University of Engineering & Management, Kolkata
University of Engineering & Management, Jaipur
Institute of Engineering & Management, Kolkata

Department of Computer Science

B.Tech in Computer Science and Engineering

COURSE STRUCTURE

Batch: 2022-2026

Semester VII (Fourth Year) Curriculum								
Sl. No	**Type of course**	**Course Code**	**Course Name**	**Hours per week**				**Credit Points**
				Lecture	**Tutorial**	**Practical**	**Sessional**	
Theory Papers								
1	Professional Core Course	PCCCS701	Compiler Design	3	0	0	0	3
2	Professional Elective Course	PECCS701	Elective-III	3	0	0	0	3
3	Open Elective Course	OECCS701	Open Elective-I	3	0	0	0	3

4	Humanities & Social Sciences including Management course	ESP(CS)701	Essential Studies for Professionals – VII (CS)	2	0	0	0	0.5
	Total			**11**	**0**	**0**	**0**	**9.5**
Practical Papers								
1	Professional Core Course	PCCCS791	Compiler Design Laboratory	0	0	4	0	2
	Total			**0**	**0**	**4**	**0**	**2**
Sessional Papers								
1	Humanities & Social Sciences including Management course	SDP781	Skill Development for Professionals - VII	0	0	0	2	0.5
2	Innovative Project	PRJCS781	Project – II	0	0	12	0	6
3	Internship	SICS781	Summer Internship - I	0	0	0	0	4
	Total			**0**	**0**	**12**	**2**	**10.5**
Mandatory Requirements								
Sl. No	**Type of course**	**Course Code**	**Course Name**	**Hours per week**				**Score/Credit/Count**

1	Co-curricular & Extra Curricular Activities	MAR	Mandatory Additional Requirements (Score)	-	-	-	-	-
2	Honours	MOOCs	Massive Open Online Course (Credit)	-	-	-	-	-
3	Certification	IFC	Industry and Foreign Certification (Count)	-	-	-	-	-
Total				**11**	**0**	**16**	**2**	**22**

Offered Elective List

Category	Course Name	Course Code
Professional Elective - III	Ad-hoc and Sensor Network	PECCS701A
	Warehousing and Business Intelligence	PECCS701B
	Computer Vision	PECCS701D
Open Elective - I	Enterprise System	OECCS701A
	Economic Policies in India	OECCS701B

University of Engineering & Management, Kolkata
University of Engineering & Management, Jaipur
Institute of Engineering & Management, Kolkata

Department of Computer Science

B.Tech in Computer Science and Engineering

DETAILED SYLLABUS

Course Code- PCCCS701
Course Title – Compiler Design
Credit – 3
Category – Professional Core Course
Semester – VII
L:T:P:S – 3:0:2:0
Pre-requisite – Basic knowledge of Data Structures, Algorithms, Formal Language and Automata Theory

Course Outcomes:

CO1	**Students will be able to learn the grammar specification for developing the lexical analyzer.**
CO2	**Design a given parser specification design top-down and bottom-up parsers.**
CO3	**Develop syntax directed translation schemes.**
CO4	**Develop algorithms to generate code for a target machine.**

Study Material	**Coursera**	**NPTEL**	**LinkedIn Learning**	**Infosys Springboard**
Lesson Plan				

Module No.	Topic	Sub-topics	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignments	Textbook Mapping
1	**Overview of compilation & Lexical Analysis**	The structure of a compiler and applications of compiler technology; Lexical analysis - The role of a lexical analyzer, specification of tokens, recognition of tokens, hand-written lexical analyzers, LEX, examples of LEX programs. Introduction to syntax analysis -Role of a parser, use of context-free grammars (CFG) in the specification of the syntax of	***International Academia:*** ***https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/pages/c11/*** (MIT Open Courseware): ***AICTE-prescribed syllabus:***	8	Write a program that takes a simple code snippet (e.g., a few lines of C or a pseudo-language) as input and identifies all the tokens present, classifying them into categories like keywords, identifiers, operators,	**Textbook-1** Chapters: 1 to 3 **Page No. – 1 to 158**

		programming languages, techniques for writing grammars for programming languages (removal left recursion, etc.), non-context-free constructs in programming languages, parse trees and ambiguity, examples of programming language grammars	***https://drive.google.com/file/d/11CBeslrHOERE-XHDJSNsIHlRONmt_tYX/view?usp=sharing*** ***Industry Mapping:*** ***https://www.coursera.org/programs/iem-faculty-learning-program-rtyr7/projects/googlecloud-form-parsing-with-document-ai-python-zdlse***		literals (integer, float, string), and punctuation. The output should be a list of (token, token_type) pairs. Implement an algorithm that takes a given regular expression (e.g., (a|b)*c) and constructs its equivalent Non-deterministic Finite Automaton (NFA). You can represent the NFA using adjacency lists or a similar data structure. Write a program that takes an NFA (represented as in the previous question) as	

					input and converts it into an equivalent Deterministic Finite Automaton (DFA) using the subset construction algorithm. Design a simple programming language with a few keywords, operators, and data types. Then, implement a handwritten lexical analyzer (without using tools like LEX) in a language like Python or Java that can tokenize programs written in this small language. Write a LEX program (or	

					Flex, its open-source counterpart) that can identify and classify the following tokens in an input file: keywords (if, else, while, int), identifiers (starting with a letter, followed by letters or digits), integer literals, and basic arithmetic operators (+, -, *, /). Extend the LEX program from the previous question to ignore C-style (/* ... */) and C++-style (// ...) comments, as well as whitespace (spaces, tabs, newlines). The output should	

					only contain the identified tokens. Given a context-free grammar for a simple arithmetic expression language (e.g., with addition, subtraction, multiplication, division, and parentheses), write a program that can: Determine if the grammar is ambiguous. (Hint: You might try to derive the same string in two different ways). Draw the parse tree for a given valid expression	

					according to the grammar. Write a program that takes a context-free grammar as input and removes any immediate left recursion present in its production rules. The output should be the modified grammar without left recursion. Choose either a top-down parsing technique (like recursive descent) or a bottom-up technique (like shift-reduce) and implement a parser for a small subset of a programming	

					language based on a given context-free grammar. The parser should take a sequence of tokens as input and output whether the input is syntactically valid according to the grammar. For a top-down parser, you might need to handle LL(1) conditions. Research and prepare a report or a short presentation explaining some common non-context-free constructs found in programming languages (e.g., declaration before use, type checking	

