(PGDCA)
Post Graduate Diploma in Computer Application

(PGDCA)
Post Graduate Diploma in Computer Application

Detailed Syllabus

Programme Code: PGDCA
Duration: 1 Years

EFFECTIVE FROM SESSION: 2022-2023

Department of Computer Sciences
Faculty of Engineering
CHHATRAPATI SHIVAJI MAHARAJ UNIVERSITY PANVEL, NAVI MUMBAI

SEMESTER I

PGDCA 1 Year - Semester 1										
Course Code	**Course Title**	**Hours/ Week**			**Theory Marks**		**Practical Marks**		**Total Marks**	**Credit**
		L	**T**	**P**	**IA**	**ESE**	**IA**	**ESE**		
PGDCA1110	Fundamental of Information Technology	3	1	-	30	70	-	-	100	4
PGDCA1120	C Programming	3	1	-	30	70	-	-	100	4
PGDCA1130	Discrete Mathematics	3	1	-	30	70	-	-	100	4
PGDCA1140	Database Management System	3	1	-	30	70	-	-	100	4
PGDCA1150	Professional English	3	1	-	30	70	-	-	100	4
PGDCA1121	C Programming Lab	-	-	4	-	-	15	35	50	2
PGDCA1141	DBMS Lab	-	-	4	-	-	15	35	50	2
	TOTAL	**15**	**3**	**8**	**150**	**350**	**30**	**70**	**600**	**24**

Fundamentals of Information Technology

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**30 Marks**
Total Credits	**4**	**End-Semester Examination**	**70 Marks**

Course Objectives

The objectives of this course are

1. To provide the foundational concepts of computer hardware, software, operating systems, peripherals, etc. along with how to get the most value and impact from computer technology.
2. To make knowledge for beginners as well as advanced learners who want to deal with computers.
3. To make students at a moderate level of expertise in the knowledge of computer basics from where a student can take himself to the next level.
4. To provide detailed information about Database & Packages.
5. To differentiate between various file system format.

Detailed Syllabus:

Unit I

Introduction of computer, Characteristics of Computers, Block diagram of computer, Block diagram of computer, Types of computer and features, Types of programming languages, Data Organization, Drives, Files, Directories, Types of Memory, Secondary storage devices, I/O devices, Number Systems.

UNIT II

Algorithm and Flowcharts

Algorithm - Definition of Algorithm, Characteristics of Algorithm, Advantages and disadvantages, Examples. Flowchart- Definition, Define symbols of flowchart, Advantages and disadvantages, Examples.

UNIT III

Windows Operating Environment

Features of MS –Windows, Control Panel, Taskbar, Desktop, Windows Application, Icons, Windows Accessories, Notepad, Paintbrush. Editors and Word Processors: Basic Concepts, MS-Word, Introduction to desktop publishing.

UNIT IV

Spread sheets and Database packages

Purpose, usage, commands, MS-Excel, Creation of files in MS-Access, Switching between application, MS -PowerPoint.

UNIT V

DOS and Linux

DOS: Introduction to Operating System, DOS - History Files and Directories, Internal and External Commands, Batch Files, Types of O.S.

Linux: File system, Linux Commands, Permission, I/O redirection, VI Editor.

Text /Reference Books:

1. Fundamentals of computers - V. Rajaraman - Prentice- Hall of India
2. Microsoft Office 2007 Bible - John Walkenbach,Herb Tyson,Faithe Wempen,cary N.Prague, Michael R.groh, Peter G.Aitken, and Lisa a.Bucki -Wiley India pvt.ltd.
3. The complete reference Linux - Richard Retersen - Tata McGraw - Hill Edition
4. A Conceptual Guide to OpenOffice.org 3 - R. Gabriel Gurley- CreateSpace Independent Publishing Platform, 2008
5. Introduction to Information Technology - Alexis Leon, Mathews Leon, and Leena Leon, Vijay Nicole Imprints Pvt. Ltd., 2013.
6. Computer Fundamentals - P. K. Sinha Publisher: BPB Publications.

Course Outcomes:

After completion of this subject students will be able to

CO1: Recognize and explain fundamental concepts and principles pertaining to computers
CO2: Recall and comprehend basic DOS commands, demonstrating knowledge of their functions and syntax, and grasp fundamental concepts of flowcharts and algorithms
CO3: Students become familiar with the Windows operating environment with additional knowledge of word processors.
CO4: Understand the detailed information about Basic database applications & Apply spreadsheets knowledge which will be used in daily work life.
CO5: Understand the concept of Linux operating system.

