

Programming in C++ Semester -II

Theory	4 Hours/Week	4 credits
Practical	3 Hours/Week	1 credit

Unit – I

Introduction to C++: Applications, Example Programs, Tokens, Data Types, Operators, Expressions, Control Structures, Arrays, Strings, Pointers, Searching and Sorting Arrays.
Functions: Introduction, Prototype, Passing Data by Value, Reference Variables, Using Reference Variables as Parameters, Inline Functions, Default Arguments, Overloading Functions, Passing Arrays to Functions.
Object Oriented Programming: Procedural and Object-Oriented Programming, Terminology, Benefits, OOP Languages, and OOP Applications.

Unit – II

Classes: Introduction, Defining an Instance of a Class, Why Have Private Members? Separating Class Specification from Implementation, Inline Member Functions, Constructors, Passing Arguments to Constructors, Destructors, Overloading Constructors, Private Member Functions, Arrays of Objects, Instance and Static Members, Friends of Classes, Member-wise Assignment, Copy Constructors, Operator Overloading, Object Conversion, Aggregation.

Unit – III

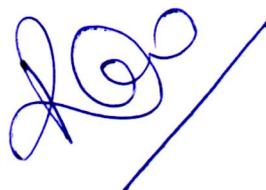
Inheritance: Introduction, Protected Members and Class Access, Base Class Access Specification, Constructors and Destructors in Base and Derived Classes, Redefining Base Class Functions, Class Hierarchies, Polymorphism and Virtual Member Functions, Abstract Base Classes and Pure Virtual Functions, Multiple Inheritance.
C++ Streams: Stream Classes, Unformatted I/O Operations, Formatted I/O Operations.

Unit – IV

Exceptions: Introduction, Throwing an Exception, Handling an Exception, Object-Oriented Exception Handling with Classes, Multiple Exceptions, Extracting Data from the Exception Class, Re-throwing an Exception, Handling the bad_alloc Exception.
Templates: Function Templates–Introduction, Function Templates with Multiple Type, Overloading with Function Templates, Class Templates – Introduction, Defining Objects of the Class Template, Class Templates and Inheritance, Introduction to the STL.

Text Tony Gaddis, Starting out with C++: from control structures through objects (7e)

References B. Lippman, C++ Primer
Bruce Eckel, Thinking in C++
K.R. Venugopal, Mastering C++
Herbert Schildt, C++: The Complete Reference
Bjarne Stroustrup, The C++ Programming Language
Sourav Sahay, Object Oriented Programming with C++



CHAIRMAN
Board of Studies
Department of Computer Science
KAKATIYA UNIVERSITY
VAREASIGUDA - 508001 (T.S.)

C++ Lab Semester -II

Practical

3 Hours/Week

1 credit

- 1 Write a program to.
 - a. Print the sum of digits of a given number.
 - b. Check whether the given number is Armstrong or not
 - c. Print the prime number from 2 to n where n is natural number given.
- 2 Write a program to find largest and smallest elements in a given list of numbers and sort the given list.
- 3 Write a program to read the student name, roll no, marks and display the same using class and object.
- 4 Write a program to implement the dynamic memory allocation and de-allocation using new and delete operators using class and object.
- 5 Write a program to find area of a rectangle, circle, and square using constructors.
- 6 Write a program to implement copy constructor.
- 7 Write a program using friend functions and friend class.
- 8 Write a program to implement constructors
 - § Default Constructor, Parameterized Constructor, Copy Constructor
 - § Define the constructor inside/outside of the class
 - § Implement all three constructors within a single class as well as use multiple classes(individual classes)Write a program to implement the following concepts using class and object
 - § Function overloading
 - § Operator overloading (unary/binary(+ and -))Write a program to demonstrate single inheritance, multilevel inheritance and multiple inheritances.
Write a program to implement the overloaded constructors in inheritance.
Write a program to implement the polymorphism and the following concepts using class and object.
 - § Virtual functions
 - § Pure virtual functionsWrite a program to implement the virtual concepts for following concepts
 - § Constructor (not applied)
 - § Destructor (applied)Write a program to demonstrate static polymorphism using method overloading.
Write a program to demonstrate dynamic polymorphism using method overriding and dynamic method dispatch.
Write a program to implement the template (generic) concepts
 - § Without template class and object
 - § With template class and object

Write the Pseudo Code and draw Flow Chart for the above programs.

Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows.

CHAIRMAN
Board of Studies
Department of Computer Science
KAKATIYA UNIVERSITY
VARANASI, U.P. (INDIA)