UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

NEW COURSE STRUCTURE– To be effective for Diploma 2023-24
[2nd Year Onwards] Based on CBCS system & OBE model
Recommended scheme of study
(For Diploma in Computer Engineering)

Semester of Study	Category of course	Course Code	Subjects	Mode of Delivery			Total Credits
				L	T	P	C
THIRD	**THEORY**						
	Program Core Course	DCE 301	Computer Programming	3	0	0	3
		DCE 303	Introduction to DBMS	3	0	0	3
		DCE 305	Computer Organization	3	0	0	3
		DCE 307	Mathematical Foundation for Computer Science	3	0	0	3
	Mandatory Course	DHS 301	**Universal Human Values- II**	2	1	0	**3**
	SESSIONAL						
	Program Core Course	DCE 302	Computer Programming Lab	0	0	2	1
		DCE 304	DBMS Lab	0	0	2	1
		DCE 306	Computer Organization Lab	0	0	2	1
		DCE 308	Linux Lab	0	0	2	1
		DCE 310	Web Technologies Lab	0	0	2	1
	Summer Internship	DSI 331	**Summer Internship- I (4 weeks) after II Semester**	0	0	0	0 (Non-credit)
TOTAL CREDITS							**20**
Total Lectures per Week				**25**			
FOURTH	**THEORY**						
	Program Core Course	DCE 401	Data Structures	3	0	0	3
		DCE 403	.NET	3	0	0	3
		DCE 405	Computer Networks	3	0	0	3
		DCE 407	Computer System Architecture	3	0	0	3
		DCE 409	Operating Systems	3	0	0	3
	Open Elective	OE-I: DOE 421/ DOE 422/ DOE 423		3	0	0	3
	SESSIONAL						
	Program Core Course	DCE 402	Data Structures Lab	0	0	2	1
		DCE 404	.NET Lab	0	0	2	1
		DCE 406	Computer Networking Lab	0	0	2	1
		DCE 408	Python Lab	0	0	2	1
	Project	DPR 431	Minor Project	0	0	4	1
	Mandatory Course	DAU 401	Essence of Indian Knowledge and Tradition	2	0	0	0 (Non-credit)
TOTAL CREDITS							**23**
Total Lectures per Week				**14**			
GRAND TOTAL FOR SECOND YEAR							**43**

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

PROGRAMME ELECTIVES (PE)

SEMESTER	Code no.	Name of the PE courses	Prerequisite/ Co-requisite courses	L	T	P	C
PE-I							
SEM-V	DPE521	Computer Graphics		3	0	0	3
	DPE522	Introduction to Cloud Computing		3	0	0	3
	DPE523	Object-oriented programming in C++		3	0	0	3
PE-II							
SEM- V	DPE524	Introduction to Computer Algorithms		3	0	0	3
	DPE525	Data Science		3	0	0	3
	DPE526	Multimedia and Animation		3	0	0	3
PE-III							
SEM- VI	DPE621	Foundations of AI/ML		3	0	0	3
	DPE622	Operations Research		3	0	0	3
	DPE623	Cyber Security		3	0	0	
PE-IV							
SEM- VI	DPE624	Internet of Things		3	0	0	3
	DPE625	Machine Learning		3	0	0	3
	DPE626	Computer Oriented Numerical and Statistical Methods		3	0	0	3

(RAMNISH SINHA)

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

OPEN ELECTIVES (OE)*

SEMESTER	Code no.	Name of the PE courses	Prerequisite/ Co-requisite courses with code	L	T	P	C
OE-I							
SEM-IV	DOE421	C Programming Language		3	0	0	3
	DOE422	Introduction to Python		3	0	0	3
	DOE423	Data Base Concepts		3	0	0	3
OE-II							
SEM- V	DOE521	Web Programming Concepts		3	0	0	3
	DOE522	Data Structures in C		3	0	0	3
	DOE523	PC Maintenance & Networking		3	0	0	3
OE-III							
SEM- VI	DOE621	Intro. to Computer Graphics		3	0	0	3
	DOE622	Intro. to Machine Learning		3	0	0	3
	DOE623	Introduction to Multimedia		3	0	0	3
*OPEN ELECTIVES TO BE OPTED ONLY BY OTHER DEPARTMENT STUDENTS							

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

DATA STRUCTURES

PROGRAMME: Diploma in Computer Engineering							
COURSECODE: DCE 401		COURSE TITLE: DATA STRUCTURES					
COMPULSORY: Program Core							
Teaching Scheme and Credits					EXAMINATION SCHEME		
L	T	P	HRS/ WK	CREDIT	PE	FINAL	TOTAL
3	0	0	3	3	50	50	100

RATIONALE:

1. Data structures are essential for organizing and storing data efficiently, allowing for optimized data retrieval and manipulation.
2. Understanding data structures helps students develop problem-solving skills by choosing the appropriate structure for a given problem.
3. Knowledge of data structures is fundamental for writing efficient algorithms, which is crucial in software development.
4. Mastery of data structures enables students to analyze the time and space complexity of algorithms, enhancing their ability to write performance-driven code.
5. Data structures are the backbone of many advanced topics in computer science, such as databases, operating systems, and artificial intelligence.

COURSE OUTCOMES:

At the end of the course students will be able to:

1.	Demonstrate proficiency in fundamental programming concepts such as arrays, functions, pointers, structures, and dynamic memory allocation.
2.	Apply abstract data types to organize and manipulate data efficiently, distinguishing between linear and non-linear data structures and employing sorting and searching algorithms effectively.
3.	Implement stack data structures, perform stack operations, understand Polish notations, and apply stacks to solve real-world problems.
4.	Implement queue data structures, perform queue operations, understand circular queues, and utilize queues in various scenarios.
5.	Implement linked list data structures, differentiate between different types of linked lists, understand their advantages and disadvantages, and apply linked lists to solve practical problems.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	**TITLE: Prerequisite of Data Structures** 1.1 Review of Arrays, 1.2 Functions and Pointers, 1.3 Overview of Structures and Union, 1.4 Dynamic Memory Allocation Course Outcome: CO1 Teaching Hours :8 hrs
2.	**TITLE: Introduction to Data Structures** 2.1 Abstract Data Types 2.2. Linear and Non-linear 2.4 Static and Dynamic Data Structures 2.5 Insertion sort, Bubble sort Algorithm & Linear and Binary searching Course Outcome: CO2 Teaching Hours :8 hrs
3.	**TITLE: Stacks** 3.1 Introduction to Stack 3.2 Operations on Stack 3.3 Implementation of Stack 3.4 Polish Notations- Infix, Postfix and Prefix 3.5 Applications of Stack Course Outcome: CO3 Teaching Hours :6 hrs
4.	**TITLE: Queues** 4.1 Introduction to Queue, 4.2 Operations on Queues, 4.3 Implementation of Simple Queue 4.4 Circular Queue 4.5 De-queue Course Outcome: CO4 Teaching Hours :8 hrs
5.	**TITLE: Linked List** 5.1 Introduction to Self-referential Structures 5.2 Linked-List and its advantages and disadvantages 5.3 Types of Linked-Lists (Singly-linked, Doubly-linked and Circularly-linked) 5.4 Implementation of Singly-Linked-List Course Outcome: CO5 Teaching Hours :10 hrs

