UNIVERSITY OF MADRAS

BACHELOR OF COMPUTER APPLICATIONS (BCA)
DEGREE PROGRAMME
SYLLABUS WITH EFFECT FROM 2023-2024

Year : I **Semester: II**

Object Oriented Programming using C++ Common for B.C.A. , B.Sc.-SA	**120C2A**
Credits 5	**Lecture Hours:4 per week**

Learning Objectives: (for teachers: what they have to do in the class/lab/field)

- To engender an appreciation for the need and characteristics of Object-orientation.
- To impart knowledge of the C++ language grammar in order to design and implement programming solutions to simple problems by applying Object-oriented thinking.

Course Outcomes: (for students: To know what they are going to learn)
CO1: Explain the various basic concepts of Object-orientation.
CO2: Write programs to implement static binding
CO3: Write programs to implement inheritance and dynamic binding
CO4: Write programs to implement templates and exception handling and learn how to use STL class library.
CO5: Write programs implementing File and Stream I/O.

Units	**Contents**
I	Object Oriented Programming Concepts: Complexity in software - The need for object-orientation – Abstraction – Encapsulation – Modularity – Hierarchy. Basic Elements of C++: Classes – Objects – Data members and member functions – private and public access specifiers - Static members - Constructors – Singleton class - Destructors - Friend Functions and Friend Classes - Array of objects – Pointer to objects - this pointer – References – Dynamic memory allocation - Namespaces.
II	Function Overloading: Overloading a function - Default arguments – Overloading Constructors. Operator Overloading: Overloading an operator as a member function – Overloading an operator as a friend function – Overloading the operators [], (), -> and comma operators – Conversion Functions.
III	Inheritance: Types of inheritance – protected access specifier –Virtual Base Class – Base class and derived class constructors. Run-time Polymorphism: Virtual Functions – Function overriding - Pure virtual function – Abstract base class.
IV	Templates: Function templates – Overloading a function template – Class templates. Standard Template Library (STL): Containers: vector, list – Iterators: forward, backward – Algorithms: removing and replacing elements, sorting, counting, reversing a sequence. Exception Handling: Exceptions – try, catch, throw – Rethrowing an exception – Restricting exceptions - Handling exceptions in derived classes - terminate(), abort(), unexpected(), set_terminate().
V	I/O Streams: Formatted I/O with ios class functions - Manipulators – Creating own manipulator – Overloading << and >> operators. File I/O: fstream class – Opening and closing a file – Reading from and writing to a text file - Unformatted and Binary I/O – Random access I/O.

Learning Resources:

Recommended Texts

1. Herbert Schildt, *C++ - The Complete Reference*, Third Edition, TMH, 1999.
2. Grady Booch, *Object Oriented Analysis and Design*, Pearson Education, 2008. (For Unit I)

Reference Books

1. Bjarne Strousstrup, *The C++ Programming Language*, Addison Wesley, 2000.
2. J. P. Cohoon and J. W. Davidson, *C++ Program Design – An Introduction to Programming and Object-Oriented Design*, Second Edition, McGraw Hill, 1999.
3. C. J. Lippman, *C++ Primer*, Third Edition, Addison Wesley, 2000.