					consistency). Provide examples of how these constructs cannot be easily handled by standard context-free grammars.	
2	**Top-down parsing**	FIRST & FOLLOW sets, LL(1) conditions, predictive parsing, recursive descent parsing, error recovery. LR-parsing - Handle pruning, shift-reduce parsing, viable prefixes, valid items, LR(0) automaton, LR-parsing algorithm, SLR(1), LR(1), and LALR(1) parsing. YACC, error recovery with YACC and examples of YACC specifications. Syntax-directed definitions (attribute grammars)-Synthesized and inherited attributes, examples of SDDs, evaluation orders for attributes of an SDD, dependency graphs. S-attributed and L- attributed SDDs and their implementation using LR-parsers and recursive	***International Academia:*** ***https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/pages/c11/*** (MIT Open Courseware): ***AICTE-prescribed syllabus:*** ***https://drive.google.com/file/d/11CBeslrHOERE-XHDJSNsIHlRONmt_tYX/view?usp=sharing*** ***Industry Mapping:***	10	Write a program that takes a context-free grammar (CFG) as input and computes the FIRST and FOLLOW sets for all non-terminals in the grammar. The output should clearly display the grammar and the calculated FIRST and FOLLOW sets. Test your program with various grammars, including those with epsilon productions and	**Textbook-1** Chapters: 4, 5 **Page No.:** 159 - 341

		descent parsers respectively.	***https://www.coursera.org/programs/iem-faculty-learning-program-rtyr7/projects/googlecloud-form-parsing-with-document-ai-python-zdlse***		left recursion. Develop a program that takes a CFG and the computed FIRST and FOLLOW sets as input. The program should then check if the given grammar satisfies the LL(1) conditions. The output should indicate whether the grammar is LL(1) or not, and if not, identify the violating productions. Implement an algorithm that takes an LL(1) grammar (or one that can be made LL(1) by left-factoring) and constructs its	

					predictive parsing table. The output should be a clear representation of the parsing table. Write a program that implements a non-recursive predictive parser using the parsing table generated in the previous question. The parser should take a string as input and determine if it is accepted by the grammar. The output should show the sequence of moves made by the parser (stack, input, action). Include basic error detection for	

					invalid input strings. Design a simple context-free grammar (e.g., for arithmetic expressions with basic operators). Implement a recursive descent parser for this grammar in your chosen programming language. The parser should take an input string and output whether it is a valid sentence of the grammar. Write a program that simulates the actions of a shift-reduce parser for a given grammar and input string. The program should output	

					the sequence of stack contents, input remaining, and the action (shift or reduce) performed at each step. Include the detection of acceptance and error states. Implement an algorithm that takes a context-free grammar and constructs its LR(0) automaton (the collection of LR(0) items and the transitions between them). The output should clearly represent the states and transitions of the automaton. Extend the LR(0)	

					automaton construction to generate the SLR(1) parsing table. Your program should take a grammar and the LR(0) automaton as input and produce the SLR(1) parsing table (action and goto parts). Handle potential conflicts and indicate if the grammar is SLR(1). Write a YACC (or Bison) specification for a simple programming language construct (e.g., assignment statements, simple arithmetic expressions, or a	

					basic conditional statement). Compile and test your YACC specification with valid and invalid input. Observe how YACC handles the parsing and reports syntax errors. Consider a simple syntax-directed definition (SDD) for calculating the value of arithmetic expressions. Implement a program that takes a parse tree (you can manually construct simple parse trees for testing) and evaluates the	

					attributes according to the SDD rules. Demonstrate the evaluation process for both synthesized and inherited attributes (if applicable in your chosen SDD).	
3	**Semantic analysis**	Symbol tables and their data structures. Representation of "scope". Semantic analysis of expressions, assignment, and control-flow statements, declarations of variables and functions, function calls, etc., using S- and L-attributed SDDs (treatment of arrays and structures included). Semantic error recovery Intermediate code generation - Different intermediate representations –quadruples, triples, trees, flow graphs, SSA forms, and their uses. Translation of expressions (including array references with subscripts) and	***International Academia:*** ***https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/pages/c11/*** (MIT Open Courseware): ***AICTE-prescribed syllabus:*** ***https://drive.google.com/file/d/11CBeslrHOERE-XHDJSNsIHlRON***	10	Create functions to insert, lookup, and delete entries in the symbol table. Each entry should store at least the variable name and its data type. Demonstrate the insertion and retrieval of symbol information for a small set of declared variables. Modify the	**Textbook-1** Chapters: 5, 6 **Page No.:** 279 to 388

		assignment statements. Translation of control-flow statements – it-then- else, while-do, and switch. Short-circuit code and control-flow translation of Boolean expressions. Back patching. Examples to illustrate intermediate code generation for all constructs	***mt_tYX/view?usp=sharing*** ***Industry Mapping:*** ***https://www.coursera.org/programs/iem-faculty-learning-program-rtyr7/projects/googlecloud-form-parsing-with-document-ai-python-zdlse***		symbol table implementation to support the concept of scope. Implement a mechanism to enter and exit scopes. Demonstrate how variables declared in different scopes are managed and how name resolution works (e.g., accessing a variable in an outer scope). Write a program that takes simple expressions (e.g., a + b, x < y) as input. Perform type checking to ensure that the operands are compatible. Report any type errors (e.g.,	

					adding an integer to a boolean). Assume variables have been previously declared and their types are available in the symbol table. Write a program that takes assignment statements (e.g., x = y + 5) as input. Perform type compatibility checks between the left-hand side variable and the right-hand side expression. Handle implicit type conversions (if applicable in your chosen scope) and report type errors.	

					Write a program that takes if-then-else statements as input. Ensure that the condition expression is of a boolean type. Demonstrate the semantic analysis process. Write a program that processes variable and function declarations. For variables, store their names and types in the symbol table. For functions, store their names, return types, and the types and number of their parameters. Detect and report redeclaration	

					errors. Write a program that takes function calls as input. Check if the function being called has been declared, if the number of arguments matches the function definition, and if the types of the actual arguments are compatible with the formal parameters. Report any errors. Write a program that takes arithmetic expressions as input and generates their corresponding quadruple representation.	