REFERENCE BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Mastering data structure through 'C'	J.B. Dixit, Laxmi Publications, 1e	978-9380386720
2.	Data Structure using 'C'	Y. Kanetkar, BPB, 4th	978-9355511898
3.	Data Structure using C	E. Balaguruswamy, McGraw Hill.	978-1259029547

E-REFERENCES:

CO VS PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	2	2	3	0	1	3	3	3	2
CO2	2	3	3	3	0	1	3	3	3	2
CO3	2	3	3	3	0	1	3	3	3	2
CO4	2	3	3	3	0	1	3	3	3	2
CO5	2	3	3	3	0	1	3	3	3	2

SIGNATURES:

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

.NET

PROGRAMME: Diploma in Computer Engineering							
COURSECODE: DCE 403		COURSE TITLE: .NET					
COMPULSORY: Program Core							
Teaching Scheme and Credits					EXAMINATION SCHEME		
L	T	P	HOURS/WK	CREDIT	PE	FINAL	TOTAL
3	0	0	3	3	50	50	100

RATIONALE:

1. .NET provides a versatile framework supporting a wide range of applications, from web to mobile and cloud services.
2. It supports multiple programming languages, offering flexibility and promoting language interoperability.
3. Extensive libraries and tools streamline development, enhancing application quality and performance.
4. .NET Core and .NET 5+ enable cross-platform development, ensuring compatibility across various operating systems.
5. A large, active community and strong support from Microsoft offer valuable resources and assistance.

COURSE OUTCOMES:

At the end of the course students will be able to:

1.	Grasp the foundational concepts of the .NET framework, the role of CLR, and be able to develop console applications with basic input/output functionalities
2.	Develop Windows applications, demonstrating competence in form design, control manipulation, and properties management.
3.	Use various looping constructs in Visual Studio .NET and develop more dynamic and responsive applications.
4.	Design and implementing databases, incorporating features like relationships, stored procedures, functions, and triggers.
5.	Establishing database connectivity using ADO.NET, and effectively store and retrieve data from tables.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	**TITLE: Introduction to .NET** 1.1 .NET framework overview 1.2 Architecture of .NET Framework 1.3 Common Language Runtime CLR, Just in Time Compiler 1.4 Console applications 1.5 Input/ Output functions Course Outcome: CO1 Teaching Hours: 6 hrs
2.	**TITLE: Visual Studio .NET** 2.1 Introduction to Windows Application 2.2 Form design, properties window 2.3 Overview of Controls- Button, Label, textbox, combo box, list box etc 2.4 Control statements- if… then, if… then… else, if… then… elseif… endif Course Outcome: CO2 Teaching Hours: 8 hrs
3.	**TITLE: Visual Studio .NET- Looping** 3.1 for….next loop 3.2 while…wend 3.3 do….while 3.4 do….until Course Outcome: CO3 Teaching Hours: 10 hrs
4.	**TITLE: MY-SQL** 4.1 Intro to My-SQL 4.2 Features of My-SQL 4.3 Designing database, Designing tables 4.4 Relationship, Stored Procedures 4.5 Functions, Triggers Course Outcome: CO4 Teaching Hour: 10 hrs
5.	**TITLE: Database Connectivity** 5.1 Overview of ADO.NET 5.2 Components of ADO.NET, 5.3 Database connectivity with ADO.NET 5.4 Store data in a table 5.5 Fetching data from a table Course Outcome: CO5 Teaching Hours: 6 hrs

REFERENCE BOOKS:

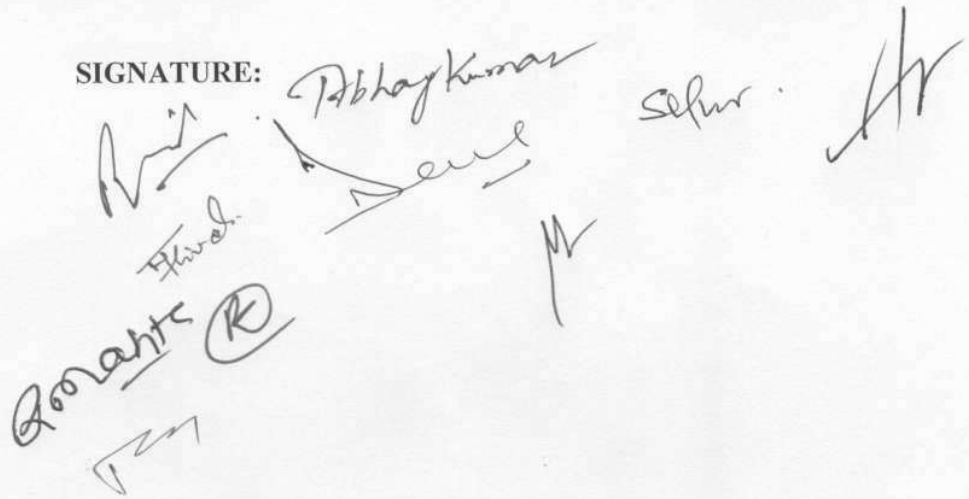
S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Visual Basic .NET Programming	Harold Davis, SYBEX, 2002e	978-0-782-14038-5

E-REFERENCES:

CO VS PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	1	2	2	0	0	3	3	2	2
CO2	2	1	3	2	0	1	3	2	3	2
CO3	2	2	2	3	0	0	3	2	3	3
CO4	1	2	3	3	0	1	3	2	3	3
CO5	1	2	3	3	0	1	3	2	3	3

SIGNATURE:



UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

COMPUTER NETWORKS

PROGRAMME: Diploma in Computer Engineering							
COURSECODE: DCE 405		COURSE TITLE: Computer Networks					
COMPULSORY: Program Core							
Teaching Scheme and Credits					EXAMINATION SCHEME		
L	T	P	HOURS/W	CREDIT	PE	FINAL	TOTAL
3	0	0	3	3	50	50	100

RATIONAL:

1. Provides a solid foundation in networking principles, crucial for designing and managing modern communication systems.
2. Equips students with knowledge of key protocols and standards (such as TCP/IP), ensuring efficient and interoperable data exchange.
3. Introduces essential network security practices, preparing students to protect data integrity and privacy in digital environments.
4. Lays the groundwork for advanced topics like cloud computing and cybersecurity, opening diverse career opportunities in IT and network management.

