DSEU

# Delhi Skill and Entrepreneurship University

# Bachelor of Computer Application (BCA)

Syllabus Document

**Effective from Academic Year 2022-23**

# Program Information

## Program Vision

To nurture youth with global competence with knowledge in Computer Applications capable of driving their career in Computation and Information Technology.

## Program Outcome

What the students will learn by the end of program

1. Basic Knowledge/Skills: An Individual with a Positive Attitude with the best Communication, Presentation, Leadership Skills, and Professional Competence.
2. Technical Knowledge/Skills: Problem-solving Capabilities with Logical Thinking.
3. Software Skill and Project Skills: Program Development and Problem Solving Skills with Knowledge of Programming Languages like C, C++, Python, Database Management, Computer Networking, etc.
4. Personality Traits and Ethics.
5. Soft Skills.

## Pedagogy and Teaching Methodology

Three years of classroom training are interspersed with industry visits, guest lectures and paid apprenticeships. You will learn to gather order information, perform historical pattern analysis, follow up with trucking companies, communicate with clients, consolidate orders by destination, and monitor all consignments in real-time. Students will be exposed to the industry interface via internships, live projects, field placements, master classes by the industry personnel, and classroom online/ offline workshops and seminars. In the third year, students will be offered an apprenticeship in the industry under the guidance of industry experts and in-house faculty members. In the last semester, students will be able to consolidate the knowledge of all the five semesters in the apprenticeship, in order to find the best career path for themselves.

## Credit scheme

| | Semester II | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Sl No. | Course Code | Course Name | Hours/week | | | | | Total Credits |
| | | | L | T | O | P | Total | |
| 1 | BCA-DC201 | Discrete Mathematics | 3 | 1 | 0 | 0 | 4 | 4 |
| 2 | BCA-DC202 | Operating Systems | 3 | 1 | 0 | 0 | 4 | 4 |
| 3 | BCA-DC203 | Free & Open Source Software (Linux Programming & Administration) | 2 | 1 | 0 | 2 | 5 | 4 |
| 4 | BCA-DC204 | Data & File Structure using C | 2 | 1 | 0 | 2 | 5 | 4 |
| 5 | BCA-DC205 | Object Oriented Programming using C++ | 2 | 1 | 0 | 2 | 5 | 4 |
| 6 | BCA-AE201 | Environmental Studies - I | 1 | 0 | 0 | 2 | 2 | 2 |
| 7 | BCA-AE202 | English Communication-II | 2 | 0 | 0 | 0 | 2 | 2 |
| 8 | FW-DG032 | Face the World Skills - II Socio - Emotional Learning | 2 | - | - | - | 2 | 2 |
| 9 | FW-DG072 | Face the World Skills - II Financial Literacy | 0 | – | 1 | - | 1 | 1 |
| **Total** | | | **17** | **5** | **1** | **6** | **28** | **27** |

# SEMESTER II

# BCA - DC201 | Discrete Mathematics

| Course Name: Discrete Mathematics | |
|---|---|
| **Semester II** | |
| **Course Code** | BCA-DC201 |
| **Course Title** | Discrete Mathematics |
| **Number of Credits** | Total: 4 (L: 3 T:1 ) |
| **Teaching hours** | 40 Hours |
| **Pre-Requisites** | Basic Knowledge of Mathematics |
| **Course Category** | Core |

## Rationale

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

## Learning Outcomes

At the end of the course student should be able:

1. To get knowledge about sets, relations, and functions.
2. To study the basics of lattices and graphs.
3. To get familiar with propositional logic.

