U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

January / February 2025 Semester End Main Examinations

Programme: B.E. **Semester: III**

Branch: CSE (IoT & Cybersecurity including Blockchain) **Duration: 3 hrs.**

Course Code: 23IC3PCOOP **Max Marks: 100**

Course: Object Oriented Programming with C++

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

			UNIT - I	***CO***	***PO***	**Marks**
1	a)		Explain salient features of Object-oriented programming language. How it is different than procedure-oriented programming.	*CO1*	*PO1*	**10**
	b)		Write a C++ program to find the largest three elements in an array	*CO2*	*PO2*	**05**
	c)		Write a program to illustrate call by value, call by address and call by reference in C++.	*CO1*	*PO1*	**05**
			OR			
2	a)		Explain the following terms with an example i) Typecast Operator ii) Bitwise Expressions iii) Logical Expressions	*CO1*	*PO1*	**06**
	b)		Create a Class **Employee** with member variables Id, Name and Location. Write member functions to read and display the data. Write a main function to create ten objects of the Employee class. Define member functions outside the class.	*CO2*	*PO2*	**08**
	c)		With suitable examples demonstrate the usage of access specifiers in C++ and how do they control the visibility of class members?	*CO1*	*PO1*	**06**
			UNIT - II			
3	a)		Write a C++ program to demonstrate default, parameterized and copy constructor.	*CO1*	*PO1*	**10**
	b)		With suitable C++ code demonstrate the implementation of a friend function named Check_MIN in C++, which compares the integer data of two classes, Foo1 and Foo2, and prints the minimum value found? Assume that both classes have a single integer data member.	*CO3*	*PO3*	**10**

			OR			
	4	a)	Write a C++ program, create class **Matrix** with member variables mat[10][10], row, column and member functions *read_matrix(), print_matrix()* and *add().* Define all functions outside the class. The prototype of add() function is: void add(Matrix m1,Matrix m2); Implement the addition of two Matrices in the program.	*CO3*	*PO3*	**10**
		b)	Explain inline functions. What are the advantages of inline functions?	*CO1*	*PO1*	**05**
		c)	Explain friend function with example and list some of the special properties of friend function.	*CO1*	*PO1*	**05**
			UNIT - III			
	5	a)	Develop a C++ program to overload binary operator to Subtract Complex Number.	*CO3*	*PO3*	**10**
		b)	Create a class named **Fruit** with a data member to calculate the number of fruits in a basket. Create two other class named **Apples** and **Mangoes** derived from class **Fruit** to calculate the number of apples and mangoes in the basket. Print the number of fruits of each type and the total number of fruits in the basket.	*CO2*	*PO2*	**10**
			OR			
	6	a)	Write a program to overload pre-increment and post-increment for a class example with two member variables.	*CO2*	*PO2*	**10**
		b)	Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class **Publication** that stores the *title* (a string) and *price* (type float) of a publication. From this class derive two classes: **Book**, which adds a page count (type int), and **Tape**, which adds a playing time in minutes (type float). Each of these three classes should have a *getdata()* function to get its data from the user at the keyboard, and a *putdata()* function to display its data. Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().	*CO3*	*PO3*	**10**
			UNIT - IV			
	7	a)	Write a C++ Program to maintain an **Employee Database** using Virtual class. Functions to be implemented are create, display, update. Employee details should contain Name, Emp-id, Salary and Experience.	*CO1*	*PO1*	**08**
		b)	Design a base class **Circle** with member variables (*radius and color*) of type double, methods (*getRadius(), getArea()*) and constructors (*Circle(radius), Circle(radius, color)*). Derive subclass called **Cylinder** from the superclass Circle with member variable (height) of type double, public methods (getHeight(), getVolume(), getArea()) and its constructors(Cylinder(height,	*CO3*	*PO3*	**08**

			radius), Cylinder(height, radius, color)). Create the two instances of cylinder and print similar cylinders if the area, volume and color of cylinders are same. Demonstrate the code reuse and polymorphism properties of Object-oriented programming by inheriting the constructors and methods of the base class.			
		c)	Write a program with a Class **Complex** that has two member variables to demonstrate pointer to objects.	*CO1*	*PO1*	**04**
			OR			
	8	a)	Explain pure virtual function and abstract classes. Create a pure virtual function, *calculateArea ()* for class **shape** which acts as base class for **circle** and **square**. Print the area of the two shapes.	*CO1*	*PO1*	**06**
		b)	Develop a C++ program to copy content of one file to another file until end of file is reached and display the copied content to the output screen.	*CO3*	*PO3*	**08**
		c)	Illustrate with a program, pointers of base class to access derived class methods.	*CO1*	*PO1*	**06**
			UNIT - V			
	9	a)	Write a C++ program that demonstrates the handling of exceptions in an inheritance tree. Create a base class called "**Mother**" and a derived class called "**Daughter**" which extends the base class. In the Mother class, implement a constructor that takes the age and throws the exception *InvalidAge( )* when the input age<0. In the Daughter class, implement a constructor that checks both the mother and daughter's age and throws an exception if the daughter's age is greater than or equal to the mother's age.	*CO3*	*PO3*	**10**
		b)	Write a program using function template to rearrange an array element of numbers in ascending order for integer array elements and floating-point array elements.	*CO1*	*PO1*	**10**
			OR			
	10	a)	Assume that you have developed an application that has a Class **Employee** with Eid, Ename and Esalary as member variables. Write read method that accepts values, throws user defined exceptions when the salary entered is less than 0. Write a C++ program to demonstrate the same.	*CO3*	*PO3*	**10**
		b)	Write a C++ program to create a class template "**Stack**" for illustrating a stack of integers and float values and print the total number of elements in the stack. Implement the push, pop, and display functions to insert elements to the stack, delete to remove the element from the stack, and display the elements of the stack.	*CO2*	*PO2*	**10**

****