COURSE OUTCOMES:

Students should be able to

1.	Understand various network topologies, devices, addressing schemes, and terminologies essential for computer networking.
2.	Demonstrate knowledge of different transmission modes, transmission media, and the characteristics of guided and wireless transmission.
3.	Apply concepts of data encoding and communication interfaces to effectively transmit digital and analog data using appropriate signals.
4.	Implement flow control, error control mechanisms, and switching techniques to ensure reliable data communication.
5.	Analyze the TCP/IP protocol stack, its architecture, and the functionalities of different layers, along with the usage of common protocols at each layer.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	**TITLE: Network Fundamentals** 1.1 Network Topology 1.2 Network devices: Ethernet card, Hub, Switch, Bridge, Router, Gateway; 1.3 Addressing: Physical address, Logical address, Classes of IP address, Subnet Mask; 1.4 Terminology: unicasting, multicasting, broadcasting, broadband, point-to-point, multipoint. Course Outcome: CO1 Teaching Hours: 8 hrs
2.	**TITLE: Data transmission and Transmission Media** 2.1 Transmission modes: Simplex, Half duplex, full duplex 2.2 Transmission impairments 2.3 Guided transmission media- Coaxial cable, UTP, STP, OFC; 2.4 Wireless transmission- Infrared, Radio waves, Microwaves Course Outcome: CO2 Teaching Hours: 8 hrs
3.	**TITLE: Data Encoding and Communication Interface** 3.1 Digital data-digital signals 3.2 Digital data-analog signals 3.3 Analog data-digital signals 3.4 Analog data-analog signals Course Outcome: CO3 Teaching Hours: 8 hrs
4.	**TITLE: Flow Control, Error Control and Switching techniques** 4.1 Flow and Error Control 4.2 Simple, Stop and wait 4.3 Stop and Wait ARQ 4.4 GO-Back-N ARQ 4.5 Selective repeat 4.6 High level data link control 4.7 Introduction to Switching Techniques- Circuit-, Packet-, and Message-switching Course Outcome: CO4 Teaching Hour: 8 hrs
5.	**TITLE: TCP/IP Protocols Stack** 5.1 Introduction to TCP/IP protocol suite 5.2 Protocols and Architecture 5.3 Application layer protocols: TELNET, HTTP, SMTP, FTP, POP, DNS, DHCP 5.4 Transport layer protocols: TCP, UDP, SCTP 5.5 Network layer protocols: ICMP, IGMP, ARP, RARP, IP Course Outcome: CO5 Teaching Hours: 8 hrs

REFERENCE BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Data Communications And Networking	Behrouz A. Forouzan McGraw-Hill, 4e	978-0-07-296775-3
2.	Data and Computer Communications	William Stallings, Pearson Education India, 9e	978-9332518865

E-REFERENCES:

CO VS PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	2	2	3	0	2	3	3	2	3
CO2	3	3	2	3	0	2	3	3	2	3
CO3	2	3	2	3	0	2	3	2	2	3
CO4	2	3	2	3	0	2	3	2	2	3
CO5	3	3	2	3	0	2	3	3	2	3

SIGNATURE:

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

COMPUTER SYSTEM ARCHITECTURE

PROGRAMME: Diploma in Computer Engineering							
COURSECODE: DCE 407			COURSE TITLE: COMPUTER SYSTEM ARCHITECTURE				
COMPULSORY: Program Core							
Teaching Scheme and Credits					EXAMINATION SCHEME		
L	T	P	HRS/ WK	CREDIT	PE	FINAL	TOTAL
3	0	0	3	3	50	50	100

RATIONALE:

1. Understanding computer system architecture is essential for designing efficient and effective computer systems.
2. Knowledge of architecture principles helps students comprehend how software and hardware interact, optimizing system performance.
3. This subject provides a foundation for advanced topics like operating systems, compilers, and embedded systems.
4. Mastery of system architecture enables students to analyze and improve the efficiency of computing resources.
5. Skills in computer system architecture are crucial for roles in hardware design, system integration, and performance engineering.

COURSE OUTCOMES:

At the end of the course students will be able to:

1.	Understand the fundamental digital components such as flip-flops, shift registers, and counters.
2.	Grasp the basic organization of computers, including instruction codes, computer registers, and computer instructions.
3.	Understand the structure and functioning of the Central Processing Unit (CPU), including general register organization, stack organization, and instruction formats.
4.	Explore input-output (I/O) organization, including I/O interfaces, asynchronous data transfer methods, and modes of transfer.
5.	Study memory technology, including memory hierarchy, cache memory, mapping, and updation schemes.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	**TITLE: Digital components** 1.1 Review of Flip-flops 1.2 Excitation Table, State Tables and State Diagrams 1.3 Shift Registers 1.4 Bi-directional Shift Register with Parallel Load 1.5 Counters Course Outcome: CO1 — Teaching Hours: 8 hrs
2.	**TITLE: Basic Computer Organization** 2.1 Instruction Codes 2.2 Computer registers 2.3 Computer Instructions 2.4 Timing and Control 2.5 Instruction Cycle Course Outcome: CO2 — Teaching Hours: 8 hrs
3.	**TITLE: Central Processing Unit** 3.1 General Register Organization 3.2 Stack Organization 3.3 Instruction Formats 3.4 Addressing Modes 3.5 Data Transfer and Manipulation 3.6 Program Control Course Outcome: CO3 — Teaching Hours: 8 hrs
4.	**TITLE: Input Output Organization** 4.1 I/O Interface 4.2 Asynchronous Data Transfer 4.3 Modes of Transfer 4.4 Direct Memory Access 4.5 CPU-IOP Communication Course Outcome: CO4 — Teaching Hour: 8 hrs
5.	**TITLE: Memory Technology** 5.1 Memory Hierarchy 5.2 Cache Memory, Mapping and Updation Schemes 5.3 Virtual Memory and Memory Management Unit 5.4 Associative Memory 5.5 Cache Memory, Mapping, Updation Schemes Course Outcome: CO5 — Teaching Hours: 8 hrs

REFERENCE BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Computer System Architecture	Morris, M. M., Pearson, 3e, 2007	9788131700709
2.	Computer Organization and Design: The Hardware/Software Interface`	David A. Patterson, John L. Hennessy, Morgan Kaufmann, 5th Edition, 2013	978-0124077263
3.	Structured Computer Organization	Andrew S. Tanenbaum, Todd Austin, Pearson, 6th Edition, 2012	978-0132916523

E-REFERENCES:

CO VS PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	1	2	2	0	1	3	3	1	2
CO2	3	1	2	1	0	1	3	3	1	2
CO3	3	2	3	2	0	1	3	3	1	3
CO4	2	2	3	2	0	1	3	3	1	3
CO5	3	2	3	2	0	1	3	3	1	3

SIGNATURES:

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

DATA STRUCTURES LAB

PROGRAMME: DIPLOMA IN ENGINEERING							
COURSE CODE: DCE 402			COURSE TITLE: Data Structures Lab				
COMPULSORY/OPTIONAL: Program Core							
Teaching Scheme and Credits					Examination scheme		
L	T	P	Hrs./Wk	Cred.	PE	FINAL	Total
0	0	2	2	1	60	40	100

RATIONALE:

1. Allows students to implement and experiment with data structures covered in theory classes, such as arrays, linked lists, stacks, queues, trees, and graphs.
2. Enhances students' ability to solve real-world problems efficiently using appropriate data structures, fostering algorithmic thinking and optimization techniques.
3. Provides practical experience in implementing and manipulating data structures through programming exercises, helping students understand their strengths, weaknesses, and performance characteristics.
4. Teaches students how to debug and optimize data structure implementations, improving their proficiency in identifying and resolving programming errors and inefficiencies.

