**UBIT (Karachi University)**

**BSCS-402 : Data Structure Credit Hour (2+1)**

Introduction, Basic Terminology, Elementary Data Organization, Data Structure, Data Structure Operations, Algorithms, Preliminaries, String Processing, Arrays, Records and Pointers, Linked Lists, Stacks, Queues, Recursion, Trees, Graphs and their Applications, Sorting and Searching.

Recommended Books :

1. Mark Allen Weiss (1993). Data Structures and Algorithm Analysis in C. The Penjamin/Cummings Publishing Company, Inc. U.K.  
2. Mark Allen Weiss (1992). Data Structures and Algorithm Analysis. The Penjamin/Cummings Publishing Company, Inc. U.K.  
3. Lipschutz, S. (1986). Theory and Problems of Data Structures. Schaum’s Outline Series, McGraw Hill Book Company, New York.

**BSCS-408 : Object Oriented Programming Credit Hour (2+1)**

Introduction, What is Object-Oriented Programming? Object-Oriented Programming Applications, Stream I/O, Error Handling, Classes, Objects, Overloading, Constructors, Destructors, Derived Classes, Methods, Passing Objects, Sample Application.

Recommended Books :

1. Parsons, David (1995) Object-Oriented Programming with C++. BPB Publications, New Delhi  
2. Smith, Norman E. (1992). Object-Oriented Programming using Turbo C. BPB Publications, New Delhi.

**BSCS-509 : Database Systems Credit Hour (2+1)**

Introductory Database concepts. Design of Logical Models : Database Planning, Physical Data Organization, Data Architecture, The Entity-Relationship Model, The Relational Model, Normalization.

Design of the Physical Model : Relational Database Management System, The Network Model, The Hierarchical Model, Semantic Models. Fundamentals of Distributed Databases, The Object Oriented DBMS. Implementation Issues : Concurrency Control and Recovery, Security and Integrity, Query Optimization, Review of DBMS software and recent development and implementation exercises. KBMS (Knowledge-Based Management Systems). (Review of Oracle, Informix and Sybase with implementations).

Recommended Books :

1. Ricardo, C. (1990). Database Systems :  
2. Fortier, Paul J. (1997). Database Systems. McGraw Hill, New York.  
3. Date, C.J. (1994). An Introduction to Database Systems, Third Edition, Vol. 1. Narosa Publication House, New Delhi.  
4. D. Mason (1993). Introduction to Oracle and SQL. Chapman & Hall, London.  
5. D.S. Bowers (1992). From Data to Database. 2nd Edition. Chapman & Hall, Glasgow.

**BSCS-606 : Distributed Database Systems Credit Hour (2+1)**

Introduction to Distributed Systems : Processing functions, storage of databases, system control. Advanced topics in distributed data and knowledge base systems : architecture, database design; query processing and optimization, concurrency; recovery. Techniques for supporting heterogeneous database systems and cooperative information systems.

Recommended Books :

1. T. Tamer, Ozsu, P. Viladuriez (1991). Principles of Distributed Database Systems. Prentice-Hall, London.  
2. Ceri, S. Pelagatti, G. (1984). Distributed Databases : Principles and Systems. McGraw Hill, New York.  
3. Coulouris, G.F. and Dollimore, J.B. (1988). Distributed Systems : Concepts and Design. Reading, M.A., Addition Wesley.  
4. Mulleender, S. (Ed.) (1989). Distributed Systems Reading, M.A., Addison Wesley.  
5. Andrew S. Tenenbaum (1995). Distributed Operating System. Prentice Hall, London.

**Object Oriented Programming (Punjab University)**

|  |
| --- |
|  |
| Introduction to Object Oriented Programming.  Objects, Classes and inheritance.  The concept of Encapsulation, Polymorphism. Early and late binding. Operator overloading and function overloading.  Constructors and Destructors. Friend functions.  In-line functions.  Virtual functions.  Class libraries.  Object-oriented software development. |

**University of Peshawar:**

Course Code Course Title Credit Hours CS3204 **Object Oriented Programming** 4(3+1)

**Prerequisites:** Programming Fundamentals

**Course Description:** This course discusses the object oriented model in programming. Students taking this course would have already taken a course on programming. This will help build good quality software using object-oriented techniques.

**Aims and Objectives:** When students complete this course, they will be able to: • Understand the importance of OOP as compared to Structured Programming. • Understand the basic constructs of a C++ program. • Define a problem and implement its solution using classes and objects. • Effectively use and implement OOP analysis and design. • Understand object-oriented programming features in C++. • Understand object-oriented concepts and how they are supported by C++. • Gain some practical experience of C++.

