and Abstract Methods	1
3.6	Interfaces	1
4	**I/O Handling and Exception Handling (5)**	
4.1	Binary files	1
4.2	Random-Access files	1
4.3	Object serialization	1
4.4	Exception handling	2
5	**Collection Framework (7)**	
5.1	Introduction to collections	1
5.2	Array List, Vector	2
5.3	Hash table, Stack	2
5.4	List, Tuple, Set, and Dictionary	2

6	**Event-Driven Programming Concurrent Programming Network programming (7)**	
6.1	Frameworks	1
6.2	Text-related GUI components, other GUI components	1
6.3	Handling mouse events and button events	1
6.4	Thread life cycle and methods	1
6.5	Runnable interface	1
6.6	Thread Synchronization	1

Course Designer(s):

1. Dr M.Vijayalakshmi, Professor, CSE – mviji@tce.edu

24AM270	OBJECT ORIENTED PROGRAMMING LAB	Category	L	T	P	Credit
		PC	0	0	2	1

Preamble

This syllabus is intended for the BECSE(AIML) students and enables them to learn Object Oriented Programming and the design of computer solutions in a precise manner. The experiments emphasize on OOP concepts, Functions, Polymorphism, Inheritance, I/O, event- driven, concurrent and network programming. The intention is to provide sufficient depth in these topics to enable candidates to apply Object Oriented Programming approach to programming.

Prerequisite

Programming Fundamentals

Course Outcomes

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

CO	Course Outcome	TCE Proficiency Scale	Expected Proficiency in %	Expected Attainment Level %
CO1	Construct Object oriented programs using methods and passing arrays, objects, and array of objects to them.	TPS3	B	80
CO2	Demonstrate Compile-time and Run-time polymorphism using object oriented programs.	TPS3	B	80
CO3	Illustrate the relationships between objects using inheritance hierarchies and aggregation	TPS3	B	80
CO4	Develop Object Oriented programs to handle exceptions	TPS3	B	80
CO5	Develop Object Oriented programs to handle data using Java collections, Files and Object Serialization	TPS3	B	80
CO6	Develop Object Oriented programs to demonstrate event driven programming, concurrent programming.	TPS3	B	80

Mapping with Programme Outcomes and Programme Specific Outcomes

Cos	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	S	M	L	M	M	M		M	M	M	M	M	M	M	M
CO2	S	M	L	M	M	M		M	M	M	M	M	M	M	M
CO3	S	M	L	M	M	M		M	M	M	M	M	M	M	M

CO 4	S	M	L	M	M	M		M	M	M	M	M	M	M	M
CO 5	S	M	L	M	M	M		M	M	M	M	M	M	M	M
CO 6	S	M	L	M	M	M		M	M	M	M	M	M	M	M

S- Strong; M-Medium; L-Low

Assessment Pattern: Cognitive Domain

Cognitive Levels	Model Examination	Terminal Examination
Remember		
Understand		
Apply	100	100
Analyse		
Evaluate		
Create		

Assessment Pattern: Psychomotor

	Experiment	CO
1.	Develop Object Oriented Program for passing arguments to a method and returning value from a method using Java and Python	CO1
2.	Develop Object Oriented Program for passing arrays and objects as arguments to method and returning objects from methods using Java and Python	CO1
3.	Construct Object Oriented Program for method overloading and constructor overloading using Java and Python	CO2
4.	Demonstrate aggregation and composition using object-oriented program using Java and Python	CO2
5.	Develop Object Oriented Program to demonstrate inheritance and overriding super class methods in Java and Python	CO3
6.	Develop Object Oriented Program to demonstrate abstract base classes abstract methods in Java and Python	CO3
7.	Construct Object Oriented Program to demonstrate exception handling in Java and Python	CO4
8.	Construct Object Oriented Program in Java and Python to demonstrate File handling and Object Serialization	CO5
9.	Develop Object Oriented Program for manipulation of data using Collections in Java and Python	CO5
10.	Develop event-driven programs using Java and Python	CO6
11.	Develop concurrent programs using Java and Python	CO6

Learning Resources

1. Herbert Schildt: “Java: The Complete Reference”, Twelfth Edition, McGraw-Hill, 2021.
2. Tony Gaddis, “Starting Out with Java: From Control Structures through Objects”, Sixth edition, Pearson Education Limited, 2016.
3. Bart Baesens, Aimee Backiel, SeppevandenBroucke, “Beginning

Java Programming:The Object-Oriented Approach", John Wiley & Sons, 2015.
4. Allen B. Downey, "Think Python: How to Think like a Computer Scientist", 2nd Edition, O'Reilly Publishers, 2016. Grady Booch, Robert Maksimchuk, Michael Engel, Bobbi Young, Jim Conallen, Kelli Houston" Object Oriented Analysis and Design with Applications", Third Edition, 2012.

Course Designers:

1. Dr M.Vijayalakshmi, Professor, CSE – mviji@tce.edu