COURSE OUTCOMES:

After completing this students should be able to:

1.	Demonstrate proficiency in fundamental programming concepts such as arrays, functions, pointers, structures, and dynamic memory allocation.
2.	Apply abstract data types to organize and manipulate data efficiently, distinguishing between linear and non-linear data structures and employing sorting and searching algorithms effectively.
3.	Implement stack data structures, perform stack operations.
4.	Implement queue data structures, perform queue operations, understand circular queues.
5.	Implement linked list data structures, differentiate between different types of linked lists.

COURSE CONTENT DETAILS:

MODULE	TOPICS/SUBTOPICS
1.	**TITLE: Prerequisite of Data Structures** 1.1 Write a program to create an array of integers and initialize it with values

	from 1 to 10. 1.2 Given an array of integers, write a function to print all the elements of the array. 1.3 Write a program to find the sum of all elements in a given array of integers. 1.4 Write a function to find the maximum element in an array of integers. 1.5 Write a program to reverse the elements of an array. 1.6 Write a function add that takes two integers as arguments and returns their sum. 1.7 Write a function multiply that takes two integers and returns their product. Set the second argument to a default value of 1. 1.8 Write a recursive function to calculate the factorial of a given number. 1.9 Write a function that swaps two integers using pass by reference. 1.10 Write a program to declare an integer variable, assign it a value, and then use a pointer to change its value. 1.11 Write a program to perform pointer arithmetic (increment and decrement pointers) on an array of integers. 1.12 Write a program to dynamically allocate memory for an array of integers, read values from the user, and then print the array. 1.13 Write a program to implement structure. Course Outcome: CO1 Teaching Hours: 6 hrs
2.	**TITLE: Sorting and searching:** 2.1 Write a program to demonstrate the difference between static arrays and dynamic arrays. Include operations to resize the dynamic array. 2. 2 Write a function to perform insertion sort on an array of integers. 2.3 Implement the bubble sort algorithm to sort an array of integers in ascending order. 2.4 Write a function to perform a linear search on an array. Return the index of the element if found, otherwise return -1. 2.5 Implement the binary search algorithm. The array should be sorted before performing the search. Return the index of the element if found, otherwise return -1. Course Outcome: CO2 Teaching Hours: 4 hrs

3.	**TITLE: Stack** 3.1 Write a function to perform the push operation on a stack. The function should take the stack and the element to be pushed as arguments. 3.2 Write a function to perform the pop operation on a stack. The function should return the popped element and handle the case when the stack is empty. 3.3 Write a function to perform the peek operation on a stack. The function should return the top element of the stack without removing it and handle the case when the stack is empty. 3.4 Write a function to check if the stack is empty. 3.5 Write a function to return the number of elements in the stack. 3.6 Implement a stack using an array with all the basic operations: push, pop, peek, is_empty, and size. Course Outcome: CO3 Teaching Hours: 4 hrs
4.	**TITLE: Queues** 4.1 Introduction to Queue. 4.2 Operations on Queues 4.3 Implementation of Simple Queue 4,4 Circular Queue 4.5 De-queue Course Outcome: CO4 Teaching Hour: 4 hrs
5.	**TITLE: Linked List:** 5.1 Introduction to Self-referential Structures 5.2 Linked-List and its advantages and disadvantages 5.3 Types of Linked-Lists (Singly-linked, Doubly-linked and Circularly-linked) 5.4 Implementation of Singly-Linked-List Course Outcome: CO5 Teaching Hours:6 hrs

REFERENCE BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Mastering data structure through 'C'	J.B. Dixit, Laxmi Publications, 1st Ed., January 2010.	978-9380386720
2.	Data Structure using 'C'	Y. Kanetkar, BPB Publications; 4th edition, March 2022.	978-9355511898
3.	Data Structure using C	E. Balaguruswamy, McGraw Hill Education (India) Pvt. Ltd., July 2017.	978-1259029547
4.	Data Structures And Algorithms Made Easy: Data Structures And Algorithmic Puzzles	Narasimha Karumanchi, Careermonk Publications, July 2023	978-8193245286

E-REFERENCES:

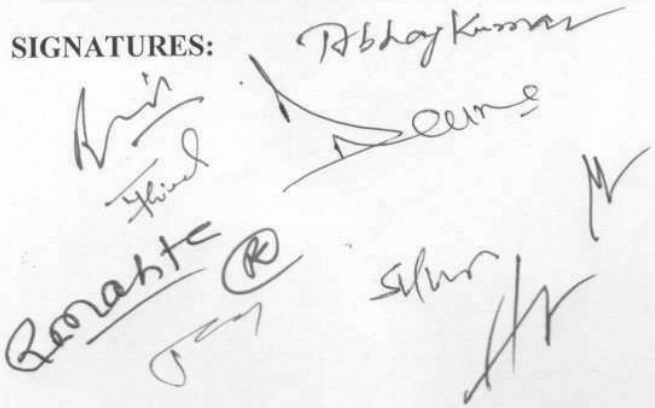
1. https://www.cs.bham.ac.uk/~jxb/DSA/dsa.pdf

2. Data Structure using C Programming, By Dr. Dipti Verma and Mr. Aditya Tiwari Course (swayam2.ac.in)https://onlinecourses.swayam2.ac.in/nou24_cs15/preview

CO VS PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	3	2	3	0	1	3	3	3	3
CO2	3	3	3	3	0	1	3	3	3	3
CO3	2	3	2	3	0	1	3	2	3	3
CO4	2	3	2	3	0	1	3	2	3	3
CO5	3	3	2	3	0	1	3	3	3	3

SIGNATURES:



UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

.NET Lab

PROGRAMME: Diploma in Computer Engineering							
COURSE CODE: DCE 402		COURSE TITLE: .NET Lab					
COMPULSORY/OPTIONAL: COMPULSORY							
Teaching Scheme and Credits					Examination scheme		
L	T	P	Hrs./Wk	Cred.	PE	FINAL	TOTAL
0	0	2	2	1	60	40	100

RATIONALE:

1. Provides hands-on experience in developing applications using the .NET framework, reinforcing theoretical knowledge with practical skills.
2. Equips students with the ability to write and debug programs in C#, fostering strong problem-solving and programming abilities.
3. Teaches the development of both web and desktop applications, broadening students' capabilities and versatility in software development.
4. Enables students to integrate applications with databases using ADO.NET, essential for building robust and data-driven applications.
5. Prepares students for careers in software development by providing practical experience with industry-standard tools and technologies, making them valuable assets in the job market.

