

# CAP455:OBJECT ORIENTED PROGRAMMING USING C++

L:3 T:0 P:2 Credits:4

**Course Outcomes:** Through this course students should be able to

CO1 :: define the fundamental concepts of object oriented programming

CO2 :: understand the working of pointers, arrays and strings

CO3 :: apply dynamic memory management in polymorphism and inheritance to solve real world problems

CO4 :: analyze the unexpected situations and manage them using exception handling mechanism

## Unit I

**Principles of OOP and C++ essentials** : procedural vs object oriented programming paradigms, Input/output streams, classes and objects, structure vs union, enumerations and classes, static data members and functions, user defined functions, inline function, friend function and friend class, reference variables, differentiate among call by value, call by address and call by reference, recursion

## Unit II

**Handling pointers, arrays and string** : pointer vs reference variables, void pointer, pointer arithmetic, pointer to pointer, dangling pointer, wild pointer, null pointer assignment, pointers as class member, pointer to objects, this pointer, pointer to data member, array declaration and processing of multidimensional arrays (inside main and inside class), array of objects, standard C++ string class- defining and assigning string objects, modifiers of string class

## Unit III

**Constructors, destructors and managing file operations** : features of constructor function, default constructor, constructor vs normal function, parameterized constructor, copy constructor, initializer lists, constructor with default arguments, destructor, opening and closing of files, modes of file, file stream functions, reading/writing of files, sequential access and random access for file processing, binary file operations, classes and file operations, structures and file operation

## Unit IV

**Operator overloading, type casting and re-usability** : importance of operator overloading, unary operator overloading, binary operator overloading, type conversions - basic type to class type and class type to basic type, importance of re-usability (inheritance), basics of inheritance - base class and derived class, types of inheritance (simple, multi-level, multiple, hierarchical and hybrid), modes of inheritance (private, protected, public), overriding of member functions, order of execution of constructors and destructor, resolving ambiguities in inheritance, virtual base class

## Unit V

**Dynamic memory management and polymorphism** : importance of dynamic memory allocation, dynamic memory allocation using new and delete operators, memory leak and allocation failures, virtual destructor, compile time polymorphism vs run time polymorphism, virtual functions, pure virtual functions, abstract classes and concrete class, self-referential classes, early binding and late binding, dynamic constructors

## Unit VI

**Handling exceptions, templates and STL** : basics of exception handling, exception handling mechanism, throwing exception mechanism, catching exception mechanism, re-throwing an exception, function template and class template, class template and inheritance, introduction and importance of Standard Template Library (STL) - containers, algorithms and iterators, container - vector and list

### List of Practicals / Experiments:

#### List of practicals/ experiments

- Principles of OOP and C++ essentials: Programs to define classes and structures, Program to demonstrate inline, non inline member functions and Static function, Program to implement function overloading, friend function and friend class, Program to demonstrate the difference between call by value, call by address and call by reference
- Handling pointers, arrays and string: Program to demonstrate the type of pointers, Program to process multidimensional array and array of objects

- Constructors, destructors and managing file operations: Program to demonstrate constructor, destructor and type of constructors, Program to demonstrate the modes of file, Program to demonstrate type of files
- Operator overloading, type casting and re-usability: Program to demonstrate the operator overloading and type conversion, Program to demonstrate the type of inheritance, Program to demonstrate the ambiguities in inheritance
- Dynamic memory management and polymorphism: Program to use new and delete for dynamicmemory management, Program to demonstrate the compile time and run time polymorphism, Program to demonstrate abstract class and dynamic constructor
- Handling exceptions, templates and STL: Program to demonstrate exception handling, Program to demonstrate function template and class template, Program to demonstrate STL- Containers, Algorithms and Iterators

**Text Books:** 1. OBJECT- ORIENTED PROGRAMMING WITH ANSI AND TURBO C+ + by ASHOK KAMTHANE, Pearson Education India

**References:** 1. OBJECT ORIENTED PROGRAMMING IN C++ by ROBERT LAFORE, PEARSON  
2. C++: THE COMPLETE REFERENCE by HERBERT SCHILDT, MCGRAW HILL EDUCATION