C Programming

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**30 Marks**
Total Credits	**4**	**End-Semester Examination**	**70 Marks**

Course Objectives:

1. The subject aims to provide exposure to problem-solving through programming.
2. It aims to train the student to the basic concepts of the C-programming language.
3. This subject involves a lab component which is designed to give the student hands-on experience with the concepts.
4. The student aims to learn various advanced concept such as arrays & pointers in C programming.
5. The students differentiate between various functions & it's used.

Detailed Syllabus:

UNIT I

Introduction to 'C' Language

History, Structures of 'C' Programming, Function as building blocks, Language Fundamentals: Character set, C Tokens, Keywords, Identifiers, Variables, Constant, Data Types, Comments, Operators, Built-in Operators and function, Console based I/O and related built-in I/O function, Concept of header files, Preprocessor directives.

UNIT II

Control structures

Decision making structures- If, If-else, Nested If -else, Switch. Loop Control structures- While, Do-while, for, nested for loop, Other statements- break, continue, goto, exit.

UNIT III

Arrays and Pointers

Arrays: Definition, declaration and initialization of one dimensional array, accessing array elements, displaying array elements, Two-Dimensional array-declaration and initialization, accessing and displaying, memory representation of array-row major, Column major, Multidimensional array.

Pointers: Definition and declaration, Initialization, indirection operator, address of operator, pointer arithmetic, dynamic memory allocation, arrays and pointers, function and pointers

UNIT IV

Structures and Functions

Structures: Definition and declaration, Variables initialization, accessing fields and structure operations, Nested structures, Union-Definition and declaration, Differentiate between Union and structure **Functions:** Basic types of function, Declaration and definition, Function call, Types of function, Parameter passing- Call by value, Call by reference, Scope of variables, Storage classes, Recursion.

UNIT V

Strings and File Handling

Strings: Definition, declaration and initialization of strings, standard library functions: -strlen (), strcpy( ), strcat( ), strcmp( ), Implementation without using standard library functions.

File handling: Definition of Files, Opening modes of files, Standard function: -fopen( ), fclose( ), feof( ), fseek( ), rewind( ),Using text files:- fgetc( ), fputc( ), fprintf( ), fscanf**( ).**

Text /Reference Books:

1. Let us C-Yashwant Kanetkar.
2. Programming in C- Balguruswamy
3. The C Programming Lang., Pearson Ecl – Dennis Ritchie
4. Structured programming approach using C-Forouzah &Ceilberg Thomson learning publication.
5. Pointers in C – Yashwant Kanetkar
6. How to solve it by Computer – R. G. Dromy
7. Introduction to algorithms – Cormen, Leiserson, Rivest, Stein

Course Outcomes:

After completion of this subject students will be able to

CO1: Comprehend the foundational concepts of C programming language.
CO2: Understand & implement some information about control structures which is to be used further in practical's.
CO3 Demonstrate the use of functions in C programming.
CO4: Understand & implement information about Arrays & its application in C programming.
CO5: Describe the detailed information about file handling in C programming and Develop basic applications.

Discrete Mathematics

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**30 Marks**
Total Credits	**4**	**End-Semester Examination**	**70 Marks**

Course Objective:

1. Introduce concepts of mathematical logic for analyzing propositions and proving theorems.
2. Use sets for solving applied problems, and use the properties of set operations algebraically.
3. Work with relations and investigate their properties.
4. Investigate functions as relations and their properties.
5. Introduce basic concepts of graphs, digraphs and trees.

Detailed Syllabus:

Unit-I

Number System: Decimal Number Systems, Binary Number System, Hexadecimal Number Systems Octal Number Systems, Binary Arithmetic

Unit-II

Propositions and Logical Operations Notation, Connections, Normal forms, Truth Tables Equivalence and Implications Theory of inference for statement calculus, Predicate calculus Rules of Logic, Mathematical Induction and Quantifiers

Unit-III

Sets, Relations and Diagraphs, Review of set concepts, Relations and digraphs, Properties of relations, Equivalence relations, Computer representation of relations and digraphs, Manipulation of relations, Partially Ordered sets (Posets)

Unit-IV

Recurrence Relations, Towers of Hanoi, Iterations, Homogeneous linear equations, with constant coefficients, particular solution, difference table, finite order differences, Line in a plane in general position