COURSE OUTCOMES:

Students should be able to:

1.	Demonstrate the ability to create console applications and effectively perform operations like accepting Input from user, perform arithmetic operations.
2.	Acquire proficiency in using essential VB.NET controls. Understand the concept of VB.NET applications and effectively design the forms in Windows applications, ensuring smooth functionality and user experience.
3.	Acquire proficiency in using essential VB.NET controls like Adrotator, Calendar, and Tree view to display messages, manage vacation schedules, and perform various tree view operations in web applications.
4.	Successfully design database, tables, query, relationship, import and export data in My-SQL.
5.	Gain hands-on experience in data binding database connectivity of windows forms with My-SQL.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	TITLE: Console Application: 1.1 Write a console application to display name 1.2 Write a console application to display name, address 1.3 Write a console application to display sum , product of two numbers 1.4 Write a console application to accept input of two numbers and display sum and product 1.5 Write a console application to accept input of a numbers and check its even or odd 1.6 Write a console application to accept input of three numbers and find the largest number. 1.7 Write a console application to accept input of three numbers and find the smallest number. Course Outcome: CO1 Teaching Hours: 4 hrs
2.	TITLE: Windows Application: 2.1. Creating Windows Application 2.2. Understanding different windows -design, properties window, tool box 2.3. Inserting controls in a form- Button, Label, textbox, combo box, list box etc 2.4. Changing properties of controls in a form- Button, Label, textbox, combo box, list box etc 2.5. Write a VB.Net program that takes a student's score as input and displays the grade obtained by the student. 2.6. Create a VB.Net program that asks the user to enter a number. If the number is positive, display "Positive Number". If the number is negative, display "Negative Number". If the number is zero, display "Zero". 2.7. Develop a VB.Net program that prompts the user to enter a year. Check if the entered year is a leap year or not. Display an appropriate message indicating whether it's a leap year or not. 2.8. Write a VB.Net program that asks the user to enter their age. Based on the age entered, classify them into different age groups such as "Child" (0-12 years), "Teenager" (13-19 years), "Adult" (20-64 years), and "Senior" (65 years and above). 2.9. Develop a VB.Net program that calculates the ticket price for a movie theater based on the age of the customer. If the customer is under 12 or over 65, apply a discount of 50%. For customers between 12 and 65, display the regular ticket price. 2.10. Write a VB.Net program that asks the user to enter the current temperature. Based on the temperature range, provide advice such as "Wear a coat" for temperatures below 10°C, "Wear a sweater" for temperatures between 10°C and 20°C, and "Enjoy the weather" for temperatures above 20°C. Course Outcome: CO2 Teaching Hours: 6 hrs

3.	TITLE: Visual Studio .NET- Looping: 3.1 Write a VB.Net program to display your name 10 times using a For...Next loop. 3.2 Write a VB.Net program to display all numbers from 1 to 100 using a For...Next loop. 3.3 Write a VB.Net program to calculate the sum of all numbers from 1 to 100 using a For...Next loop. 3.4 Create a VB.Net program to display all even numbers between 1 and 50 using a While...End While loop. 3.5 Develop a VB.Net program to calculate the factorial of a given number using a Do While...Loop. 3.6 Write a VB.Net program to check if a given number is prime or not using a Do Until...Loop. 3.7 Create a VB.Net program that prompts the user to enter a number and displays its multiplication table (up to 10) using a For...Next loop. 3.8 Develop a VB.Net program to generate the Fibonacci series up to a specified limit using a Do While...Loop. 3.9 Create a VB.Net program to calculate the power of a number using exponentiation using a For...Next loop. 3.10 Develop a VB.Net program that converts temperatures from Celsius to Fahrenheit for values between 0 and 100 degrees Celsius using a While...End While loop. 3.11 Write a VB.Net program that generates a random number between 1 and 100. Allow the user to guess the number and provide feedback using a Do While...Loop until they guess correctly. Course Outcome: CO3 Teaching Hours: 6 hrs
4.	TITLE: MY-SQL 4.1 Installation and configuration of XAMP/WAMP server 4.2 Overveiw of My-SQL. 4.3 Create database 4.4 Create tables 4.5 Create relationship between tables 4.6 Create Stored Procedure 4.7 Create Function 4.8 Create Trigger Course Outcome: CO4 Teaching Hour: 4 hrs
5.	TITLE:Database Connectivity 5.1 Design a Windows from 5.2 Database connectivity with ADO.NET 5.3 Store data in a table 5.4 Fetching data from a table 5.5 Update data stored in a table Course Outcome: CO5 Teaching Hours:4 hrs

REFERENCE BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Visual Basic .NET Programming	Harold Davis, SYBEX, 2002	978-0-782-14038-5
2.	Visual Basic.Net	Shirish Chavan, Pearson, 2004	9.78813E+12
1.	VISUAL BASIC.NET: THE COMPLETE REFERENCE	Jeremy R. Shapiro, McGraw Hill Education, 1st, 2017	9780070495111

F-REFERENCES:

https://www.vbtutor.net/vb2019/vb2019hndbk_preview.pdf

CO VS PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	2	1	3	0	0	3	3	2	3
CO2	3	2	3	3	0	1	3	3	2	3
CO3	3	2	3	3	0	1	3	3	2	3
CO4	3	3	3	3	0	0	3	3	3	3
CO5	3	2	3	3	0	0	3	3	3	3

SIGNATURES:

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

COMPUTER NETWORKING LAB

PROGRAMME: Diploma in Computer Engineering							
COURSE CODE: DCE 406			COURSE TITLE: Computer Networking Lab				
COMPULSORY/OPTIONAL: COMPULSORY							
Teaching Scheme and Credits					Examination scheme		
L	T	P	Hrs./Wk	Cred.	PE	FINAL	TOTAL
0	0	2	2	1	60	40	100

RATIONALE:

1. Provides hands-on experience with networking hardware and software, reinforcing theoretical concepts learned in lectures.
2. Equips students with practical skills in configuring and managing network devices such as routers, switches, and firewalls.
3. Enhances problem-solving abilities by allowing students to design, implement, and troubleshoot network topologies and protocols.
4. Offers insights into network security practices, including setting up VPNs, implementing encryption, and managing access controls.
5. Prepares students for careers in network administration and IT support by providing real-world networking experience, making them valuable assets in the job market.

COURSE OUTCOMES:

Students should be able to:

1.	Identify, connect, and troubleshoot various network devices and connectors, prepare network cables, and use command line diagnostics effectively.
2.	Connect and configure PCs in a network, setting up workgroups, and managing file and print sharing with appropriate security settings.
3.	Installing server operating systems and configuring Active Directory to manage network resources and user permissions.
4.	Create and manage user accounts, assign administrative roles, and configure user permissions to ensure secure and efficient network administration.
5.	Perform disk management tasks, like creating simple, spanned, and striped volumes, to optimize storage solutions and enhance data organization.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	**TITLE: Network Components** 1.1 Identification of various network components/devices e.g. Connectors, Hub, Switch, Modem 1.2 Preparation of cross and parallel cable. 1.3 Setting IP address. 1.4 Using command line diagnostics: ipconfig and ping. Course Outcome: CO1 Teaching Hours: 4 hrs
2.	**TITLE: Setting-up of small home/office network:** 2.1 Connecting PCs in a network. 2.2 Configuring PCs in a network. 2.3 Creating workgroup. 2.4 File and print sharing. 2.5 Setting-up file sharing options (read/write/full control). 2.6 Setting-up print sharing options. 2.7 Installation of network printer. 2.8 Configuring and managing computer security. 2.9 Account lockout. 2.10 Password policy. 2.11 Audit policy. 2.12 User Rights Assignment. 2.13 Security Options. Course Outcome: CO2 Teaching Hours: 6 hrs
3.	**TITLE: Server OS** 3.1 Installation of server Operating system. 3.2 Installation of Active directory. 3.3 Configuring access permissions. Course Outcome: CO3 Teaching Hours: 4 hrs
4.	**TITLE: Managing user accounts& Remote Assistance** 4.1 Creating user accounts 4.2 Making a user account member of Administrative group. 4.3 Assigning permissions 4.4 Remote Assistance Course Outcome: CO4 Teaching Hour: 4 hrs
5.	**TITLE: Disk Management, file sharing & Managed Switch configuration:** 5.1 Create simple volume 5.2 Create spanned volume 5.3 Create striped volume 5.4 Create Mirrored volume 5.5 Managed Switch Configuration Course Outcome: CO5 Teaching Hours:6 hrs