## Syllabus

| Unit | Title | Hours |
|---|---|---|
| **Unit 1** | **Set Theory, Relations & Functions** | **10** |
| 1.1 | Sets, Definitions, Types of Sets, Operations on Sets, Multisets, Computer Representation of Sets. | 3 |
| 1.2 | Types of Relations, Composition, Inverse, Closure, Matrix Representation of Relation, Partial Ordering, Relational Model for Databases. | 4 |
| 1.3 | Functions, Types of Functions, Composition, Inverse Hash Function, | 3 |

| | | |
|---|---|---|
| | Growth of Function | |
| **Unit 2** | **Mathematical Logic** | **10** |
| 2.1 | Introduction, Propositions statements, Truth Tables, Logic Equivalence, Algebra of Propositions, Types of Propositions, Tautologies & Contradictions | 4 |
| 2.2 | Normal forms, Logic in proof, Methods of Proof, Mathematical Induction, Predicate Calculus. | 4 |
| 2.3 | Number Theory, Divisibility theory, Congruences, Application of Congruences | 2 |
| **Unit 3** | **Boolean Algebra & Logic Circuit** | **10** |
| 3.1 | Introduction, Boolean Algebra, Basic Operations, Boolean Function, DeMorgan's Theorem, SOP & POS Form, Normal Form, Boolean Function as a Canonical form, Simplification of Boolean Expression by Algebraic Method, K-Maps. | 5 |
| 3.2 | Group Theory, Definition, Types of Groups, Homomorphism & Isomorphism of Groups. | 5 |
| **Unit 4** | **Graph Theory and Automata** | **10** |
| 4.1 | Graph Theory, Types of Graphs, Operation on Graphs. | 3 |
| 4.2 | Language Grammar & Automata: Introduction, Strings, Languages, Regular Expression, Grammars, Finite State Machines, Finite State Automata, Moore & Mealy Machine, Pushdown Automata, Turing Machine | 7 |

## Learning Approach

To understand the basics of Discrete Mathematics and applications in Computer Science.

## References/suggested learning resources


- A Textbook Of Discrete Mathematics: C.L. Liu
- Discrete Mathematical Structures: Trembely and Manohar
- A Textbook of Discrete Mathematics: Swapan Kumar Sarkar
- Discrete Mathematics: S.K. Chakraborty and B.K.Sarka

# BCA-DC202 | Operating System

| Course Name: Operating System | |
|---|---|
| **Semester II** | |
| **Course Code** | BCA-DC202 |
| **Course Title** | Operating System |
| **Number of Credits** | Total: 4 (L: 3 T:1 P: 0) |
| **Teaching hours** | 40 Hours |
| **Prerequisites** | Basic Knowledge of Computer and programming |
| **Course Category** | Core |

## Rationale

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

## Learning Outcomes

Upon successful completion of this course, the student shall be able to:

1. Help students become familiar with the fundamental concepts of operating systems.
2. Help students become competent in recognizing operating systems features and issues.
3. Provide students with a sufficient understanding of operating system design and how it impacts application systems design and performance.
4. Exhibit familiarity with the fundamental concepts of operating systems.
5. Exhibit competence in recognizing operating systems features and issues.
6. Apply a mature understanding of operating system design and how it impacts application systems design and performance.

## Syllabus

| Unit | Title | Hours |
|---|---|---|
| **Unit 1** | **Introduction to Operating System, Processes & CPU Scheduling** | **10** |
| **1.1** | Introduction to Operating System: Introduction, Role, Types of OS; Batch Systems, multiprogramming, time-sharing, parallel, distributed, and real-time systems, Operating system structure, Operating system components, and services. | |
| **1.2** | Processes: Process Concept, Process Scheduling, Operation on Processes, Cooperating Processes, Threads. | |
| **1.3** | CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling. | |
| **Unit 2** | **Interprocess Communication, Synchronization & Dead Locks** | **10** |
| **2.1** | Interprocess Communication and Synchronization: Background, The Critical-Section Problem, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Critical Regions, Monitors, Message Passing. | |
| **2.2** | Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock, Deadlock Handling | |
| **Unit 3** | **Memory Management & Virtual Memory** | **12** |
| **3.1** | Memory Management: Background, Logical vs. Physical Address space, swapping, Contiguous allocation, Paging, Segmentation, Segmentation with Paging. | |
| **3.2** | Virtual Memory: Demand Paging, Page Replacement, Page-replacement Algorithms, Performance of Demand Paging, Allocation of Frames, Thrashing, Demand Segmentation | |
| **Unit 4** | **File-System Interface & Implementation** | **8** |
| **4.1** | File-System Interface: File Concept, Access Methods, Directory Structure. | |
| **4.2** | File-System Implementation: Introduction, File-System Structure, Basic File System, Allocation Methods, Free-Space Management, Directory Implementation. | |

## Learning Approach

To understand the basics of operating systems and its applications.