Unit-V

Groups and Applications Monoids, semi groups, Product and quotients of algebraic structures, Isomorphism, homomorphism, auto morphism, Normal subgroups, Codes and group codes

Unit-VI

Classification of Languages Overview of Formal Languages: Representation of regular languages and grammars, finite state machines

Reference:

1. “Discrete Mathematical Structures”: Tremblay and Manohar, Tata McGraw Hill
2. “Discrete Mathematics”: 1st edition by Maggard, Thomson
3. “Discrete Mathematics”: Semyour Lipschutz, Varsha Patil IInd Edition Schaum’s Series TMH
4. “Discrete Mathematical Structures”: Kolman, Busby and Ross, Prentice Hall India, Edition 3

Course Outcomes:

At the end of the course student will be able to

CO1: Understand and construct the notion of mathematical thinking, mathematical proofs and to implement them in problem solving.
CO2: Ability to understand relations, Digraph and lattice.
CO3: Ability to understand use of functions, graphs and their use in programming applications.
CO4: Understand use of groups and codes in Encoding-Decoding.
CO5: Discuss the concept of formal language,Analyze the formal language & non formal language.

Database Management System

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**30 Marks**
Total Credits	**4**	**End-Semester Examination**	**70 Marks**

Course Objectives:

The objectives of this course are

1. To make the students learn and practice data modeling using the entity-relationship and developing database designs.
2. To make students understand the use of Structured Query Language (SQL) and learn SQL syntax.
3. To illustrate students how to apply normalization techniques to normalize the database
4. To make students understand the needs of database processing and learn techniques for controlling the consequences of concurrent data access.

Detailed Syllabus:

Introduction to Database Concepts:

UNIT 1

Introduction, Characteristics of databases, File system v/s Database system, Users of Database system, Data Independence, DBMS system architecture, Database Administrator

UNIT 2

Entity–Relationship Data Model

The Entity-Relationship (ER) Model: Entity types : Weak and strong entity sets, Entity sets, Types of Attributes, Keys, Relationship constraints : Cardinality and Participation, Extended Entity Relationship (EER) Model: Generalization, Specialization and Aggregation

UNIT 3

Relational Model and Relational Algebra

Introduction to the Relational Model, relational schema and concept of keys, Mapping the ER and EER Model to the Relational Model

Relational Algebra – unary and set operations, Relational Algebra Queries.

UNIT 4

Structured Query Language (SQL)

Overview of SQL, Data Definition Commands, Data Manipulation commands, Data Control commands, Transaction Control Commands.

Set and string operations, aggregate function - group by, having,.

Views in SQL, joins , Nested and complex queries, Integrity constraints :- key constraints, Domain Constraints, Referential integrity , check constraints

Triggers

UNIT 5

Relational–Database Design and Transaction Management

Pitfalls in Relational-Database designs , Concept of normalization, Function Dependencies , First Normal Form, 2nd , 3rd , BCNF, multi valued dependencies , 4NF.

Transaction concept, Transaction states, ACID properties, Concurrent Executions, Serializability – Conflict and View.

Text /Reference Books:

1. G. K. Gupta ―Database Management Systems‖, McGraw – Hill.
2. Korth, Slberchatz, Sudarshan, ―Database System Concepts‖, 6th Edition, McGraw – Hill
3. Elmasri and Navathe, ―Fundamentals of Database Systems‖, 5th Edition, Pearson education.
4. Peter Rob and Carlos Coronel, ―Database Systems Design, Implementation and Management‖, Thomson Learning, 5th Edition.

Course Outcomes:

At the end of this course students will able to:

CO1: Describe the fundamentals of a database systems
CO2: Use and Design ER diagrams for real life problems.
CO3: Describe the conceptual model to relational model and Design relational algebra queries.
CO4: Create and Analyze databases using SQL.
CO5: Analyze and use the concepts of normalization to relational database design.

Professional English

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**30 Marks**
Total Credits	**4**	**End-Semester Examination**	**70 Marks**

Course Objectives:

1. To acquaint the students with appropriate language skills with the purpose of improving the existing ones-LSRW.
2. To make the learners understand the importance and effective use of non-verbal communication.
1. To make the learner proficient in public speaking and presentation skills.
2. To guide and teach the students to utilize the principles of professional business and technical writing for effective communication in the global world.

