| U.S.N. | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|

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

**Autonomous Institute Affiliated to VTU**

## August 2024 Semester End Main Examinations

**Programme: B.E.** **Semester: III**
**Branch: Information Science and Engineering** **Duration: 3 hrs.**
**Course Code: 22IS3PCOOP** **Max Marks: 100**
**Course: Object oriented Programming using 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

1 a) Write a function using reference variables as arguments to swap the values of a pair of integers. **06**

b) Develop a C++ program that creates a vector of size M using new operator. **06**

c) Create a class called Employee that has emp_code and emp_name as data members. Member function get_data() to accept the data and member function display () to display the employee details. Write a main() function to create class EMP with an array of employee objects. Accept and Display the details for N employees. **08**

### UNIT - II

2 a) Illustrate the usage of default arguments in C++. **06**

b) Define function overloading. Develop a C++ program to define three overloaded functions to swap two integers, swap two floats and swap two doubles. **06**

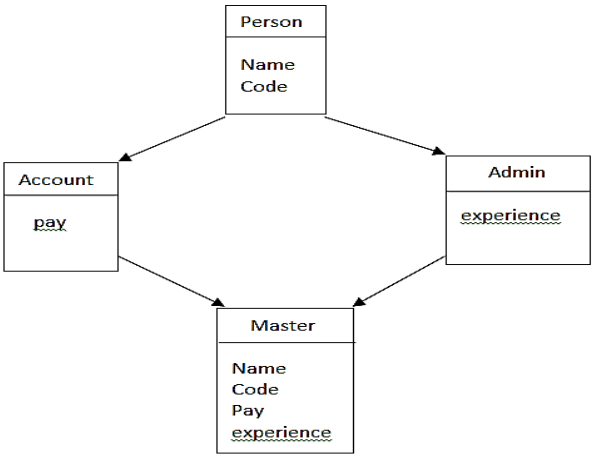
c) Consider a person object with the name of the person, age and city. Write a C++ program to accept the details using parameterized constructor. Also show the usage of default constructor and copy constructor appropriately. **08**

### UNIT - III

3 a) Demonstrate how pre-increment and post-increment operators can be overloaded. **06**

b) Illustrate how the ambiguities that arise in multiple inheritance can be resolved. **06**

c) Develop a C++ program to create a network of classes as given in the figure. The class **MASTER** derives information from both **ACCOUNT** and **ADMIN** classes which in turn derives information from the class PERSON. Implement the following operations : **08**
(i) Display the information contained in MASTER objects.
(ii) Increase the pay of master objects by Rs.5000 if the experience is more than 20 years.




**OR**

| | | | |
|---|---|---|---|
| 4 | a) | Write a C++ program to calculate the percentage of a student using multi-level inheritance. Accept the marks of three subjects in Base class. A class will be derived from the above mentioned class which includes a function to find the total marks obtained and another class derived from this class which calculates and displays the percentage of student. | **06** |
| | b) | Develop a C++ program illustrating how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named alpha, beta and gamma such that alpha, beta are base classes and gamma is derived class inheriting alpha and beta. | **06** |
| | c) | Develop a C++ program to add two complex numbers by overloading the '+' operator. Also show the difference by overloading '+' operator using friend function. | **08** |

**UNIT - IV**

| | | | |
|---|---|---|---|
| 5 | a) | Differentiate between Static binding and Dynamic binding. | **06** |
| | b) | Demonstrate the working of pointer to objects with a suitable example. | **06** |
| | c) | Write a program that creates a text file by reading the data for the students from the terminal. The data of each student consists of Roll number, name and marks. Again open the same file to display the student information. | **08** |

**OR**

| | | | |
|---|---|---|---|
| 6 | a) | Discuss the significance and the rules for virtual functions. | **06** |
| | b) | Illustrate how function overriding works in C++. | **08** |
| | c) | Explain fill(), width() and precision() functions with suitable examples. | **06** |

**UNIT - V**

| | | | |
|---|---|---|---|
| 7 | a) | Write a C++ program to illustrate catching all exceptions. | **06** |
| | b) | Write a Class template to sort an array of integers and floats using Bubble sort technique. | **08** |
| | c) | Explain the components of Standard Template Library. | **06** |

**********