					Each quadruple should have the format (operator, operand1, operand2, result). Handle operator precedence and associativity appropriately. Write a program that takes assignment statements and while-do loops as input and generates their corresponding triple representation. Each triple should have the format (operator, operand1, operand2). For control flow, you'll need to represent the conditions and	

					jump targets using triple indices. Take a simple sequence of assignments and arithmetic operations with variable reassignments. Convert this code into SSA form by introducing new versions of variables where necessary. Demonstrate the key property of SSA: each variable is assigned a value only once.	
4	**Run-time environments**	Stack allocation of space and activation records. Access to non-local data on the stack in the case of procedures with and without nesting of procedures. Introduction to machine code generation and optimization- Simple	***International Academia:*** ***https://ocw.mit.edu/courses/6-004-computation-structures-spring-2017/pages/c11/***	8	Design the layout of an activation record for a function in a hypothetical programming language. Include space for parameters,	**Textbook-1** **Chapters:** 7, 8, 9, 10 **Page No.:** 389 to 751

		machine code generation, examples of machine-independent code optimizations.	(MIT Open Courseware): ***AICTE-prescribed syllabus:*** ***https://drive.google.com/file/d/11CBeslrHOERE-XHDJSNsIHlRONmt_tYX/view?usp=sharing*** ***Industry Mapping:*** ***https://www.coursera.org/programs/iem-faculty-learning-program-rtyr7/projects/googlecloud-form-parsing-with-document-ai-python-zdlse***		local variables, the return address, and the control link (dynamic link). Illustrate how this layout would be instantiated for a simple function call with a few parameters and local variables. Write a program (in any language) to simulate the stack during a sequence of simple function calls (without nesting). The program should track the stack pointer and the contents of each activation record pushed onto and popped from the stack. Demonstrate	

					this with at least three sequential function calls. Write a small program with a function that declares and uses local variables. Then, manually trace the execution, showing how the activation record for this function is created on the stack and how the local variables are accessed (relative to the frame pointer). Implement a function that takes parameters passed by value. Simulate the stack during a call to this function,	

					showing how the parameter values are copied into the activation record. Implement a function that takes parameters passed by reference (if your chosen simulation language allows). Simulate the stack during a call, illustrating how the addresses of the actual arguments are stored in the activation record. Extend the stack simulation from question 2 to include nested function calls.	

					Demonstrate how the control links in the activation records are used to maintain the dynamic chain and enable the correct return from nested calls. Include at least one level of nesting. Consider a program with nested procedures and static scoping. Implement a scenario where an inner procedure needs to access a variable declared in an enclosing procedure. Manually trace the stack and demonstrate how the static	

					link (access link) in the activation record of the inner procedure would be used to find the non-local variable in the activation record of the outer procedure. Choose a very simple arithmetic expression (e.g., a = b + c * 2). Design a sequence of hypothetical machine code instructions (e.g., using a simplified assembly-like language with instructions like LOAD, STORE, ADD, MUL) to evaluate this expression. Assume	

					variables a, b, and c are stored in memory locations. Take a code snippet with constant expressions (e.g., x = 5 + 3 * 2; y = z + 8). Apply the constant folding optimization manually to simplify the code. Show the original and optimized code snippets. Write a small code segment that includes a variable that is assigned a value but never subsequently used. Identify this dead code and demonstrate how it can be	

					eliminated to optimize the code. Show the original and optimized code snippets.	

Textbooks:

1. Compilers: Principles, Techniques, and Tools, by A.V. Aho, Monica Lam, Ravi Sethi, and J.D. Ullman, (2nd ed.), Addison-Wesley, 2007 (main text book, referred to as ALSU in lab assignments).

Reference books:

1. K.C. Louden, Compiler Construction: Principles and Practice, Cengage Learning,
2. D. Brown, J. Levine, and T. Mason, LEX and YACC, O?Reilly Media, 1992.

Online Resources: **https://drive.google.com/file/d/19i8gYQD2xDzf8nUCxdqp7czFLhaDrVFp/view?usp=sharing**

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3		3	3				3	3		3
CO2	3	2		3	3						2	2

CO3	3	2		3	3				3	2		3
CO4	3	2	1	2	2				2	2		2

3: Strong correlation

2: Medium correlation

1: Weak correlation

University of Engineering & Management, Kolkata
University of Engineering & Management, Jaipur
Institute of Engineering & Management, Kolkata

Department of Computer Science

B.Tech in Computer Science and Engineering

DETAILED SYLLABUS

Course Code- PECCS701A
Course Title – Ad-hoc and Sensor Networks
Credit – 3
Category – Professional Elective Course
Semester – VII
L:T:P:S – 3:0:0:0
Pre-requisite – Basic knowledge of Computer Networks & Operating System

Course Outcomes:

CO1	**Students will able to understand the principles, architecture and challenges of Ad hoc and Wireless Sensor Networks.**

CO2	**Students will able to analyze and evaluate the performance of various energy-efficient routing protocols such as AODV, DSDV, PAMAS, LEACH & MAC etc. used in these networks.**
CO3	**Students will able to understand the transport layer and all possible security concerns along with processes in Ad hoc and Sensor Networks.**
CO4	**Students will able to learn various encryption-decryption methods such as DES, AES, RSA, Diffie-Hellman etc. along with its applicability in real scenario. Also able to understand the security features and trust policies in designing an Operating System.**

Study Material	**Coursera**	**NPTEL**	**Linkedin Learning**	**Infosys Springboard 5G**

Module No.	Topic	Sub-topics	Mapping with Industry and International Academia	Lecture Hours	Textbook Mapping
1	**AD HOC NETWORKS – INTRODUCTION AND ROUTING PROTOCOLS**	Elements & issues of Ad hoc Networks, Commercial applications of Ad hoc Network, Ad hoc wireless Internet, Issues in Designing a Routing Protocol for Ad Hoc Networks, Classifications of Routing Protocols, Table Driven Routing Protocols – Destination Sequenced Distance Vector (DSDV), On–Demand Routing protocols –Ad hoc On–Demand Distance Vector Routing (AODV).	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/6-452-principles-of-wireless-communications-spring-2006/* ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICT*	6	**Textbook-2** Chapters: 1, 2 and 3. *5G Networks fundamentals* (*https://www.coursera.org/learn/5g-network-fundamentals*)

			E%20-%20UG%20CSE.pdf ***Industry Mapping:*** *Wireshark, Packet Tracer, OPNET, NS3*		
2	**WIRELESS SENSOR NETWORKS – INTRODUCTION & ARCHITECTURES**	Challenges for Wireless Sensor Networks, Enabling Technologies for Wireless Sensor Networks (WSNs), WSNs' applications, Single-Node Architecture – Hardware Components, Energy Consumption of Sensor Nodes, Network Architecture – Sensor Network Scenarios	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/mas-863-how-to-make-almost-anything-fall-2002/* ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%20-%20UG%20CSE.pdf* ***Industry Mapping:*** *Wireshark, Packet Tracer, OPNET, NS3*	6	**Textbook-2** Chapters: 1, 2 and 3.