Detailed Syllabus:

UNIT I

Communication and Communication Process

Introduction to Communication, Forms and functions of Communication, Barriers to Communication ((linguistic and semantic, psychological, physical, mechanical, cultural), and overcoming them, Types of communication: verbal and non-verbal communication. Reading: Introduction to Reading, Barriers to Reading, Types of Reading: Skimming, Scanning, Fast Reading, Strategies for Reading, Comprehension. Listening: Importance of Listening, Types of Listening, and Barriers to Listening.

UNIT II

Writing Skills, Reading Skills & Listening Skills

Features of Good Language, Technical Style of writing, Writing Emails and it's etiquettes, Technical Reports: Report Writing: Types, Format and Structure of reports.

UNIT III

Letter Writing

Types of letters: Job application letter, complaint letter, enquiry letter, reply to enquiry, sales letter, Essential and non-essential parts of letters, formats of letters.

UNIT IV

Grammar

Types of sentences, Antonyms and Synonyms, Use of Auxiliaries and Modal Auxiliaries, Synonyms and Antonyms, Pairs of confused words, Common Errors in sentences

UNIT V

Soft Skills

Body language, Teamwork and skills, Decision making ability, Negotiation skills and Interview skills.

UNIT VI

Dialogues Writing and Speaking

Greeting someone and responding to greet, Thanking someone and responding to thanks, Making inquiry and responding to enquiry on telephone, Making request and responding to request.

Text Books and References:

1. Communication in Organizations by Dalmar Fisher, Jaico Publishing House
2. Communication Skills by Meenakshi Raman & Sangeeta Sharma,
3. Oxford University Press
4. Business Correspondence & Report-writing by R.C. Sharma& Krishna Mohan, Tata McGraw-Hill Education
5. Effective Technical Communication by Ashraf Rizvi, Tata McGraw-Hill.
6. Technical Writing & Professional Communication for non-native speakers of English by Thomas N. Huckin & Leslie A. Olsen, McGraw –Hill.
7. Mastering Communication by Nicky Stanton, Palgrave Master Series

Course Outcomes:

After completion of this subject students will be able to

CO1: Understand and evaluate information they listen to and express their ideas with greater clarity.
CO2: Understand the various channels of communication in a business organization.
CO3: Remember the expanded vocabulary and Speak convincingly to an audience
CO4: Analyze the digital content.
CO5: Evaluate through result oriented writing both within and outside the organization.
CO6: Create & Apply effective and easy to understand technical description, instructions.

C Programming Lab

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**15 Marks**
Total Credits	**2**	**End-Semester Examination**	**35 Marks**

Suggested list of experiments:

1. Write a C program to display "This is my first C Program.
2. Write a C program to add two numbers and display its sum.
3. Write a C program to multiply two numbers (4 and 5) and display its product.
4. Write a C program to calculate area and circumference of a circle.
5. Write a C program to perform addition, subtraction, division and multiplication of two numbers.
   Write a Program to Check Whether a Number is Prime or not.
6. Write a program to find the largest and smallest among three entered numbers and also display whether the identified largest/smallest number is even or odd.
7. Write a program to check whether the entered year is a leap year or not (a year is leap if it is divisible by 4 and divisible by 100 or 400.)
8. Write a program to find the factorial of a number.
9. Write a program to check if the number is Armstrong or not. (Hint: A number is Armstrong if the sum of cubes of individual digits of a number is equal to the number itself).
10. Write a program to print day names using switch cases.
11. Write a program to determine whether the input character is capital or small letter, digits or special symbol.
12. Write a program to reverse a given integer.
13. Write a program to print numbers in reverse order with a difference of 2.
14. Write a program to print the sum of digits of a number using a for loop.
15. Write a program to check whether a number is Palindrome or not.
16. Write a program to generate Fibonacci series.
17. Write a Program to Search an element in an array.
18. Write a Program to find the largest and smallest element in Array.
19. Write a program to add, subtract, multiply and divide two integers using user defined type function with return type.
20. Write a program to calculate the sum of the first 20 natural numbers using a recursive function.
21. Write a program to swap values of two variables using a pointer.
22. Write a program to add two numbers using pointers.

23. C Program to list all files and sub-directories in a directory.
24. C Program to count the number of lines in a file.
25. C Program to print contents of file.
26. C Program to copy contents of one file to another file.