REFERENCE BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Windows Server 2022 Administration Fundamentals	Bekim Dauti, Packt Publishing, First, 2022	9781803232157
2.	Mastering Windows Server 2022	Jordan Krause, Packt Publishing, First, 2023	9781837633425
3.	Practical Guide to Advanced Networking	Jeffery S. Beasley & Piyasat Nilkaew, Pearson Education,Third, 2012	9789332515895

G-REFERENCES:

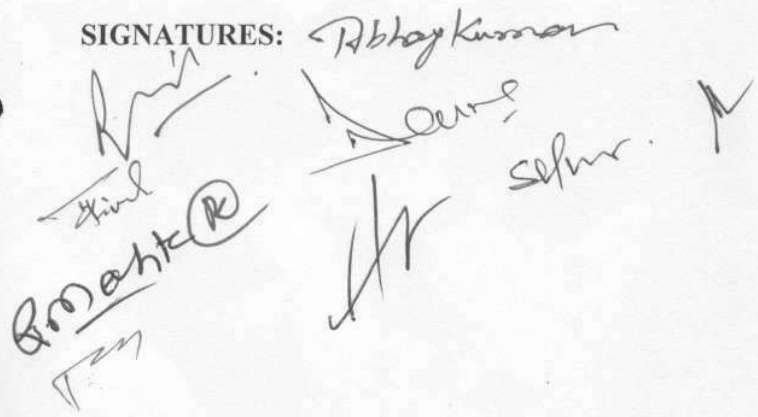
https://learn.microsoft.com/en-us/windows-server/

https://ptgmedia.pearsoncmg.com/images/9780789749048/samplepages/0789749041.pdf

CO-PO/PSO MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	2	1	3	0	2	3	3	2	3
CO2	3	2	3	3	0	2	3	3	2	3
CO3	3	2	3	3	0	2	3	3	2	3
CO4	3	3	3	3	0	2	3	3	3	3
CO5	3	2	3	3	0	2	3	3	3	3

SIGNATURES:



UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

PYTHON LAB

PROGRAMME: Diploma in Computer Engineering							
COURSE CODE:DCE 408				COURSE TITLE: Python Lab			
COMPULSORY/OPTIONAL: COMPULSORY							
Teaching Scheme and Credits					Examination scheme		
L	T	P	Hrs./Wk	Cred.	PE	FINAL	TOTAL
0	0	2	4	1	60	40	100

RATIONALE:

1. Reinforces theoretical concepts by providing hands-on experience in writing, compiling, and debugging Python programs.
2. Enhances problem-solving skills through developing programs that use control flow mechanisms, functions, objects, and classes.
3. Introduces GUI development with Tkinter, enabling students to create interactive applications.
4. Teaches handling of data structures (lists, tuples, sets, dictionaries) and file operations, preparing students for real-world programming challenges.
5. Builds a strong foundation in Python, opening career opportunities in software development, data analysis, and web development.

COURSE OUTCOMES:

1.	Write programs demonstrating fundamental concepts of Python programming.
2.	Write simple programs involving control flow mechanisms in Python, including iterative (loops) and conditional (if-else) statements.
3.	Write simple programs involving concepts of functions, objects, and classes.
4.	Develop simple graphical user interface (GUI) using Tkinter (the standard GUI toolkit for Python), creating interactive applications and incorporating widgets.
5.	Write simple programs involving various operations on lists, tuples, sets and dictionaries and also file handling operations.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	**TITLE: Variables, I/O and Expressions** 1.1. Write a program that converts temperature from Celsius to Fahrenheit and vice versa. The program should prompt the user to enter the temperature and

	the unit they are converting from. 1.2 Write a Python program that solves a quadratic equation ax2 + bx + c = 0. Prompt the user to enter the coefficients (a, b, c) and display the roots. 1.3 Write a program that counts the number of words in a given sentence. Prompt the user to enter the sentence and display the word count. 1.4 Write a program that calculates the Body Mass Index (BMI) using the formula BMI = weight(kg) / (height(m))2. Prompt the user to enter their weight in kilograms and height in meters. 1.5 Write a program that calculates the simple interest. Prompt the user to enter the principal amount, the rate of interest, and the time period in years. Course Outcome: CO1 Teaching Hours: 6 hrs
2.	**TITLE: Control Statements** 2.1 Write a program that calculates the factorial of a given number. Use a while loop to iterate through the numbers and calculate the factorial. 2.2 Write a program that checks if a given number is prime or not. Prompt the user to enter the number and display the result using nested if statements. 2.3 Write a program that prints the multiplication table of a given number. Use a for loop to iterate through the numbers and print the table. 2.4 Write a program that generates a random number between 1 and 100 and asks the user to guess it. Provide hints if the guess is too high or too low. Use a while loop for continuous guessing until the correct number is guessed. 2.5 Write a program that validates a user-entered password based on the following criteria: at least 8 characters long, contains at least one uppercase letter, one lowercase letter, one digit, and one special character. Use a combination of if-else statements and boolean expressions to validate the password. Course Outcome: CO2 Teaching Hours: 6 hrs
3.	**TITLE: Functions, Objects and Classes** 3.1 Define a function to calculate the factorial of a number recursively. Call the function to compute the factorial of a user-provided number. 3.2 Define a Python class called Circle with attributes radius and methods to calculate the area and circumference of the circle. Create an object of the Circle class, set its radius, and call the methods to display the area and circumference. 3.3 Create a class called Student with attributes name and age. Implement methods to set and get these attributes. Instantiate a Student object, set its attributes, and display them. 3.4 Create a class called Employee with attributes name, salary, and position. Implement a method to give a raise to the employee's salary. Create an object of the Employee class, set its attributes, give a raise, and display the updated salary.