**Course Contents:** Object Oriented Programming (OOP) and its Significance as a Modeling Technique. Comparison of Structured Programming and OOP, Classes and Objects in C++, Abstraction, New User Defined Data Types, Creating Objects from Classes, Accessing Member, Access Specifiers, Member Functions, Defining Member Functions, Constructors and Properties, Default Constructor, Constructor Overloading, Copy Constructor, Deep Copy, Shallow Copy, Destructors, “this” Pointer, Constant Member Function, Static Variables, Accessing Static Data Member, Static Member Function, Comparison of Global Variables and Static Variable, Arrays of Objects, Pointer to Objects Templates, Function Templates, Class Templates, Data Encapsulation and Abstraction, Importance of Data Encapsulation and Abstraction, Correctly Using the Access Modifiers, Friend Functions, Composition, Aggregation, Operator Overloading, Overloading Assignment Operator, Friend Function and Operator Overloading, Unary Operators Overloading, Inheritance and Importance, Inheritance in C++, Comparison of Overloading and Overriding, Hierarchy of Inheritance, Types of Inheritance, Private Inheritance, Protected Inheritance Multiple Inheritance Problem in Multiple Inheritance, Polymorphism and Importance, Virtual Functions, Static Binding, Dynamic Binding, Abstract Classes and Concrete Classes, Virtual Destructors, Virtual Functions and Pure Virtual Functions, Virtual Functions Usage, Dynamic Dispatch, Namespaces and Using Namespaces, Memory Management and Importance, Memory Areas(Heap, Stack), Use of new Operator, malloc() and calloc() Functions Calls.

**Recommended Books** 1. Deitel, P., Deitel, H. (2016). C++ How to Program (Latest ed.). Prentice Hall. 2. Laurence, P. (2017). C++: The Ultimate Crash Course to Learning the Basics of C++ (Latest ed.).Prentice Hall.

**Bibliography**

1. Lafore, R. (2005). Object-oriented programming in C (Latest ed.). Indianapolis, Ind: Sams.

2. Kanetkar, Y. (2004). Basic programming in C++ (Latest ed.). BPB Publications.

Course Code Course Title Credit Hours CS4301 **Data Structures and Algorithms** 4(3+1)

**Prerequisites:** Programming Fundamentals

**Course Description:** The purpose of this course is to provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. The main objective of the course is to teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter. This course is also about showing the correctness of algorithms and studying their computational complexities. This course offers the students a mixture of theoretical knowledge and practical experience.

**Aims and Objectives:** Students successfully completing this course should be able to: • Be familiar with basic techniques of algorithm analysis. • Be familiar with writing recursive methods. • Master the implementation of linked data structures such as linked lists and binary trees. • Be familiar with several sub-quadratic sorting algorithms including quicksort, mergesort and heapsort.

**Course Contents:** Introduction and Overview, Abstract Data Type, Arrays, Stacks(Push and Pop), Infix, Postfix and Prefix, Basic Operations, Queues(Insertion, Deletion, De-queues), Heap, Lists, Linked Lists, Searching(Binary and Sequential), Sorting, Sorting and Hashing, Recursion, Trees, Linked Lists Implementation, Binary Trees, B-Trees, Trees Traversal, Basic Operations, Traversals Sets, Graph, Representation of Directed and Undirected Graphs, Traversals, Minimum Cost Spanning Tree, Complexity(Space and Time).

**Recommended Books** 1. Miller, B. N., & Ranum, D. L. (2011). Problem Solving with Algorithms and Data Structures Using Python (2 nd Ed.). Franklin, Beedle & Associates Inc.

2. Lambert, K. (2014). Fundamentals of Python: Data Structures (Latest ed.). Nelson Education.

**Prerequisites:** **Data Structures and Algorithms**

**Course Description:** Establish proficiency in the fundamental concepts of database systems and the database system development life cycle of an enterprise database.. It discusses the theoretical foundations of database architecture, Relational Algebra and Relational calculus. This course also focuses on the principles, design and implementation of multi-user databases and its applications in client/server environment.

**Aims and Objectives:** When students successfully complete this course, they will be able to: • Understand intricacies involved in the design and implementation of a database system. • Have in-depth knowledge of entity-relationship (ER) modeling • Demonstrate competence with the fundamental tasks involved in gathering and analyzing user requirements, build conceptual, logical, and physical data models that reflect the organization’s requirements. • Apply normalization techniques. • Query a relational DBMS to create and maintain databases in a client server environment.

**Course Contents:** Introduction to database systems, The Relational Data Model, Relational Algebra and Relational Calculus, Database environment, Database Management System, Conceptual database modeling using the entity-relationship model and Enhanced EntityRelationship modeling, Schema quality through the study of functional dependencies and normalization, Logical and Physical Database Design, The SQL (DDL, DML, and DCL), Database application development, data integrity, constraints (entity integrity, referential integrity, domain integrity etc. ), Database Transaction Management.

**Recommended Books** 1. Connolly, T. M., & Begg, C. E. (2014). Database Systems: A Practical Approach To Design, Implementation And Management (6 th Ed.). Addison Wesley Publishing Company.

**Bibliography** 1. Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems (3 rd Ed.). WCB/McGraw Hill. 2. Elmasri, R., & Navathe, S.B. (2016). Fundamentals of Database Systems (Global Ed.). Pearson Education Limited. 3. Hoffer, J., Venkataraman, R., & Topi, H. (2015). Modern database management (Latest ed.). Prentice Hall Press. 4. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). Database system concepts (4 th Ed.). New York: McGraw-Hill.