## References/suggested learning resources


- Silberschatz and Galvin, “Operating System Concepts”, John Wiley, 8th Ed., 2009.
- Milan Kovic., “Operating Systems”, Tata McGraw Hill, 2001
- Deitel, Deitel, and Choffnes, “Operating Systems”, Pearson,3rd Edition
- Tannenbaum, “Operating Systems”, PHI, 4th Ed., 2000.
- Madnick E. and Donovan J., “Operating Systems”, Tata McGraw Hill, 2001.
- Flynn McHoes, “Operating System”, Cengage Learning, 2006.
- Pbitra Pal Choudhury, “Operating System Principles and Design”, PHI, 2009.
- William Stallings, “Operating Systems Internals & Design Principles”, Pearson

# BCA - DC203 | Free & Open Source Software

# (Linux Programming & Administration)

| Course Name: Free & Open Source Software (Linux Programming & Administration) | |
|---|---|
| **Semester II** | |
| **Course Code** | BCA-DC203 |
| **Course Title** | Free & Open Source Software (Linux Programming & Administration) |
| **Number of Credits** | Total:4 (L: 2 T:1 P:2) |
| **Teaching hours** | 40 Hours |
| **Pre-requisites** | NIL |
| **Course Category** | Core |

## Rationale

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

## Learning Outcomes

At the end of the course student should be able to:

1. Explain the features of free & open source software
2. Familiarization with LINUX

## Syllabus

| Unit | Title | Hours |
|---|---|---|
| **Unit 1** | **Open source software** | **8** |
| 1.1 | Features, advantages over proprietary software, examples, Free software: concepts, features, Free software Vs Open Source software, Free software movements. Policies, GPL, Free OS | 4 |

| | | |
|---|---|---|
| 1.2 | Social Impact Open source vs. closed source, Open source government, Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source, and Open Source in Government. | 4 |
| **Unit 2** | **Introduction to Linux** | **8** |
| 2.1 | History and Features of Linux, Various flavors of Linux, Linux Kernel and Shell, Graphical Desktops- GNOME, KDE, Linux File System and Directories, Linux commands bc, cal, cat, cd, chgrp, chmod, clear, cmp, cp, kill, rm, rmdir, tty, wc, who, grep, write, telnet, whois, mv, find, ps, mkdir, more, date, mount, show, mount, etc. Pipeline and redirection concepts, using floppy and cd-rom in linux | 4 |
| 2.2 | Shell Programming: Available shells under Linux (viz. Bash, TCSH, Korn or so on), different Shell features, editors, shell commands, shell scripts: shell variables, environmental variables, the purpose of shell scripts, writing, storing and executing scripts, Filters- The grep family, advanced filters-sed and awk. | 4 |
| **Unit 3** | **Resource Management in Linux** | **8** |
| 3.1 | File and directory management, Command-line shortcuts, File Types, Ownership and Permissions, File management and manipulation, Moving users & its directories, Miscellaneous Tools, Editors, system calls for files Process Management, Signals, IPC: Pipes, FIFOs, System vs IPC, Message Queues, system calls for processes, Memory Management, library, and system calls for memory. | 5 |
| 3.2 | Introduction to Networking in LINUX: Socket Introduction, Elementary TCP Sockets (Socket Function, Connect Function, Bind, Listen, Accept, Fork and Exec), TCP Client server Example, Elementary UDP Sockets. | 3 |
| **Unit 4** | **Linux Administration** | **8** |
| 4.1 | Managing Users and Groups Creating and managing user/s and group commands, User management Tools, Users, and Access Permissions, Updating users and group attributes, PAM (Pluggable Authentication Modules) | 4 |
| 4.2 | Booting and Shutting down Boot Loaders, The init process, RC scripts, enabling and disabling services, Booting in recovery mode | 4 |

## List of Practical

1. Installation of Linux
2. Configuration of Linux
3. Shell Scripting Programs using various Looping, Read Statement, Formatted Output, Case Condition, etc
4. Writing and Executing C & C++ Programs
5. Using Vi Editor & various Commands
6. Find, awk, grep commands
7. User Administration
8. Installation of Web Server

## Learning Approach

To understand the basics of open-source operating systems and their applications.

## References/suggested learning resources

**a) Books**


- Richard Peterson, Linux Programming: A Beginners Guide, DreamTech.
- Unix Concepts and Applications by Sumitabha Das, Tata McGraw Hill Education, 2006
- Arnold Robbins, “Linux Programming by Examples: The Fundamentals”, Pearson Education, 2nd Ed., 2008.
- Mark G. Sobell, “A Practical Guide to Ubuntu Linux”, Pearson, 2nd Ed., 2008.
- Evi Nemeth, Garth Snyder, Trent R. Hein, “Linux Administrator Handbook”, Pearson, 2nd Ed., 2007.


**b) Open source software and website address**

- The Linux Documentation Project: http://www.tldp.org/
- Docker Project Home: http://www.docker.com
- Linux kernel Home: http://kernel.org
- Open Source Initiative: https://opensource.org/
- Linux Documentation Project: http://www.tldp.org/
- Wikipedia: https://en.wikipedia.org/
- https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia
- Github: https://help.github.com/
- The Linux Foundation: http://www.linuxfoundation.org/

# BCA-DC204 | Data & File Structure using C

| Course Name: Data & File Structure using C | |
|---|---|
| **Semester II** | |
| **Course Code** | BCA-DC204 |
| **Course Title** | Data & File Structure using C |
| **Number of Credits** | Total: 4 (L: 2 T:1 P: 2) |
| **Teaching hours** | 40 Hours |
| **Pre-requisites** | NIL |
| **Course Category** | Core |

## Rationale

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

## Learning Outcomes

At the end of the course student should be able to:

1. Familiarize with the fundamentals of data and file structures and their operations like insertion, deletion, searching, and sorting.
2. Understand and implement data structures like arrays, linked lists, stacks, queues, trees, graphs, and heaps.
3. Identify suitable data structures to model data used in real-world applications.
4. Familiarize with File Structures & Hashing Techniques

## Syllabus

| Unit | Title | Hours |
|---|---|---|
| **Unit 1** | **Introduction to Linear Data Structure: Array, Searching & Sorting Techniques** | **10** |
| 1.1 | Introduction to Data Structures: Basic Terminology, Elementary Data Organization, Classification of data structures and its operations, Abstract Data Type. | |
| 1.2 | Arrays, Sparse and Dense Matrices: Introduction to single and multidimensional arrays (up to three dimensions), Operations on | |

| | | |
|---|---|---|
| | Arrays, Memory Representation and address calculation, Introduction to Sparse Matrices and Dense Matrices, Sparse Matrices – Lower, Upper Triangular Matrices and Tridiagonal matrices; operations on Sparse Matrices. | |
| 1.3 | Searching: Introduction, Linear (Sequential) Search, and Binary Search. | |
| 1.4 | Sorting Techniques: Insertion Sort, Selection Sort, Bubble Sort, Quick Sort and Merge Sort. | |
| **UNIT 2** | **Introduction to Linear Data Structures: Stacks, Queues and Linked List** | **10** |
| 2.1 | Linked List: Introduction to linked lists and representation in memory, Singly linked list and operations such as traversal, insertion, deletion, searching on it, Introduction to Circular linked list, a Linked list with headers and Doubly Linked List, Applications of a Linked List.<br><br>Polynomial Representation and Operations using Arrays & Linked List. | |
| 2.2 | Stacks & Queues: Introduction and primitive operations on the stack, Arithmetic Expressions: Polish Notation, Stack application: Evaluation of postfix expression, Conversion from infix to postfix, Introduction and primitive operations on queues, Circular Queues, D-queues and Priority Queues, Application of Queues. | |
| **Unit 3** | **Introduction to Non-Linear Data Structures: Trees, Graphs and Heaps** | **10** |
| 3.1 | Trees: Introduction and terminology, Binary Trees, Representing Binary Trees in Memory, Linked Representation of Binary Tree, Traversal of binary trees, Recursive algorithms for tree operations such as traversal, insertion and deletion, Introduction to Expression Trees, Binary Search Trees, AVL trees, m-way search trees, Applications of Tree data structure. | |
| 3.2 | Graphs: Introduction and Terminology, Representation – Array Based, Linked and Set Representation, Graph Traversals and its applications. | |
| 3.3 | Heaps: Introduction, Structural Properties, Classification of Heap: Max-Heap Tree and Min-Heap Tree, Applications of Heap data structure. | |
| **Unit 4** | **Introduction to Hashing, File Structures, and File Indexing Techniques** | **10** |
| 4.1 | Hashing: Introduction, Hash Table, Hash Functions: Types and Requirements, Collision Management – Chaining and Open Addressing, Hashing applications. | |
| 4.2 | File Structures: Introduction, Concept of Fields, Records and Files, Unordered File (Heap File) and Ordered File, Sequential File | |

| | | |
|---|---|---|
| | Organization, Direct File Organization, and Indexed Sequential Organization, Choice of File Organization. | |
| 4.3 | File Indexing Techniques: Introduction, Single-Level Ordered Indexes: Primary Index, Clustering Index, and Secondary Index, Introduction to Multilevel indexes, Indexing with binary search trees, B-Tree and $B^+$-Tree. | |

## List of Practical

**The student is encouraged to use any programming language - C/C++/Java**

1. Programs on arrays (single-dimensional and multidimensional - up to three dimensions), array operations such as traversing, insertion, deletion, merging, etc.
2. Programs on matrices, Triangular matrices, Diagonal matrices, and Tridiagonal matrices.
3. Programs to determine whether a matrix is a Dense or Sparse matrix, operations on sparse matrices such as addition, subtraction, multiplication, and transpose.
4. Programs to perform linear search and binary search for an element from the entered numbers and determine its position using both arrays and linked lists.
5. Programs to implement different sorting techniques - Insertion Sort, Selection Sort, Bubble Sort, Quick Sort and Merge Sort.
6. Programs to implement a Linked List and perform operations on it such as Traversal, Searching, Determination of Predecessor and Successor of a node, Insertion, Deletion, Sorting, Merging, Reversing, Concatenation of Two Lists, Splitting of a Linked List.
7. Programs to implement Circular Linked List, Linked List with Header, and Doubly Linked List.
8. Programs on Polynomial Representation and Operations using arrays and linked lists.
9. Programs to implement Stack using array and linked list, Push and Pop operations on Stacks.
10. Program on Stack applications such as balanced parenthesis, evaluation of postfix expressions, and conversion of expressions from infix to postfix form.
11. Programs to implement Queue using array and linked list, Insertion & Deletion of the element from Queue.
12. Program to implement Circular Queue, D-Queue, and Priority Queues.
13. Programs to implement a binary tree and perform the inorder, preorder, and postorder traversal, insertion, and deletion of a node from the binary tree.
14. Program to implement a binary search tree and perform operations such as searching, insertion, and traversal.
15. Program to implement a graph, add or remove edges and vertices to the graph.
16. Program to construct a Heap given the Heap Elements.
17. Program to implement the Hash Function.
18. Program to prepare the Hashing Table, enter the elements and map them in the index using Hash Function.

## Learning Approach

To understand the basics of data structures and file structures and their applications.

## References/suggested learning resources

**a) Books**

- Introduction to Data Structures in C, Ashok N. Kamthane, Pearson
- Data Structures Using C, A.K.Sharma, Pearson
- Data Structures With C, Seymour Lipschutz, McGraw Hill Education
- Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed.

