

GANPAT UNIVERSITY									
FACULTY OF TECHNOLOGY									
Programme	Bachelor of Technology				Branch/Spec.	Computer Science & Engineering (CBA/MA/BDA)			
Semester	II				Version	1.0.1.0			
Effective from Academic Year	2018-19				Effective for the batch Admitted in	June 2018			
Subject code	2CSE203		Subject Name	Essentials of Software Foundation & Programming – II					
Teaching scheme					Examination scheme (Marks)				
(Per week)	Lecture(DT)		Practical(Lab.)		Total	CE	SEE	Total	
	L	TU	P	TW					
Credit	3	0	2	0	5	Theory	40	60	100
Hours	3	0	4	0	7	Practical	40	60	100
Pre-requisites:									
C Programming, Flow Charting.									
Learning Outcome:									
After completing the Course, students will learn:									
<ul style="list-style-type: none"> • Articulate the principles of object-oriented problem solving and programming. • Outline the essential features and elements of the C++ programming language. • Explain programming fundamentals, including statement and control flow and recursion. • Apply the concepts of class, method, constructor, instance, data abstraction, function abstraction, inheritance, overriding, overloading, and polymorphism. • Program with basic data structures using array • Program using objects and data abstraction, class, and methods in function abstraction. • Analyze, write, debug, and test basic C++ codes using the approaches introduced in the course. • Analyze problems and implement simple C++ applications using an object-oriented software engineering approach. 									
Theory syllabus									
Unit	Content								Hrs
1	Introduction to dynamic memory allocation C programming basics, dynamic memory allocation functions								2
1	Introduction to OOP What is object oriented programming? Why do we need object oriented, Programming characteristics of object-oriented languages, C and C++.								3
2	C++ Programming basics Output using cout Directives. Input with cin, type bool, thesetw manipulator, type conversions.								2
3	Functions Returning values from functions. Reference arguments. Overloaded function. Inline function. Default arguments. Returning by reference.								4
4	Object and Classes Making sense of core object concepts (Encapsulation, Abstraction, Polymorphism, Classes, Messages Association, Interfaces) Implementation of class in C++, C++ Objects as physical object, C++ object as data types constructor. Object as function arguments. The default copy constructor, returning object from function. Structures and classes. Classes objects and memory static class data. Const and classes								5
5	Arrays and string arrays fundamentals Arrays as class member data, Arrays of object, string, the standard C++ String class								3

6	Operator overloading Overloading unary operations. Overloading binary operators, data conversion, pitfalls of operators overloading and conversion keywords. Explicit and Mutable.	4
7	Inheritance Concept of inheritance. Derived class and based class. Derived class constructors, member function, inheritance in the English distance class, class hierarchies, inheritance and graphics shapes, public and private inheritance, aggregation: Classes within classes, inheritance and program development	4
8	Pointer Addresses and pointers. The address of operator and pointer and arrays. Pointer and Faction pointer and C-types string. Memory management -New and Delete, pointers to objects, debugging pointers	5
9	Virtual Function: Virtual Function, friend function, Static function, Assignment and copy initialization, this pointer, dynamic type information	3
10	Streams and Files Streams classes, Stream Errors, Disk File I/O with streams, file pointers, error handling in file I/O with member function, overloading the extraction and insertion operators, memory as a stream object, command line arguments, and printer output	4
11	Templates and Exceptions Function templates, Class templates Exceptions	3
Practical content		
<ul style="list-style-type: none"> • Implement programs to learn dynamic memory allocation using its functions. • Implement programs with the use of cin, cout and various arithmetic operators. • Write programs implementing signed & unsigned types, other data types, type conversion and library functions. • Implement the programs using manipulators, various operators, looping and decision structures. • Implement programs to learn about functions, inline function, function overloading, structures, scope and storage class. • Implement programs of class, objects and array concepts. • Implement programs to learn about string objects, enum types and operator overloading • Implement programs of single inheritance, hierarchical inheritance, multiple and multilevel inheritance, hybrid inheritance and aggregation. • Write programs implementing pointers, polymorphism and virtual functions. • Write programs implementing friend function, static member function, this pointer, operator overloading with file and command line argument. • Implement programs to learn about reading and writing files with OOP. • Implement programs to learn about function template, class template and exception handling. 		
Text Books		
1	Object Oriented Programming in C++ By Robert Lafore	
Reference Books		
1	Thinking in C++ By Bruce Eckel	
2	C++ : The complete Reference By Herbert Schildt	
3	Object Oriented Programming with C++ By E. Balagurusamy	