Database Management System Lab

Teaching Scheme		**Evaluation Scheme**	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**15 Marks**
Total Credits	**2**	**End-Semester Examination**	**35 Marks**

Suggested list of experiments:

1. Creating a Table with constraints.
2. Writing SQL statements Using ORACLE
    1. Writing basic SQL SELECT statements.
    2. Restricting and sorting data.
    3. Displaying data from multiple tables.
    4. Aggregating data using group functions.
    5. Creating and managing tables.
3. Normalization in ORACLE.
4. Creating a cursor in oracle.
5. Creating procedure and functions in oracle.
6. Creating packages and triggers in oracle.
7. Design a Database and create required tables. For e.g. Bank, College Database
8. Apply the constraints like Primary Key , Foreign key, NOT NULL to the tables.
9. Write a sql statement for implementing ALTER,UPDATE and DELETE
10. Write the queries to implement the joins
11. Write the query for implementing the following functions: MAX(),MIN(),AVG(),COUNT()
12. Write the query to implement the concept of Integrity constrains
13. Write the query to create the views
14. Perform the queries for triggers
15. Perform the following operation for demonstrating the insertion , updation and deletion using the referential integrity constraints
16. Write the query for creating the users and their role.

SEMESTER II

PGDCA 1 Year - Semester 2										
Course Code	**Course Title**	**Hours/ Week**			**Theory Marks**		**Practical Marks**		**Total Marks**	**Credit**
		L	**T**	**P**	**IA**	**ESE**	**IA**	**ESE**		
PGDCA2010	Object Oriented Programming (C++)	3	1		30	70			100	4
PGDCA2020	Software Engineering	3	1		30	70			100	4
PGDCA2030	Data Communication & Networking	3	1		30	70			100	4
PGDCA2040	Dot Net Technology using C#	3	1		30	70			100	4
PGDCA2011	Object Oriented Programming (C++) Lab			4			15	35	50	2
PGDCA2041	Dot Net Technology using C# Lab			4			15	35	50	2
PGDCA2051	Project			4			40	60	100	2
	TOTAL	**15**	**3**	**8**	**150**	**350**	**70**	**130**	**600**	**22**

OBJECT ORIENTED PROGRAMMING WITH C++

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**30 Marks**
Total Credits	**4**	**End-Semester Examination**	**70 Marks**

Course Objectives:

The objectives of this course are

1. To make students understand Object Oriented Programming concepts using the C++ language.
2. To introduce the principles of data abstraction, inheritance to the learner.
3. To introduce the principles of virtual functions and polymorphism.
4. To illustrate students how to handle formatted I/O and unformatted I/O.
5. To make the students familiarize with the concept of exception handling.

Detailed syllabus:

UNIT I

Object-Oriented Thinking: Different paradigms for problem solving, need for OOP paradigm, differences between OOP and Procedure oriented programming, Overview of OOP concepts-Abstraction, Encapsulation, Inheritance and Polymorphism.

UNIT II

C++ Basics: Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References. Flow control statement- if, switch, while, for, do, break, continue, goto statements. Functions - Scope of variables, Parameter passing, Default arguments, inline functions, Recursive functions, Pointers to functions. Dynamic memory allocation and de-allocation operators-new and delete, Preprocessor directives.

UNIT III

C++ Classes and Data Abstraction: Class definition, Class structure, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Dynamic creation and destruction of objects, Data abstraction, ADT and information hiding.

UNIT IV

Inheritance: Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtual base class.

UNIT V

Virtual Functions and Polymorphism: Static and Dynamic binding, virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, Implications of polymorphic use of classes, Virtual destructors.

References:

1. Object Oriented Programming in C++ -Robert Lafore, edition, Galgotia publications
2. The Complete Reference C++, Herbert Schildt, 4th Edition, TMH
3. Y. Kanetkar, 'Let's C++', BPB publications
4. E Balagurusamy, 'Object oriented programming with C++', 4th edition, TMH
5. Sourav Sahay , 'Object-Oriented Programming with C++' , Oxford University Press, 2006.

Course Outcomes:

At the end of this course students will able to

CO1: Differentiate between Procedural & Object oriented programming language.
CO2: Understand Object Oriented Programming concepts using the C++ language
CO3: Understand & implement the principles of data abstraction, inheritance
CO4: Implement the principles of virtual functions and polymorphism.
CO5: Examine the handling formatted I/O and unformulated.

