

OBJECT ORIENTED PROGRAMMING

CONTENTS

UNIT-I PRINCIPLES OF OBJECT-ORIENTED PROGRAMMING

1.1	Characteristics of object-oriented languages.....	1.1
1.2	C++ Program structure	1.7
1.3	Procedure Oriented Programming vs Object Oriented Programming	1.12
1.4	C++ constructs and syntax	1.13
1.5	Tokens	1.16
1.6	Variables	1.17
1.7	Data-types	1.21
1.7.1	Basic Data Types	1.22
1.7.2	Derived Data Types	1.26
1.7.3	User-Defined Data Types	1.28
1.8	Type conversion	1.32
1.9	Operators	1.36
1.9.1	Arithmetic Operators	1.37
1.9.2	Relational Operators	1.38
1.9.3	Logical Operators	1.40
1.9.4	Bitwise Operators	1.41
1.9.5	Assignment Operators	1.42
1.9.6	Ternary or Conditional Operators	1.44
1.9.7	Miscellaneous Operators	1.45
1.9.8	Operator Precedency and Associativity	1.46
1.10	Expressions	1.46
1.11	Namespace	1.49
1.12	Flow Control and Decision Making Statements	1.57
1.12.1	Selection Statement	1.59
1.12.2	Looping Statements	1.63
1.12.3	Jump Statements in Programming	1.67
	Short Question and Answers	1.70
	Big Questions	1.78
	Practical	1.79

UNIT-II CLASSES AND OBJECTS

2.1	Abstraction mechanism: Classes	2.1
-----	--------------------------------------	-----

2.2	Objects	2.2
2.3	Member data and member functions	2.4
2.4	Constructors and types	2.12
2.5	Destructors	2.17
2.6	Inline function	2.20
2.7	Friend function	2.21
2.8	Array of objects	2.28
2.9	Objects as function arguments.....	2.32
2.10	Memory allocation for objects	2.36
2.11	Static members static data and static function	2.39
2.11.1	Static Data Member	2.39
2.11.2	Static Member Function	2.43
Short Question and Answers		2.47
Big Questions		2.51
Practical		2.53
 UNIT-III INHERITANCE AND COMPILE TIME POLYMORPHISM		
3.1	Inheritance: Derived Classes	3.1
3.2	Types of Inheritance	3.5
3.2.1	Single inheritance	3.5
3.2.2	Multilevel Inheritance	3.8
3.2.3	Multiple Inheritance	3.10
3.2.4	Hierarchical inheritance	3.12
3.2.5	Hybrid inheritance	3.15
3.2.6	Advantages of Inheritance in C++	3.17
3.2.7	Disadvantages of Inheritance in C++	3.17
3.3	Operator Overloading: Compile time Polymorphism	3.17
3.4	Overloading Functions	3.19
3.5	Overloading Operators	3.23
3.5.1	Overloading Unary Operators	3.25
3.5.2	Overloading Binary Operators	3.29
3.5.3	Rules for operator overloading.....	3.32
3.5.4	Which operators Cannot be overloaded?	3.32
3.5.5	Advantages of an operator overloading in C++	3.33