	Course Outcome: CO3	Teaching Hours: 6 hrs
4.	**TITLE: GUI Programming with Tkinter** 4.1 Create a simple Tkinter GUI application that displays the text "Hello, World!" in a window. 4.2 Create a basic calculator application using Tkinter with buttons for numbers, arithmetic operations, and a display to show the result of calculations. 4.3 Create a GUI application that allows users to convert temperature between Celsius and Fahrenheit. Include entry fields for input and labels to display the converted temperature. 4.4 Create a Tkinter application that displays an image using the PhotoImage widget. Add buttons to navigate through a folder of images. 4.5 Create a GUI application using Tkinter that allows users to browse files and directories on their system. Display the file structure in a treeview widget. 4.6 Use a standard dialog boxes like message box, file dialog, and color dialog into a Tkinter application. Use them to display messages, open files, and choose colors interactively. Course Outcome: CO4 Teaching Hour: 6 hrs	
5.	**TITLE: Advanced Data Structures & File Handling** 5.1 List Functions: Create functions that operate on lists, such as a function to find the maximum element, the minimum element, the sum of elements, and the average of elements in a list. 5.2 2D List Operations: Develop a program that works with a 2D list, such as finding the sum of each row and column, transposing the matrix, or finding the diagonal elements. 5.3 File Writing: Write a Python program that creates a text file and writes multiple lines of text to it using the write method. 5.4 File Reading: Develop a program that reads a text file line by line using the readline method and prints each line to the console. 5.5 File Reading and Writing: Create a program that reads data from one text file, processes it (e.g., calculates statistics), and writes the results to another text file. Course Outcome: CO5 Teaching Hours: 6 hrs	

TEXT BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Introduction to Programming Using Python	Liang Y. Daniel Pearson, 1e, 2017	978-9332551848
2.	A Beginners Guide to Python 3 Programming	John Hunt Springer Nature	978-3-030-20290-3
1.	Introduction to Computation and Programming Using Python	John V. Guttag, The MIT Press, Cambridge, Massachusetts	

E-REFERENCES:

CO VS PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	2	1	3	0	0	3	3	2	3
CO2	3	2	3	3	0	1	3	3	2	3
CO3	3	2	3	3	0	1	3	3	2	3
CO4	3	3	3	3	0	0	3	3	3	3
CO5	3	2	3	3	0	0	3	3	3	3

SIGNATURES:

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

C PROGRAMMING LANGUAGE

PROGRAMME: Diploma in Computer Engineering							
COURSE CODE: DOE 421			COURSE TITLE: C Programming Language				
COMPULSORY/ OPTIONAL: OPEN ELECTIVE							
Teaching Scheme and Credits					EXAMINATION SCHEME		
L	T	P	HRS/WK	CREDIT	PE	FINAL	TOTAL
3	0	0	3	3	50	50	100

RATIONALE:

1.	To demonstrate proficiency in algorithm development, programming debugging, understanding the structure of a C program, and utilizing various data types, variables, and constants effectively.
2.	To enable students to apply arithmetic operators, increment and decrement operations, assignment operators, and develop a clear understanding of operator precedence and associativity.
3.	To enable them to use conditional statements and various looping constructs to control the flow of program execution effectively.
4.	To enable them to use iterative constructs efficiently, along with understanding the concepts of continue and break statements for iteration control.
5.	To train them to perform operations on array elements, define and use functions, manage local and global variables.

COURSE OUTCOMES:

At the end of the course students will be able to:

1.	Demonstrate proficiency in algorithm development, programming debugging, understanding the structure of a C program, and utilizing various data types, variables, and constants effectively.
2.	Apply arithmetic operators, increment and decrement operations, assignment operators, and have a clear understanding of operator precedence and associativity.
3.	Use conditional statements like if(), nested if(), switch()...case, and various looping constructs to control the flow of program execution effectively.
4.	Use for(), while(), and do...while() loops efficiently, along with understanding the concepts of continue and break statements for loop control.
5.	Declare and initialize arrays, perform operations on array elements, define and use functions, manage local and global variables.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	**TITLE:** Programming techniques and overview of c language 1.6 Algorithm and programming development, program debugging 1.7 Structure of a C program, flowcharts 1.8 Data-types, variables and constants, Preprocessors. 1.9 Declaration and initialization of variables, sizeof() operator 1.10 I/O Functions (Formatted and Unformatted) Course Outcome: CO1 Teaching Hours: 8 hrs
2.	**TITLE:** Operators and Expressions 2.11 Arithmetic operators 2.12 Increment and decrement 2.13 Assignment operators, type casting 2.14 Precedence and associativity 2.15 Math functions Course Outcome: CO2 Teaching Hours: 8 hrs
3.	**TITLE:** Decision Making , Branching and Looping: 3.10 if() statement 3.11 Relational and logical operators, conditional operators 3.12 Nested if() statement, if()… else if() ladder 3.13 switch() … case statement, default and break Course Outcome: CO3 Teaching Hours: 8 hrs
4.	**TITLE:** Iterative (Looping) Statements 4.11 for() loop 4.12 while() loop 4.13 do… while() loop 4.14 continue and break statements Course Outcome: CO4 Teaching Hours: 8 hrs
5.	**TITLE:** Arrays, Functions 5.13 Declaration and initialization of 1D arrays 5.14 Operations on array elements 5.15 Idea of functions- Function Declaration, Defining functions, Function call Course Outcome: CO5 Teaching Hours: 8 hrs

REFERENCE BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Programming in C	Kamthane, A. N., Kamthane, A. A. Pearson, 3e	978-9332543553
2.	Programming with C	Gottfried, B. S., McGraw Hill (Schaum's Outlines), 4e	978-9353160272
3.	Programming in ANSI C	Balagurusamy, E., McGraw Hill.8e	978-9351343202

CO VS PO MAPPING:

CO/PO-PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	3	2	3	0	1	2	3	3	3
CO2	3	3	2	3	0	1	2	3	3	2
CO3	3	3	3	3	0	1	2	3	3	3
CO4	3	3	3	3	0	1	2	3	3	3
CO5	3	3	3	3	0	1	2	3	3	3

SIGNATURES:

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

INTRODUCTION TO PYTHON

PROGRAMME: Diploma in Computer Engineering							
COURSECODE: DOE 422		COURSE TITLE: INTRODUCTION TO PYTHON					
COMPULSORY/OPTIONAL: OPEN ELECTIVE							
Teaching Scheme and Credits					EXAMINATION SCHEME		
L	T	P	HRS/ WK	CREDIT	PE	FINAL	TOTAL
3	0	0	3	3	50	50	100

RATIONALE:

1. Python is a versatile and widely-used programming language known for its readability and simplicity, making it an ideal choice for beginners in programming.
2. The language supports multiple programming paradigms, including procedural, object-oriented, and functional programming, providing a comprehensive foundation for students.
3. Python's extensive standard library and active community support enable students to quickly develop solutions for a wide range of applications, from web development to data analysis.
4. Learning Python opens opportunities in various fields such as data science, artificial intelligence, machine learning, and automation, reflecting current industry trends and demands.
5. Python's use in educational, scientific, and professional settings enhances students' career prospects, making it a valuable skill for both academic and professional growth.