Software Engineering

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**30 Marks**
Total Credits	**4**	**End-Semester Examination**	**70 Marks**

Course Objective:

The objectives of this course are

1. To make the students familiarize with system analysis and design is the backbone of Application software development.
2. To illustrate students the steps in analysis and design of the system.
3. It includes the knowledge of preparing projects systematically.
4. It is important to know about various aspects of the system analysis and design so that the students will be able to understand the responsibilities while designing and implementing the project.

Detailed Syllabus:

UNIT I

Introduction to Software Engineering, System Concepts: Types of systems : (open, closed, static and dynamic systems). Introduction, Programmes v/s Software Products Emergence of Software Engineering- Early Computer Programming, High-level Language Programming, Control flow based Design, Data Structure Oriented Design, Object Oriented Design

UNIT II

Software Life Cycle Models, Requirement of Life Cycle Model, Classic Waterfall Model, Prototyping Model, Evolutionary Model, Spiral Model, introduction to agile methodology. Comparison of different Life Cycle Models

UNIT III

Software Planning, Responsibilities of Software Project Manager - Metrics for Project Size Estimation- LOC(Lines of Code), Function Point Metric - Project estimation Techniques- Using COCOMO Model.

UNIT IV

Requirement Analysis and Specification, Requirement gathering and Analysis, Software Requirement Specifications(SRS), Characteristics of good SRS

UNIT V

Software Design and Implementation, Characteristics and features of good Software Design Cohesion and Coupling, Software design Approaches- Function Oriented Design (Data flow diagrams, Data dictionary, Decision Trees and tables), Object Oriented Design, Structured Coding Techniques, Coding Styles, and documentation.

UNIT VI

Software Testing, Concept of Testing, Testing type cycle (V-Model), Verification v/s Validations, Unit Testing, Black Box Testing, White Box Testing, Integration testing, System testing, Configuration management, Overview of test cases.

Text Books:

1. Software Engineering by Rajib Mall, PHI Publishers, New Delhi
2. An Integrated Approach to Software Engineering by Pankaj Jalote, Narosa Publishing House Pvt Ltd, Darya Ganj, New Delhi 110002
3. Software Engineering, Sangeeta Sabharwal, New Age International, Delhi
4. Software Engineering by KK Aggarwal and Yogesh Singh
5. Software Engineering – A Practitioner's Approach by RS Pressman, Tata McGraw Hill Publishers, New Delhi

Course Outcomes:

At the end of this course students will able to:

CO1: Identify the problem and corresponding requirement for development of software.
CO2: Discuss the various phases of the system development life cycle.
CO3: Understand the expected benefits and scope of the projects.
CO4: Design data flow diagrams and decision tables.
CO5: Design a feasibility study of the system & Evaluate the feasibility study result.
CO6: Implement different testing techniques on simple programs.

Data Communication & Networking

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**30 Marks**
Total Credits	**4**	**End-Semester Examination**	**70 Marks**

Course Objectives:

The objectives of this course are

1. Describe communication protocols and layered network architectures.
2. Explain convention computer system interfacing standards and peer to peer data link communication protocols
3. Design basis network systems.
4. Analyze data communication technology.

Detailed Syllabus:

Unit- I

Introduction to Networking

Introduction to computer network, network application, network software and hardware components (Interconnection networking devices), Network topology, protocol hierarchies, design issues for the layers, connection oriented and connectionless services, Reference models: Layer details of OSI, TCP/IP models. Communication between layer.

Unit- II

Physical Layer

Electromagnetic Spectrum, Guided Transmission Media: Twisted pair, Coaxial, Fiber optics. Unguided, media (Wireless Transmission): Radio Waves, Microwave.

Unit- III

Data Link Layer

DLL Design Issues (Services, Framing, Error Control, Flow Control), Error Detection and Correction(Hamming Code, CRC, Checksum) , Elementary Data Link protocols , Stop and Wait, Sliding Window(Go Back N, Selective Repeat). Access Control sublayer, Channel Allocation problem, Multiple ,access Protocol(Aloha, Carrier SenseMultiple Access (CSMA/CD),(CSMA/CA).

Unit- IV

Network Layer

Network Layer design issues, Communication Primitives: Unicast, Multicast, Broadcast. IPv4 Addressing (classfull and classless), Subnetting, Supernetting design problems, IPv4 Protocol, Network Address Translation (NAT),Routing algorithms : Shortest Path(Dijkastra‘s), Distance Vector Routing, Protocols - ARP,RARP,Token & Leaky bucket algorithms.