3.5.6	Disadvantages of an operator overloading in C++	3.33
3.6	Operator Overloading with Friend Functions	3.34
Short Question and Answers		3.38
Big Questions		3.41
Practical		3.42
 UNIT-IV POINTERS AND RUNTIME POLYMORPHISM		
4.1	Pointers with arithmetic operations	4.1
4.1.1	Introduction to Pointers	4.1
4.1.2	Arithmetic Operations of Pointers	4.3
4.2	this pointer	4.11
4.2.1	Return Calling Object's Reference Using this Pointer	4.12
4.2.2	Characteristics of the "this" Pointer	4.15
4.2.3	Key uses of this pointer	4.16
4.2.4	Limitations of this Pointer	4.16
4.3	Pointers to Derived classes and Base classes	4.17
4.3.1	Example of Pointer to Classes	4.17
4.3.2	Pointers to Base Class Objects	4.18
4.3.3	Pointers to Derived Class Objects	4.19
4.3.4	Base Class Pointer Pointing to a Derived Class Object (Polymorphism)	4.20
4.3.5	Base Class Pointer with Virtual Functions (Runtime Polymorphism)	4.23
4.4	Compile time versus Runtime Polymorphism	4.24
4.5	Virtual functions	4.26
4.5.1	Rules for Virtual Functions	4.30
4.5.2	Limitations of Virtual Functions	4.30
4.6	Late Binding	4.30
4.7	Abstract classes	4.32
4.8	Pure virtual functions	4.35
4.9	Virtual Destructors	4.37
4.10	Virtual base class	4.38
Short Question and Answers		4.41
Big questions		4.44
Practical		4.45

UNIT-V TEMPLATES AND EXCEPTION HANDLING

5.1	Templates	5.1
5.2	Class Templates	5.2
5.3	Function Templates.....	5.6
5.4	Overloading of Template Functions	5.7
5.5	String	5.10
5.5.1	Define a C-string	5.10
5.5.2	String Object	5.12
5.5.2.1	Basic Operations in String	5.12
5.5.2.2	String Functions	5.18
5.6	Iterators	5.22
5.6.1	Container Iterator Functions	5.22
5.6.2	Iterators Operations	5.24
5.6.3	Types of Iterators	5.29
5.6.4	Features	5.30
5.6.5	Iterator Adaptors	5.32
5.6.6	Iterator Utility Functions in C++	5.33
5.6.7	Applications of Iterators with Examples	5.34
5.7	Hashes	5.37
5.8	IO streams	5.39
5.8.1	Streams	5.40
5.8.2	C++ IO Headers, Templates and Classes	5.41
5.8.3	Buffered IO	5.44
5.8.4	The <iostream> Header and the Standard Stream	5.46
5.8.5	The Stream Insertion << and Stream Extraction >> Operators	5.47
5.8.6	The ostream Class	5.47
5.8.7	The istream class	5.49
5.8.8	Unformatted Input/Output Functions	5.50
5.8.9	States of stream	5.52
5.8.10	Formatting Input/Output via Manipulators in <iomanip> and <iostream>	5.52
5.9	Exception Handling	5.58
5.9.1	Standard Exceptions	5.61
5.9.2	try-catch Block	5.62

5.9.3	Throwing Exceptions	5.62
5.9.3.1	Throwing Built-in Types	5.63
5.9.3.2	Throwing Standard Exceptions	5.64
5.9.3.3	Throwing Custom Exceptions	5.64
5.9.4	Catching Exceptions	5.66
5.9.4.1	Catch by Value or Reference	5.66
5.9.4.2	Multiple catch Statements	5.68
5.9.4.3	Nested Try Catch Blocks	5.71
5.9.5	Catch an Unknown Exception	5.73
5.9.6	Rethrowing an Exception	5.74
5.9.7	Processing Unexpected Exceptions	5.76
5.9.8	Advantages of Exception Handling in C++	5.77
5.9.9	Disadvantages of Exception Handling in C++	5.77
Short Question and Answers		5.78
Big Questions		5.85
Practical		5.86
 UNIT-VI I/O SYSTEMS AND FILE I/O		
6.1	C++ Streams	6.1
6.2	Formatted and Unformatted I/O	6.9
6.2.1	Formatted I/O	6.9
6.2.2	Unformatted I/O	6.27
6.3	File stream classes	6.33
6.4	File modes	6.34
6.5	File operations	6.35
6.6	Sequential Read / Write operations	6.45
6.7	Binary and ASCII Files	6.48
6.8	Error handling in file I/O with member function.	6.52
6.8.1	Error Handling Functions	6.52
6.8.2	Error handling in file I/O using member functions	6.56
Short Question and Answers		6.59
Big Questions		6.82
Practical Activities		6.83