COURSE OUTCOMES:

At the end of the course students will be able to:

1.	Demonstrate proficiency in writing Python programs to solve simple problems, applying concepts such as variable assignment, expressions, and common functions effectively.
2.	Construct and implement control structures in Python, including if statements, loops, and other control flow mechanisms, to manage program execution efficiently.
3.	Showcase competency in defining and implementing functions, understanding variable scopes, and utilizing classes and objects to organize and manipulate data effectively in Python programs.
4.	Exhibit proficiency in creating graphical user interfaces (GUIs) using Tkinter in Python, including event handling, widget management, and basic interaction design.

5.	Work with advanced data structures such as lists, tuples, sets, and dictionaries, as well as file handling operations.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	**TITLE: Introduction to Python** 1.11 Writing a Simple Program 1.12 Reading Input from the Console 1.13 Identifiers, Variables, Assignment Statements and Expressions 1.14 Numeric Data-type and Operators 1.15 Evaluating Expressions and Operator Precedence 1.16 Common Functions and Strings 1.17 Objects and Methods Course Outcome: CO1 Teaching Hours: 8 hrs
2.	**TITLE: Control Statements** 2.16 Boolean Data type and Relational Expressions 2.17 if Statements, Two-Way selection, 2.18 Nested if, and Multi-Way Selection 2.19 Loops: while and for loops, 2.20 Nested loops, break and continue statements Course Outcome: CO2 Teaching Hours: 8 hrs
3.	**TITLE: Functions, Objects and Classes** 3.14 Defining and Calling a Function 3.15 The Scope of Variables 3.16 Defining Classes for Objects 3.17 Accessing Members of Objects, The self-Parameter 3.18 Examples using Classes Course Outcome: CO3 Teaching Hours: 8 hrs
4.	**TITLE: GUI Programming with Tkinter** 4.15 Introduction to Tkinter, Processing Events, 4.16 The Widget Classes and Canvas. 4.17 The Geometry Managers, The Place Manager, and The Pack Manager. 4.18 Displaying Images, Mouse, Key Events, and Bindings, Standard Dialog Boxes. Course Outcome: CO4 Teaching Hour: 8 hrs
5.	**TITLE: Advanced Data Structures & File Handling** 5.16 List and List Operations 5.17 Using List with Functions 5.18 Working with 2D Lists 5.19 Tuples, Sets and Dictionaries 5.20 File Handling with open, write, read, readline, readlines methods

	Course Outcome: CO5	Teaching Hours: 8 hrs

REFERENCE BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Introduction to Programming Using Python	Liang Y. Daniel , Pearson, 1e, 2017	978-9332551848
2.	A Beginners Guide to Python 3 Programming	John Hunt, Springer Nature	978-3030202903
3.	Introduction to Computation and Programming Using Python	John V. Guttag, The MIT Press, Cambridge, Massachusetts	978-0262529624

E-REFERENCES:

CO VS PO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	2	2	3	0	0	2	3	3	3
CO2	2	3	3	3	0	0	2	3	3	3
CO3	3	2	3	3	0	0	2	3	3	3
CO4	2	2	3	3	0	0	2	3	3	3
CO5	3	3	3	3	0	0	2	3	3	3

SIGNATURES:

UNIVERSITY POLYTECHNIC
BIRLA INSTITUTE OF TECHNOLOGY MESRA – 835215 (RANCHI)
DEPARTMENT OF COMPUTER ENGINEERING

DATABASE CONCEPTS

PROGRAMME: Diploma in Computer Engineering							
COURSECODE: DOE 423			COURSE TITLE: Introduction to DBMS				
COMPULSORY/OPTIONAL: OPEN ELECTIVE							
Teaching Scheme and Credits					EXAMINATION SCHEME		
L	T	P	HOURS/WK	CREDIT	PE	FINAL	TOTAL
3	0	0	3	3	50	50	100

RATIONAL:

1.	To provide an overview of database applications, the purpose of database systems, various data models and types of users.
2.	To introduce the Entity-Relationship (ER) model, focusing on entities, relationships, ER design issues and related concepts.
3.	To explain different operations used in relational query languages- Relational Algebra and Relational Calculus, and Structured Query Language (SQL).
4.	To discuss database normalization principles, addressing redundancy, anomalies, functional dependencies and lossless decomposition.
5.	To explore security and integrity in databases, including backup and recovery strategies.

COURSE OUTCOMES:

At the end of the course Students will be able to:

1.	Understand the purpose of using database systems and various data models and types of users.
2.	Create Entity-Relationship (ER) diagram with their understanding of entities and relationship existing among them.
3.	Form statements in relational query languages- Relational Algebra and Relational Calculus, and Structured Query Language (SQL), for different specific needs.
4.	Apply the principles of normalization to decompose the table preventing redundancy and anomalies.
5.	Understand the idea of security and integrity in databases, including backup and recovery strategies.

COURSE CONTENT DETAILS:

Module	Topics/Subtopics
1.	**TITLE:** Introduction to Database 1.18 Database Applications, Purpose of Database Systems, 1.19 View of Data – Data Abstraction – Instances and Schemas , Data Models – the ER Model – Relational Model – Other Models 1.20 Transaction Management – data base Architecture – Storage Manager 1.21 Database Languages – DDL, DML, DCL– Database Access for applications Course Outcome: CO1 Teaching Hours :8 hrs
2.	**TITLE:** Data base design 2.21 Entity Relationship Model 2.22 Relationships and Relationship sets 2.23 ER Design Issues– Weak Entity Sets, ER Database schema, ER Schema-to-tables, Keys-Schema Diagrams Course Outcome: CO2 Teaching Hours :8 hrs
3.	**TITLE:** Relational Query Languages 3.19 Relational Algebra– Selection, Projection and Set Operations 3.20 Relational calculus– Tuple Relational Calculus, Domain relational calculus. 3.21 Structured Query Language (SQL)– CREATE, INSERT, UPDATE, DELETE, SELECT Statements. 3.22 Set Operations, ORDER BY, HAVING, GROUP BY Clauses, Aggregate Functions. Course Outcome: CO3 Teaching Hours :8 hrs
4.	**TITLE:** Database Normalization 4.19 Redundancy and Anomalies 4.20 Functional dependencies 4.21 Lossless Decomposition 4.22 1st, 2nd and 3rd Normal Course Outcome: CO4 Teaching Hours :8 hrs
5.	**TITLE:** Security and Integrity 5.21 Database security 5.22 Previlages 5.23 Data integrity 5.24 Types of integrity constraints Course Outcome: CO5 Teaching Hours :8 hrs

REFERENCE BOOKS:

S. N.	Title	Author, Publisher, Edition and Year of publication	ISBN
1.	Data base System Concepts	Silberschatz, A., Korth, H. F., Sudarshan, S., McGraw hill, 7e,	978-0078022159
2.	Introduction to Database Management System	ISRD Group, Tata McGraw-Hill, 2013	978-0070591196

CO VS PO MAPPING:

CO/PO-PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	2	1	0	1	0	2	3	1	0
CO2	3	2	3	2	0	0	2	3	2	2
CO3	3	3	2	2	0	0	2	3	3	3
CO4	3	3	2	2	1	0	2	3	2	2
CO5	3	2	1	2	2	0	2	3	1	1

SIGNATURES: