L	T	P	C
3	0	0	3

Course Code: CS E213
Semester: III

OBJECT ORIENTED PROGRAMMING

Course objective: This course will help the learner to employ object oriented concepts for developing programs catering to different applications

UNIT - I 11 Periods

Procedural programming, An Overview of C: Types- Fundamental types, boolean, Character type, Integer types, Float types, prefixes and suffixes**,** void**,** Declarations-scope**,** Reference data types, Variables, Declared Constants, enumerated constants, the typecasting operator, Operator and Expressions, **Input and Output** (*C*-way), **Statements**- Statement Summary, Declarations as Statements, Selection Statements, Iteration Statements, goto Statements**, Pointers, Arrays, and References**-Pointers, arrays, pointers into arrays, pointers and const, Pointers and Ownership, References, **Functions** - Function Declarations, argument passing-reference argument, array arguments, overloaded functions, **Error handling**, **Namespaces**, **Preprocessor directive:** Trigraph sequence, Digraph Sequence, #define, #undef, #ifdef, #ifndef, #if, #endif, #else, #elif and #line

UNIT - II 11 Periods

Some difference between C and C++: Single line comments, Local variable declaration within function scope, function declaration, function overloading, Reference variable, parameter passing – value vs reference, passing pointer by value or reference, Operator new and delete, Inline Functions in contrast to macro. **The Fundamentals of Object Oriented Programming:** Necessity for OOP, Data Hiding, Data Abstraction, Encapsulation, Procedural Abstraction, Class and Object, **More extensions to C in C++ to provide OOP Facilities:** Scope of Class and Scope Resolution Operator, Member Function of a Class, private, protected and public Access Specifier, this Keyword, Constructors and Destructors, **Error handling (exception)**

UNIT - III 12 Periods

Essentials of Object Oriented Programming: Operator overloading, **Polymorphism-** Overloading, **Class relationship-**Inheritance – Single and Multiple, Virtual and abstract base class, Friend class, Class Hierarchy, Inherited constructors, Pointers to Objects, Assignment of an Object to another Object, Polymorphism through dynamic binding, Virtual Functions, overriding and hiding, **Generic Programming: Template**-class template, function template, template specialization

UNIT - IV 11 Periods

Input and Output: Streams, Files, Library functions, formatted output. **Object Oriented Design and Modelling**: UML concept, Use case for requirement capturing, Class diagram, Activity diagram and Sequence Diagram for design, Corresponding C++ code from design

TEXT BOOKS

1. Bjarne Stroustrup, The C++ Programming Language, Pearson Addison-Wesley Professional, US, 4th Edition, 2013.
2. Debasish Jana, C++ and Object-Oriented Programming Paradigm, PHI Learning Pvt. Ltd., New Delhi, 3rd Edition, 2014.

REFERENCE BOOKS

1. Bjarne Stroustrup, Programming - Principles and Practice Using C++, Pearson Addison-Wesley Professional, US, 2nd Edition, 2014.
2. Bjarne Stroustrup, The Design and Evolution of C++, Pearson Addison-Wesley Professional, US,1st Edition, 1994.

UNITWISE LEARNING OUTCOMES

After successful completion of the course the leaner will be able to

Unit I	• Illustrate the object-oriented concepts
Unit II	• Apply the concepts of classes and objects for a given application
Unit III	• Construct user defined data types with overloaded operators • Develop applications by making use of inheritance
Unit IV	• Develop applications using file streams

COURSE OUTCOMES

Upon successful completion of this course, the learner will be able to

- Describe the object-oriented concepts and write programs using basic constructs and functions in C++
- Employ the concepts of classes and objects for a given application
- Create user defined data types and demonstrate operator overloading, inheritance, data conversion
- Use pointers and demonstrate memory management, virtual and friend functions
- Develop applications using file streams

L	T	P	C
0	0	2	1

Course Code: CSE216
Semester: III

OBJECT ORIENTED PROGRAMMING LABORATORY

Course Objective: To help the learners understand the underlying concepts of C++ language and able to develop programs for various real time problems and computations

LIST OF EXPERIMENTS

1. Programs using Branching
2. Programs using Multi Dimensional Array
3. Programs using Function Overloading and Inline Functions, pass and return by reference
4. Programs using preprocessor directives
5. Programs using Classes and Objects (Array as Data Member and Array of Objects)
6. Programs using Constructors and Destructor
7. Programs using Operator Overloading
8. Programs using Inheritance
9. Programs using Virtual Functions and Friend Functions
10. Programs using Templates
11. Programs using Files
12. Programs using Exception Handling

COURSE LEARNING OUTCOMES

Upon successful completion of this course, the learner will be able to

- Apply overloading concepts for functions and inline functions.
- Create the user defined objects by creating classes with constructors
- Differentiate string in C and ‘string’ class in C++ and the built in functions to do operations on strings.
- Apply data conversion from one unit to another for both in built data types and user defined using overloading member functions
- Create base and derived classes and inherit the parent class properties along with virtual and friend functions
- Develop applications using file streams and multi files and templates

.