Unit- V

Transport & Application Layer

Connection management (Handshake), UDP, TCP,DNS, HTTP,SMTP, Telnet, FTP

References:

1. Forouzen, "Data Communication and Networking", TMH
2. A.S. Tanenbaum, Computer Networks, Pearson Education
3. W. Stallings, Data and Computer Communication, Macmillan Press
4. Bhavneet Sidhu, An Integrated approach to Computer Networks, Khanna Publishing House
5. Gary R.Wright,W.Richard Stevens "TCP/IP Illustrated,Volume2 The Implementation" Addison-Wesley
6. Michael A. Gallo and William M. Hancock "Computer Communication and Networking Technology" Cengage Learning
7. Anuranjan Misra, “Computer Networks”, Acme Learning
8. G. Shanmugarathinam, ”Essential of TCP/ IP”, Firewall Media.

Course Outcomes:

At the end of this course students will able to

CO1: Explain the concept of Signals, OSI & TCP/IP reference models and discuss the functionalities of each layer in these models.
CO2: Discuss and Analyze flow control and error control mechanisms and apply them using standard data link layer protocols
CO3: Design subnets and calculate the IP addresses to fulfil network requirements of an organization
CO4: Differentiate and implement various routing algorithms to find shortest paths for packet delivery.
CO5: Analyze the features and operations of various application layer protocols such as HTTP, DNS and SMTP.

Dot Net Technology using C#

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**15 Marks**
Total Credits	**2**	**End-Semester Examination**	**35 Marks**

Pre-requisite of course: Object oriented concepts, Programming fundamentals

Course Objective:

1. Students will learn the data types, variables, control structures, classes and objects, inheritance, interfaces, and exceptions.

2. Students will learn how to write efficient and effective code in C# and gain a strong foundation for further study in computer science and programming.

3. Students will learn the basics of software development, including object-oriented programming concepts, event-driven programming, and working with databases.

4. students will have a comprehensive understanding of the C# programming language and be able to write their own programs using the .net framework

Unit I Introduction To .Net Architecture

Introduction to .NET Framework Architecture, Program Execution in .NET, CLR structure, Assemblies, Creating strong named assemblies, putting dll in GAC ,Garbage Collection, DLL Hell, Side by Side Execution, Debugging.

Unit II Basics of C#

Basic datatypes, declaring variables and constants Type Conversion, Boxing and Unboxing, Array, Structure, String Manipulation, String Builder, Decision making statements, Conditional Loops, Switch Case

Unit III Basics of C#

Creating Class, declaring variables and methods, Access Modifiers, Constructors, Abstract Class, Partial Class, Inheritance, method overloading, method overriding, Anonymous method, Properties, Indexers, Exception Handling

Unit IV Building GUI with C#

Working with C# windows applications, Working with common form controls. Visual Inheritance, Creating MDI Form, Event Handling

Unit V Playing with data using ADO.net

Overview of ADO.Net framework, working with SQL server database, Managed Provider, Dataset, working with data source, Connected and disconnected architecture, Binding data with Data grid, Binding data with Crystal Report

References:

1. Pro C# 7 With .NET and .NET Core Apress Edition, Troelsen, Andrew, Japikse, Philip
2. Professional C# .Net Christian Nagel, Wrox Publication
3. C# The Basics Vijay Mukhi, BPB Publications
4. PRO XAML with C# Application Development Strategies (covers WPF, Windows 8.1, and Windows Phone 8.1) byJames, Buddy, Lalonde, Lori Apress Edition.
5. The Complete Reference C# by Herbert ShildtMcgrow Hill Publication

Course Outcomes: After completion of this course, student will be able

CO1: Explain the architecture of the .NET framework.
CO2: To design Console based C# application using Dot-Net framework.
CO3: To design GUI based desktop application using C# Win-form
CO4: Describe & implement the concept of ADO .net technology
CO5: To Design and develop basic applications using WPF.