3	**PROTOCOLS IN WSNs**	MAC Protocols for WSNs, Low Duty Cycle Protocols and Wakeup Concepts – S-MAC, The Mediation Device Protocol, Contention based protocols – PAMAS, Schedule based protocols – LEACH, IEEE 802.15.4 MAC protocol, Routing Protocols, Energy Efficient Routing, Challenges and Issues in Transport layer protocol.	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/16-36-communication-systems-engineering-spring-2009/* ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%20-%20UG%20CSE.pdf* ***Industry Mapping:*** *Wireshark, Packet Tracer, OPNET, NS3*	4	**Textbook-2** Chapters: 4 and 5.
4	**SENSOR NETWORK SECURITY**	Network Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Layer wise attacks in wireless sensor networks, possible solutions for jamming, tampering, black hole attack, flooding attack. Key Distribution and Management, Secure Routing – SPINS, reliability requirements in sensor networks.	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/6-857-network-and-computer-security-spring-2014/* ***AICTE-prescribed syllabus:***	6	**Textbook-1** Chapters: 25 to 29 **Textbook-2** Chapters: 611 to 762

			https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%20-%20UG%20CSE.pdf ***Industry Mapping:*** *Wireshark, Packet Tracer, OPNET, NS3*		
5	**SECURITY IN COMPUTING ENVIRONMENT**	Need for Security, Security Attack, Security Services, Information Security, Methods of Protection. Terminologies used in Cryptography, Substitution Techniques, and Transposition Techniques. Encryption and Decryption: Characteristics of Good Encryption Technique, Properties of Trustworthy Encryption Systems, Types of Encryption Systems, Confusion and Diffusion, Cryptanalysis.	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/6-857-network-and-computer-security-spring-2014/* ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%20-%20UG%20CSE.pdf* ***Industry Mapping:*** *Wireshark, Packet Tracer, OPNET, NS3*	8	**Textbook-1** Chapters: 3 to 4, 6, 9-11
6	**SYMMETRIC KEY ENCRYPTION**	Data Encryption Standard (DES), Double and Triple DES, Security of the DES, Advanced Encryption Standard (AES) Algorithm, DES and	***International Academia:*** (MIT Open Courseware):	8	**Textbook-1** Chapters: 3 to 4, 6,

		AES Comparison. **Public Key Encryption:** Characteristics of Public Key System, RSA Technique, Key Exchange, Diffie-Hellman Scheme, Cryptographic Hash Functions, Digital Signature, Certificates, Certificate Authorities. **Protection of Computing Resources:** Secure Programs, Non-malicious Program Errors, Viruses and Other Malicious Code, Targeted Malicious Code, Methods of Control.	*https://ocw.mit.edu/courses/6-875-cryptography-and-cryptanalysis-spring-2005/* ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%20-%20UG%20CSE.pdf* ***Industry Mapping:*** *Wireshark, Packet Tracer, OPNET, NS3*		9-11
7	**SECURITY FEATURES & POLICIES IN OPERATING SYSTEM**	Objects to be Protected, Protection Methods of Operating Systems, Memory Protection, File Protection, and User Authentication. Network, IP and Web Security	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/6-828-operating-system-engineering-fall-2012/* ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%20-%20UG%20CSE.pdf*	3	**Textbook-2** Chapters: 7.

			Industry Mapping: *Wireshark, Packet Tracer, OPNET, NS3*		

Textbooks:

1. William Stallings: Cryptography and Network Security, seventh edition ISBN 978-1-292-15858-7 or sixth edition ISBN 978-0-273-79335-9.

2. Protocols and Architectures for Wireless Sensor Networks, H. Karl and A. Willig, Wiley Publishers, 2005.

3. Russ Housley, Tim Polk, " Planning for PKI: Best Practices Guide for Deploying Public Key Infrastructure," Wiley, March 2001, 352 pages.

4. John E. Canavan, “The Fundamentals of Network Security," Artech House, February 2001, 350 pages.

Reference books:

1. Andrew Mason, Mark J. Newcomb, “Cisco Secure Internet Security Solutions," Cisco Press, May 2001, 528 pages.

2. Warwick Ford, Michael S. Baum, " Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption (2nd Edition)," Prentice Hall, December 2000, 620 pages.

University of Engineering & Management, Kolkata
University of Engineering & Management, Jaipur
Institute of Engineering & Management, Kolkata

Department of Computer Science

B.Tech in Computer Science and Engineering

DETAILED SYLLABUS

Course Code- PECCS701B
Course Title – Data Warehousing and Business Intelligence
Credit – 3
Category – Professional Elective Course
Semester – VII
L:T:P:S – 3:0:0:0
Pre-requisite – Basic knowledge of RDBMS

Course Outcomes:

CO1	**Students will be able to learn the general principles of Data Warehouse, Data Marts, Data Lake, and understand the importance of metadata, etc.**

CO2	**Students will be able to learn data warehouse architectures and models (Star, Snowflake and Fact Constellation), different OLAP techniques.**
CO3	**Students will be able to learn the concepts of data cube and implementation of data warehousing, best practices and issues.**
CO4	**Students will be able to know the concepts of business analysis, data visualisation techniques and get acquainted with the features of BI tools**

Study Material	**Coursera**	**NPTEL**	**Linkedin Learning**

Module No.	Topic	Sub-topics	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignments	Textbook Mapping
1	**Data Warehousing Concepts**	Definition, Characteristics of a Data Warehouse, Data Marts, Data Lake, Identifying Business Data Flow & Processes, Data sources, ETL Overview, Data Extraction process, Transformation process, Assuring Data Quality, Transportation process, Maintaining Warehouse Data, Concept of Metadata	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/15-062-data-mining-spring-2003/pages/syllabus/* (University of Texas at Dallas):	8	-	**Textbook-1** Chapter 1: The Chess Pieces (Basic Elements of Data Warehousing) **Textbook-2** Chapter 2: Data warehouse: the building Blocks

			https://dox.utdallas.edu/syl60773 ***Makaut, WB-prescribed syllabus:*** https://makautexam.net/aicte_details/Syllabus/CSE/sem6.pdf ***Industry Mapping:*** https://www.coursera.org/specializations/data-warehousing			**Reference 1:** Chapters: 1: Data Warehousing, Business Intelligence, and Dimensional Modelling Primer (https://www.coursera.org/specializations/data-warehousing)
2	**Data Warehouse Architecture and Models**	Warehouse architecture, Identifying Warehouse Data: Fact data, Dimension data, Hierarchies, Summaries (roll-ups), Data Warehouse models: Star, Snowflake and Fact Constellation, Modelling the Data Warehouse, User query requirements and User query progression, OLTP vs OLAP Access,	***International Academia:*** (MIT Open Courseware): https://ocw.mit.edu/courses/15-062-data-mining-spring-2003/pages/syllabus/	12		**Textbook-1** Chapter 5: A First Course on Dimensional Modelling Chapter 8: Introducing Data Warehouse Architecture Chapter 9:

		Relational OLAP (ROLAP) Access, Multidimensional OLAP (MOLAP), OLAP Query Techniques, Introduction of Data Mining concepts, Introduction to Big Data, Hadoop	(University of Texas at Dallas): *https://dox.utdallas.edu/syl60773* ***Makaut, WB-prescribed syllabus:*** *https://makautexam.net/aicte_details/Syllabus/CSE/sem6.pdf* ***Industry Mapping:*** *https://www.coursera.org/specializations/data-warehousing*			Back Room Technical Architecture **Textbook-2** Chapter 7: Architectural Components (*https://www.coursera.org/specializations/data-warehousing*)
3	**Data Cube Computation, Implementation and Maintenance of Data Warehouse**	Data cube concept, materialization of different data cubes, methods of data cube computation, multi-feature cubes Best practices of DW project Implementation and DW deployment process, and issues.	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/15-062-data-mining-spring-*	8		**Textbook-1** Chapter 19: Maintaining and Growing the Data Warehouse **Textbook-2** Chapter 12: Data

			2003/pages/syllabus/ (University of Texas at Dallas): *https://dox.utdallas.edu/syl60773* ***Makaut, WB-prescribed syllabus:*** *https://makautexam.net/aicte_details/Syllabus/CSE/sem6.pdf* ***Industry Mapping:*** *https://www.coursera.org/specializations/data-warehousing*			Extraction, Transformation, and Loading (*https://www.coursera.org/specializations/data-warehousing*)
4	**BI Implementation and Tools**	Key Drivers, Key Performance Indicators and operational metrics, BI Architecture/ Framework, Best Practices, Business Decision Making, Big Data and BI, Data Analytics – Objective Curve, principles of	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/15-062-data-mining-spring-*	8		**Reference 1:** Chapter 21: Big Data Analytics (*https://www.coursera.org/specializations/data-*

		Analytics and tools, Dash Boards, Scorecards, Getting Data: Connections, Extracts, Metadata, Joins, Blends, Filters, Common Visualizations; Bar Charts, Treemaps, Area Charts, Pie Charts, Circle Charts, Box and Whisker, Histograms, Scatterplots, Line Charts, Geographic Visualizations, Dashboards, Storyboards Overview of tools: Microsoft PowerBI, SAP Business Intelligence, Tableau, Oracle BI, SAS Business Intelligence	*2003/pages/syllabus/* (University of Southern California): *https://web-app.usc.edu/soc/syllabus/20201/16272.pdf* ***Makaut, WB-prescribed syllabus:*** *https://makautexam.net/aicte_details/Syllabus/CSE/sem6.pdf* ***Industry Mapping:*** *https://www.coursera.org/specializations/data-warehousing*			*warehousing*)

Textbooks:

1. Ralph Kimball (Author), Margy Ross (Author), Warren Thornthwaite (Author), Joy Mundy (Author), The Data Warehouse Lifecycle Toolkit 2nd Edition, Wiley.

2. Data Warehousing Fundamentals for IT Professionals, Second Edition by Paulraj Ponniah, Wiley India.

Reference books:

1. Ralph Kimball, Margy Ross, "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modelling", John Wiley & Sons
2. Sam Anahory, Dennis Murray, "Data Warehousing in the Real World", PEARSON

Online Resources:

CO-PO Mapping:

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

3: Strong correlation

2: Medium correlation

1: Weak correlation

University of Engineering and Management

Institute of Engineering & Management, Salt Lake Campus

Institute of Engineering & Management, New Town Campus

University of Engineering & Management, Jaipur

Department of Computer Science

B.Tech in Computer Science and Engineering

DETAILED SYLLABUS

Course Code - PECCS701D
Course Title - Computer Vision
Credit – 3
Category – Professional Elective
Semester – VII
L:T:P:S – 3:0:0:0
Prerequisites - Familiarity with Python programming, including NumPy, OpenCV basics, and linear algebra fundamentals.

Course Outcomes:

CO1	**Understand and apply basic image processing techniques such as filtering, geometric transformations, and histogram operations.**
CO2	**Implement and evaluate feature detection and matching techniques for object recognition and image registration.**
CO3	**Apply segmentation algorithms and contour analysis methods for object extraction and shape analysis in images.**
CO4	**Design and train convolutional neural networks (CNNs) for image classification tasks and apply object detection models such as YOLO/SSD.**
CO5	**Demonstrate practical skills in using popular computer vision libraries (OpenCV, Keras, TensorFlow) to develop end-to-end vision pipelines.**

Study Material	**Coursera**	**NPTEL**	**Linkedin Learning**	**Infosys Springboard 5G**

Module No	Topic	Sub Topic	Mapping with Industry and International Academia	Contact Hours	Assignment Question Mapped with the Module	Reference Textbooks

1	**Image Processing Fundamentals**	Introduction to Computer Vision Image Representation: Grayscale, RGB, Binary Basic Image Processing Operations: Filtering, Blurring, Sharpening Geometric Transformations: Scaling, Rotation, Translation Image Histograms and Histogram Equalization	***NPTEL:*** Fundamentals of Digital Image Processing: https://onlinecourses.nptel.ac.in/noc22_ee116/preview?utm_source=chatgpt.com ***International Academia:*** ***(MIT OCW)*** Computational Photography https://www.youtube.com/playlist?list=PLUl4u3cNGP61pwA6paIRZ30q1sjLE8b6c ***Industry Mapping:*** Image Processing, Robotic Vision, AR/VR	9	1.Implement a Python program to apply Gaussian blur, median filter, and sharpening on an image. Compare the results. 2. What is the difference between image blurring and sharpening? Explain with suitable filters.	**Text Book:** Chapter 1 Page 1-44

2:	**Feature Detection and Matching**	Edge Detection: Sobel, Canny Corner Detection: Harris Corner Detector Keypoint Detection and Description: SIFT, SURF, ORB Feature Matching: Brute-Force and FLANN based matcher	***International Academia:*** (MIT OpenCV tutorials) https://fab.cba.mit.edu/classes/865.15/people/desmond.lim/project-06.html ***Industry Mapping***: Autonomous Vehicle, AR/VR, Robotics	9	1. Apply Canny edge detection on an image and visualize the edges. 2. Detects corners in an image using the Harris Corner Detector.	**Text Book 3:** Chapter 4
3	**Image Segmentation and**	Image Thresholding:	***NPTEL***	9	1.	**Text Book 1:**

	Contour Analysis	Global, Adaptive Contour Detection and Analysis Morphological Operations Image Segmentation: Watershed, GrabCut	(Computer Vision) https://onlinecourses.nptel.ac.in/noc21_cs101/preview?utm_source=chatgpt.com ***Industry Mapping:*** Used in biomedical imaging, satellite image analysis, surveillance		Perform adaptive thresholding on an image with uneven illumination. 2. Detect contours in an image and calculate area, perimeter, and bounding box for each contour.	Chapter 10
4	**Deep Learning for Computer Vision**	Introduction to Convolutional Neural Networks (CNNs) CNN Architectures: LeNet, AlexNet, VGG	***NPTEL*** (Deep Learning)	9	1. Build and train a simple CNN for CIFAR-10 or MNIST image classification. Q2. Apply transfer learning using a pretrained VGG16	**Text Book 2:** Chapter 1 - 4

		Image Classification using CNNs Object Detection: YOLO, SSD basics Transfer Learning for Vision Tasks	https://onlinecourses.nptel.ac.in/noc21_cs101/preview?utm_source=chatgpt.com ***Industry Mapping:*** Essential for ML Engineer, AI Researcher, and Data Scientist roles		model to classify custom image dataset (flowers, animals, etc.).	

Textbooks:

1. "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods, Pearson
2. Deep Learning for Computer Vision" by Rajalingappaa Shanmugamani
3. Computer Vision: Algorithms and Applications" by Richard Szeliski

Reference books:

1. "Learning OpenCV 4" by Gary Bradski and Adrian

2. Programming Computer Vision with Python" by Jan Erik Solem

CO-PO Mapping:

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

3: Strong correlation

2: Medium correlation

1: Weak correlation

University of Engineering & Management, Kolkata
University of Engineering & Management, Jaipur
Institute of Engineering & Management, Kolkata

Department of Computer Science

B.Tech in Computer Science and Engineering

DETAILED SYLLABUS

Course Code- OECCS701A
Course Title – Enterprise Systems
Category – Open Elective Course
Credit – 3
Semester – VII
L:T:P:S – 3:0:0:0
Pre-requisite – Basic knowledge of Business, Finance & Accounting, Awareness on Business Environment

Course Outcomes:

CO1	**Understand Enterprise systems models.**

CO2	**Understand Enterprise Software architecture.**
CO3	**Design and implement ERP models.**
CO4	**Implement interactive networks and applications.**
CO5	**Develop models for ERP for large projects.**

Module No.	Topic	Sub-topics	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignments	Textbook Mapping
1	**Introduction to Enterprise systems concepts**	Features, capabilities and Overview of Commercial Software, re-engineering work processes for IT applications, Business Process Redesign, Knowledge engineering and data warehouse. Business Modules: Finance, Manufacturing (Production), Human Resources, Plant Maintenance, Materials Management, Quality Management, Sales & Distribution	***International Academia:*** (Lecture Notes | Logistics and Supply Chain Management | Engineering Systems Division | MIT OpenCourseWare) ***AICTE-prescribed syllabus:*** (Final BTECH LSCM.pdf) ***Industry Mapping:***	6	1. Familiarization with ERPNext - open-source alternative to SAP. 2. Familiarization with Odoo ERP software. 3. Familiarization with anyLogistix software.	**Textbook-1** Chapters: 1, 2 and 3 **Textbook-2** Chapters: 1-4, 11 and part V.

			ERPNext, anyLogistix, Odoo (GitHub - frappe/erpnext: Free and Open Source Enterprise Resource Planning (ERP),Case study – anyLogistix, Open Source ERP and CRM | Odoo)			
2	**Enterprise Resource Planning (ERP)Architecture and Technologies**	Service Oriented Architecture (SOA): Principles of loose coupling, encapsulation, Interoperability. Decision Support System: On-Line Analytical Processing, Electronic Data Exchange, Customer Relationship Management (CRM), Supplier Relationship Management (SRM),	***International Academia:*** (Lecture Notes | Logistics and Supply Chain Management | Engineering Systems Division | MIT OpenCourseWare) ***AICTE-prescribed syllabus:*** (Final BTECH LSCM.pdf) ***Industry Mapping:***	8	1. Create a model of customer relationship management and business intelligence systems for catalogue and online retailers. 2. Create a model of Supplier Relationship Management for Healthcare system.	**Textbook-1** Chapter: 3 **Textbook-2** Chapter: part II.

			ERPNext, anyLogistix, Odoo (GitHub - frappe/erpnext: Free and Open Source Enterprise Resource Planning (ERP),Case study – anyLogistix, Open Source ERP and CRM | Odoo)			
3	**ERP Software Architecture and Technologies**	Introduction to MVC, MVC method of software development in a 3-tier environment, Microsoft .NET framework, PHP, Ruby on Rails, JavaScript, Ajax and Overview of SAP and Oracle Applications		10	1. Create ASP.NET MVC Application using Visual Studio. 2. Create JSP application using MVC framework	**Text Book-3** Chapters:1 - 6. **Coursera course: ASP.NET Core MVC [.NET 8] - The Complete Guide Specialization** ***https://www.coursera.org/specializations/packt-asp-net-core-mvc-net-***

						8-the-complete-guide
4	**ERP Network Architecture**	Introduction to MPLS, Virtual Private Networks (VPN), Storage area networks, Storage units, Back-up strategies, Local Area Network (LAN) technologies and products, Data Centres. Firewalls, Network monitoring and enforcement of policies, ERP Security Issues, Authentication, Authorisation, Access control, Roles, single-sign on, Directory servers, Audit trails, Digital signatures, Encryption, review of IPSec, SSL.	***International Academia:*** *(1.264J Lecture 37 Notes: Networks: Enterprise, VPN, MPLS. Course summary, Lecture Notes | Logistics and Supply Chain Management | Engineering Systems Division | MIT OpenCourseWare)* ***AICTE-prescribed syllabus:*** *(Final BTECH LSCM.pdf)* ***Industry Mapping: ERPNext, anyLogistix, Odoo (GitHub - frappe/er***	10	1. Configure and test a VPN connection. 2. Configure Firewall in Windows/Linux OS. 3. Configure an MPLS VPN in Cisco Packet Tracer.	**Text Book-4** Chapters: 1-12. **Enterprise Network Infrastructure : Coursera course** ***https://www.coursera.org/learn/packt-enterprise-network-infrastructure-pbclr?msockid=3e887050530967811ce4656152bb66d3***

			pnext: Free and Open Source Enterprise Resource Planning (ERP), *Case study – anyLogistix*, *Open Source ERP and CRM | Odoo*)			
5	**Trends in ERP**	Enterprise Application Integration (EAI), ERP and E-Commerce, ERP and Internet, Future Directions in ERP	***International Academia:*** (Lecture Notes | Logistics and Supply Chain Management | Engineering Systems Division | MIT OpenCourseWare) ***AICTE-prescribed syllabus:*** (Final	6	Design with the help of CASE tools to aid ERP Software acquisition process - Case study	**Text Book-1** Chapter 11

			BTECH LSCM.pdf) ***Industry Mapping: ERPNext, anyLogistix, Odoo (GitHub - frappe/erpnext: Free and Open Source Enterprise Resource Planning (ERP)***,Case study – anyLogistix, Open Source ERP and CRM | Odoo)			

Textbooks:

1. Alexis Leon, Enterprise Resource Planning, 2020,4th Edition, Tata McGraw Hill.
2. Alexis Leon, ERP Demystified, McGrawhill education.
3. Simone Chiaretta, Keyvan Nayyeri, Beginning ASP.NET MVC 1.0, Wrox Press.
4. A Practical Introduction to Enterprise Network and Security Management, Bongsik Shin, CRC Press Taylor & Francis Group.

Reference books:

1. The Age of Metapreneurship: A Journey into the Future of Entrepreneurship Design of Enterprise Systems: Theory, Architecture, and Methods, Ronald Giachetti.
2. Entrepreneurship: The Practice and Mindset Paperback, by Heidi M. Neck (Author), Christopher P. Neck (Author), Emma L. Murray.

3. ENTREPRENEURSHIP: The Art, Science, and Process for Success, Charles E. Bamford Associate Professor of Strategy & Entrepreneurship (Author), Garry D.Bruton Dr.

Online Resources:

Coursera course:

1. Enterprise Systems, Jason Chan, University of Minnesota.

LinkedIn course:

1. Enterprise Architecture Foundations, Dave Swersky.

CO-PO Mapping:

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

3: Strong correlation

2: Medium correlation

1: Weak correlation

University of Engineering & Management, Kolkata
University of Engineering & Management, Jaipur
Institute of Engineering & Management, Kolkata

Department of Computer Science

B.Tech in Computer Science and Engineering

DETAILED SYLLABUS

Course Code- OECCS701B
Course Title – Economic Policies in India
Credit – 3
Category – Open Elective
Semester – VII
L:T:P:S – 3:0:0:0
Prerequisite – Principles of Management

Course Outcomes:

CO1	**Understand the historical evolution of India's economic policies.**

CO2	**Analyze the impact of liberalization, privatization, and globalization on the Indian economy.**
CO3	**Evaluate sectoral economic policies and their influence on technology-driven industries.**
CO4	**Examine contemporary economic challenges and their technological implications.**

Study Material	**Coursera**	**NPTEL**	**Linkedin Learning**	**Infosys Springboard 5G**

Module No.	Topic	Sub-topics	Mapping with Industry and International Academia	Lecture Hours	Corresponding Lab Assignments	Textbook Mapping
1	**Historical Overview and Evolution of Economic Policies**	Economic policies in pre-independence India ● The mixed economy model post-independence ● The Five-Year Plans and their impact ● Industrial Policy Resolutions (1948, 1956, 1991) ● The Green Revolution and agricultural reforms	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/14-771-development-economics-fall-2021/* ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%2*	8	1. Design a policy brief proposing an alternative industrial strategy for post-independence India, considering the limitations of the 1956 Industrial Policy Resolution. 2. Formulate a hypothetical Five-Year Plan for India	**Textbook-1** Chapters: 1 to 8 *https://www.coursera.org/learn/art-and-science-of-economic-policy*

			0-%20UG%20CSE.pdf ***Industry Mapping:*** Public Sector Enterprises, Import Substitution Industrialization		focusing on inclusive agricultural growth, drawing lessons from the Green Revolution and earlier planning models.	
2	**Liberalization, Privatization, and Globalization (LPG) Era**	• Economic crisis of 1991 and the need for reforms • Structural reforms under the LPG policy • Impact of liberalization on industrial growth • Privatization: Strategies and challenges • Globalization and its effect on trade and investment	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/14-771-development-economics-fall-2021/* ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%20-%20UG%20CSE.pdf*	8	3. Design a policy roadmap for a developing economy facing a crisis similar to India's 1991 balance of payments crisis. 4. Formulate a new privatization model for India that balances efficiency, public accountability, and social welfare. 5. Construct a model industrial growth plan that leverages	**Textbook-2** Chapter: 6 *https://www.coursera.org/learn/art-and-science-of-economic-policy*

			Industry Mapping: Foreign Direct Investment (FDI)		liberalization but addresses its sectoral disparities. 6. Develop a strategic globalization plan for India that enhances trade competitiveness while protecting domestic industries.	

3	**Sectoral Economic Policies and Their Impact**	Agriculture and rural development policies ● Industrial policies and MSME sector promotion ● Financial sector reforms: Banking, insurance, and NBFCs ● Energy, infrastructure, and digital economy initiatives ● Labor market policies and social security reforms	***International Academia:*** (MIT Open Courseware): https://ocw.mit.edu/courses/14-771-development-economics-fall-2021/ ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%20-%20UG%20CSE.pdf* ***Industry Mapping:*** Financial Inclusion through NBFCs and Microfinance Institutions (MFIs)	14	7. Design an integrated rural development policy that combines agricultural reform, digital innovation, and rural employment generation. 8. Formulate a strategic plan to boost the MSME sector through policy innovation, access to credit, and digital transformation. 9. Construct a model framework for inclusive financial sector reform covering banks, insurance, and NBFCs for underserved populations. 10. Develop a sustainable infrastructure policy integrating renewable energy, smart cities, and digital connectivity.	**Textbook-1** **Textbook-2** Chapter 9 https://www.coursera.org/learn/art-and-science-of-economic-policy

					11. Create a labor policy blueprint that balances labor rights, gig economy flexibility, and universal social security.	
4	**Contemporary Economic Policies and Challenges**	● Fiscal and monetary policies: Role of RBI and government ● Digital India and fintech revolution ● Start-Up India and Make in India initiatives ● Environmental and	***International Academia:*** (MIT Open Courseware): *https://ocw.mit.edu/courses/14-771-development-*	6	12. Design a post-pandemic economic recovery plan focusing on employment generation, MSME revival, and inflation control. 13. Construct a	**Textbook-1** Chapters:21,22, 23 **Textbook-2** Chapter:2, 9,19 https://www.coursera.org/le

		sustainable economic policies ● Challenges: Unemployment, income inequality, inflation, and regional disparities ● Post-pandemic economic recovery strategies	*economics-fall-2021/* ***AICTE-prescribed syllabus:*** *https://www.aicte-india.org/sites/default/files/Model_Curriculum/AICTE%20-%20UG%20CSE.pdf* ***Industry Mapping:*** FinTech and Digital Payment Ecosystems (e.g., UPI, Aadhaar-linked banking), Green Economy and ESG (Environmental, Social, and Governance) Frameworks.		sustainable development policy that aligns India's environmental goals with industrial growth targets. 14. Formulate a national fintech roadmap that accelerates financial inclusion and supports Digital India. 15. Design a comprehensive policy proposal to boost entrepreneurship in rural India under Start-Up India and Make in India missions.	arn/art-and-science-of-economic-policy

Textbooks:

1. "Indian Economy: Performance and Policies" by Uma Kapila

2. "Indian Economy" by Ramesh Singh

Reference book:

1. "The Indian Economy: Problems and Prospects" edited by Bimal Jalan

Online Resources: https://www.youtube.com/watch?v=haIUbZ4eOAs&list=PLXLsXwBXH6Bmjp8vfb7yuf5RF7jXceNQe

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	0	0	0	3	3	3	3	2	1	3
CO2	0	0	0	0	0	3	3	3	3	1	1	3
CO3	0	0	0	0	0	3	3	3	3	2	2	2
CO4	0	0	0	0	0	3	2	3	2	2	3	3

3: Strong correlation

2: Medium correlation

1: Weak correlation

University of Engineering & Management, Kolkata

University of Engineering & Management, Jaipur

Institute of Engineering & Management, Kolkata

Department of Computer Science

B.Tech in Computer Science and Engineering

DETAILED SYLLABUS

Course Code- ESP(CS)701
Course Title – Essential Studies for Professionals (CS) - VII
Credit – 0.5
Category – Humanities & Social Sciences including Management course
Semester – VII
L:T:P:S – 2:0:0:0
Pre-requisite – Basic knowledge of Data Structures, Basics of Compiler, Operating systems and Computer network

Course Outcomes:

CO1	**To develop a detailed knowledge of compiler designs.**
CO2	**To learn all types of Data Base Management Systems' fundamentals.**
CO3	**To understand Operating Systems and its applications.**
CO4	**To use fundamentals of Computer networks and its methods.**

Module No.	Topic	Sub-topics	Mapping with Industry and International Academia	Lecture Hours	Textbook Mapping
1	**Compiler Design**	Lexical analysis, parsing, syntax-directed translation. Runtime environments. Intermediate code generation. Local optimization, Data flow analyses: constant propagation, likeness analysis, common sub expression elimination	***International Academia:*** (MIT Open Courseware): ***AICTE-prescribed syllabus:*** ***Industry Mapping:***	6	1. G.K publishers GATE Computer Science Engineering, 2. McGraw hill GATE 2020 Computer Science Engineering, 3. Wiley GATE 2020 Computer Science Engineering
2	**Databases**	Integrity constraints, normal forms. File organization, indexing (e.g., B and B+ trees). Transactions and concurrency control	***International Academia:*** (MIT Open Courseware): ***AICTE-prescribed syllabus:*** ***Industry Mapping:***	12	1. G.K publishers GATE Computer Science Engineering, 2. McGraw hill GATE 2020 Computer Science Engineering, 3. Wiley GATE 2020 Computer Science Engineering
3	**Operating System**	Processes, threads, inter-process communication, concurrency and synchronization, Deadlock, CPU scheduling. Memory management and virtual memory, File system. Flow and error control techniques, switching. IPv4/IPv6, routers and routing algorithms (distance vector, link state). TCP/UDP and sockets, congestion control.	***International Academia:*** (MIT Open Courseware): ***AICTE-prescribed syllabus:***	12	1. G.K publishers GATE Computer Science Engineering, 2. McGraw hill GATE 2020 Computer Science Engineering, 3. Wiley GATE 2020 Computer Science Engineering

		Application layer protocols (DNS, SMTP, POP, FTP, HTTP). Basics of Wi-Fi.	***Industry Mapping:***		
4	**Computer Networks**	Concept of layering: OSI and TCP/IP Protocol Stacks; Basics of packet, circuit and virtual circuit- switching; Data link layer: framing, error detection, Medium Access Control, Ethernet bridging; Routing protocols: shortest path, flooding, distance vector and link state routing; Fragmentation and IP addressing, IPv4, CIDR notation, Basics of IP support protocols (ARP, DHCP, ICMP), Network Address Translation (NAT); Transport layer: flow control and congestion control, UDP, TCP, sockets; Application layer protocols: DNS, SMTP, HTTP, FTP, Email	***International Academia:*** (MIT Open Courseware): ***AICTE-prescribed syllabus:*** ***Industry Mapping:***	6	1. G.K publishers GATE Computer Science Engineering, 2. McGraw hill GATE 2020 Computer Science Engineering, 3. Wiley GATE 2020 Computer Science Engineering

Text books:

1. G.K publishers GATE Computer Science Engineering,
2. McGraw hill GATE 2020 Computer Science Engineering,
3. Wiley GATE 2020 Computer Science Engineering

Online Resources:

Module	Platform	Course Link
Compiler Design	NPTEL	Link
Compiler Design	Stanford	https://youtu.be/sm0QQO-WZlM
Databases	NPTEL	Link
Databases	Coursera (IBM)	Link
Operating Systems	NPTEL	Link
Operating Systems	Coursera (Google)	Link
Computer Networks	NPTEL	Link
Computer Networks	Coursera (Stanford)	Link

CO-PO Mapping:

CO \ PO	PO1 (Engineering Knowledge)	PO2 (Problem Analysis)	PO3 (Design/Dev of Solutions)	PO4 (Investigation)	PO5 (Modern Tool Usage)	PO6 (Engineer & Society)	PO7 (Environment & Sustainability)	PO8 (Ethics)	PO9 (Individual & Team Work)	PO10 (Communication)	PO11 (Project Management)	PO12 (Life-long Learning)
CO1	2	1	1	1	1	-	-	-	-	-	-	3
CO2	2	1	1	1	1	-	-	-	1	1	1	3
CO3	2	1	1	1	1	1	-	-	1	1	1	3
CO4	2	1	1	1	1	1	-	-	1	1	1	3

3: Strong correlation

2: Medium correlation

1: Weak correlation