**b) Open source software and website address:**

- Code::Blocks: https://www.codeblocks.org/
- Visual Studio Code: https://code.visualstudio.com/
- Flowgorithm: http://www.flowgorithm.org/

# BCA-DC205 | Object Oriented Programming using C++

| Course Name: Object Oriented Programming using C++ | |
|---|---|
| **Semester II** | |
| **Course Code** | BCA-DC205 |
| **Course Title** | Object Oriented Programming using C++ |
| **Number of Credits** | Total: 4 (L: 2 T:1 P: 2) |
| **Teaching hours** | 40 Hours |
| **Pre-Requisites** | Introduction to Programming using 'C' Data Structure Concepts |
| **Course Category** | CORE |

## Rationale

The accelerated expansion of computing technologies and applications into all our lives means students need to understand the principles of computer science now, more than at any other time. It is necessary for all students to understand the ethical and social role of computer applications in society.

## Learning Outcomes

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

1. Learn how to implement Object Oriented concepts through C++.
2. To gain knowledge of objects, Class, Data Abstraction, Encapsulation, and Inheritance.
3. To understand Polymorphism and Dynamic Binding.
4. To know about constructing programs using the Bottom-up design approach.

## Syllabus

| Unit | Title | Hours |
|---|---|---|
| **Unit 1** | **Introduction to OOPS and C++ environment** | **10** |

| | | |
|---|---|---|
| 1.1 | Introduction: Introducing Object-Oriented Approach, Relating to other paradigms (functional, data decomposition). Features of Procedure oriented programming, Basic Concepts of Object Oriented Programming, Benefits of OOP, Applications of OOP, Difference between C and C++, cin, cout, new, delete operators. | |
| 1.2 | C++ Environment: Program development environment, the language and the C++ language standards, C++ standard libraries. Introduction to various C++ compilers, C++ standard libraries, Testing the C++ program in Turbo C++/Borland, C++/Microsoft VC++/GNU C++ compiler. | |
| **UNIT 2** | **Classes and Objects** | **10** |
| 2.1 | Classes and Objects: Encapsulation, information hiding, abstract data types, Object & classes, attributes, methods, C++ class declaration, references, this pointer, Function Overloading, Constructors and destructors, instantiation of objects, Default parameter value, C++ garbage collection, Dynamic memory allocation, Meta class/abstract classes. | |
| **Unit 3** | **Inheritance and Polymorphism** | **10** |
| 3.1 | Inheritance and Polymorphism: Inheritance, Class hierarchy, derivation – public, private & protected, Aggregation, composition v/s classification hierarchies, Polymorphism, Categorization of polymorphism techniques, Method polymorphism, Polymorphism by parameter, Operator overloading, Parametric polymorphism, Virtual Function, Early v/s Late Binding. | |
| **Unit 4** | **Generic Programming and Files and Exception Handling** | **10** |
| 4.1 | Generic Programming – Introduction, templates, template functions, Overloading of template functions, Overriding inheritance methods. | |
| 4.2 | Files and Exception Handling: Persistent objects, Streams and files, Namespaces, The basic stream classes: C++ predefined streams, Error handling during file operations, Command Line Arguments. Types of Exception, Catching and Handling Exceptions. | |

## List of Practicals

Practicals will be based on the topic covered in the content. Therefore students must understand the concept and implementation of the contents in the syllabus by doing practicals as suggested by the faculty member.

## Learning Approach

To understand the basics of Object-Oriented concept and their implementation.

## References/suggested learning resources

**a) Books**


- Ashok N. Kamthane, “Object-Oriented Programming With Ansi And Turbo C++”, Pearson Education.
- A.R.Venugopal, Rajkumar, T. Ravishanker “Mastering C++”, TMH, 1997.
- E. Balguruswamy, “C++ ”, TMH Publication ISBN 0-07-462038-x.
- Mahesh Bhave, “Object-Oriented Programming with C++”, Pearson Education.
- D. Parasons, “Object-Oriented Programming with C++”, BPB Publication.
- Lawlor, “The Art of Programming Computer Science with C++”, Vikas Publication.
- Schildt Herbert, “C++: The Complete Reference”, 4th Ed., Tata McGraw Hill, 1999.
- R. Lafore, “Object-Oriented Programming using C++”, Galgotia Publications, 2004.


**b) Open source software and website address**

- TURBO C / Code::Blocks
- TURBO C++
- https://www.udemy.com
- https://www.coursera.org