Object Oriented Programming C++ Lab

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**15 Marks**
Total Credits	**2**	**End-Semester Examination**	**35 Marks**

S.No.	Practical Name
1	Write programs that illustrates the simple C++ concepts without classes.
2	Write programs that illustrates the Control Structures.
3	Write C++ programs to illustrate the concepts pointers, functions and function overloading.
4	Write C++ programs to illustrate the concepts: classes, friend functions.
5	Write C++ programs to illustrate the concepts: constructors, constructor overloading and destructors
6	Write C++ programs to illustrate the concepts: operator overloading (both Unary and Binary).
7	Write C++ programs to illustrate the concepts: Simple, Multiple, Multilevel inheritance
8	Write C++ programs to illustrate the concepts: 45 Lectures Polymorphism (Virtual functions, Pure Virtual functions).
9	Write programs to illustrate the file handling in C++
10	Write programs to illustrate the exceptions in C++

Dot Net Technology using C# Lab

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**15 Marks**
Total Credits	**2**	**End-Semester Examination**	**35 Marks**

1) Write a C# console program to know whether a number is even or odd.
2) Write a C# console program that implements basic calculation functions (Addition, Subtraction, Multiplication, Division) using switch case.
3) Write a C# console program to implement following pattern
*
* *
* * *
* * * *
4) Create a console application based on the assigned definition to implement basic OOP concepts like abstract class, inheritance, interface ,constructors ,properties ,method overloading and method overriding.
5) Create a console application to implement advance concept like Reflection, Delegates, Attributes, Indexer.
6) Create a console application to demonstrate multithreaded application.
7) Create a console application that implements all Collection classes.
8) Create a console application that reads a content from user and store it in file using FILEIO.
9) Create a console application that list all the files in given folder which are having read only permission.
10) Create a windows application to implement fully functional notepad.
11) Create a windows application that implements MDI form and dialog boxes.
12) Implement the tutorial4 definition in windows form using ADO.net, Grid view control, and Crystal report

Project

Teaching Scheme		Evaluation Scheme	
Lectures	**4 Hrs/Week**	**Internal Assessment Test**	**40 Marks**
Total Credits	**2**	**End-Semester Examination**	**60 Marks**

Course Objectives:

1. To prepare students to excel in computer applications to succeed in industry/ technical profession.
2. To provide students with a solid foundation in mathematical and computing fundamentals and techniques required to solve related problems and also to pursue higher studies and research.
3. To train students with good computing breadth so as to comprehend, analyze, design and create computing solutions for the real life problems.

Course Contents

A project is an inquiry, conducted personally by a trainee(s) who is responsible for using a variety of methods (e.g. analysis, interpretation, planning etc) to undertake a task or study a subject (knowledge or skill or attitude) and to write a report, or solve a problem etc., in line with the objectives of the project. A project can also be termed as an open-ended assignment, the outcome of which is not known at inception and whose progress depends mostly on the intelligence, skills, creativity and energy of the students. The project work exposes the students to real life problems and introduces them to the procedures and practices used in industry. The project work also helps the students to gain confidence in tackling problems of their own. The project work is needed to strengthen and supplement the learning experience of students. Project-based instruction is an authentic instructional model or strategy in which students plan, implement, and evaluate projects that have real-world applications beyond the classroom. Learning activities that are interdisciplinary, long term, and student centred are emphasized, rather than short, isolated lessons. Most important, students find projects fun, motivating, and challenging because they play an active role in choosing the project and in the entire planning process. Teachers are increasingly working with students who have a wide range of abilities, come from various backgrounds. Institutes are seeking ways to respond to the needs of these students. Project-based instruction provides one way to introduce a wider range of learning opportunities into the classroom. It can engage students from diverse backgrounds because students can choose topics that are related to their own experiences, as well as allow them to use individual learning styles.

There are a wide range of project types such as service learning projects, work-based projects, task-oriented projects, problem-solving projects and so forth, but authentic projects all have in common these defining features:

1) Student centred, student directed
2) A definite beginning, middle, and end
3) Content meaningful to students; directly observable in their environment
4) Real-world problems
5) First hand investigation
6) Sensitivity to local culture and culturally appropriate
7) Specific goals related to curriculum and institute, district, or state standards
8) A tangible product that can be shared with the intended audience
9) Connections among academic, life, and work skills
10) Opportunity for feedback and assessments from expert sources 2
11) Opportunity for reflective thinking and student self-assessment
12) Authentic assessments (portfolios, journals, etc.)

Course Outcomes: After completion of this course, student will be able

CO1: Discover potential research areas in the field of IT.
CO2: Conduct a survey of several available literatures in the preferred field of study
CO3: Compare and contrast the several existing solutions for research challenges.
CO4: Demonstrate an ability to work in teams and manage the conduct.
CO5:Formulate and propose a plan for creating a solution for the research plan identified.
CO6: Report and present the findings of the study conducted in the preferred domain.