Department of Information Science and Engineering

LAB PROGRAMS (AY 2022-23)

COURSE NAME: Object Oriented Programming using C++

COURSE CODE: 19IS3PCOOP

SEM : III

1. **Simple Classes and Objects:** Create an EMPLOYEE class with following members: Employee Number, Employee Name, Basic, DA, IT, Net Salary. Member functions: to read the data, to calculate Net Salary and to print data members. Develop C++ program to demonstrate reading N employees information and compute Net Salary of each employee. (Consider Dearness Allowance (DA) = 52% of Basic and Income Tax (IT) = 30% of the gross salary. Net Salary = Basic + DA - IT).

2. **Array of objects**: Develop a C++ program to demonstrate array of objects in class Student. Create appropriate member variables and functions to read and display students' USN and marks of six courses in three tests. Also calculate the average marks for each course taking best two of three. Test the program for N students.

3. **Polymorphism:** Develop a C++ program to create a class called COMPLEX with the following overloading functions ADD that return a COMPLEX number.

   i. ADD (a, s2) – where a is an integer (real part) and s2 is a complex number.

   ii. ADD (s1, s2) – where s1 and s2 are complex numbers.

4. **Friend function:** Create two classes, Class Roll Royce and Class Lamborghini with appropriate member variables (Model name, year, price and mileage) and member functions (read and print). Demonstrate the comparison of two cars w.r.t price and mileage using friend function.

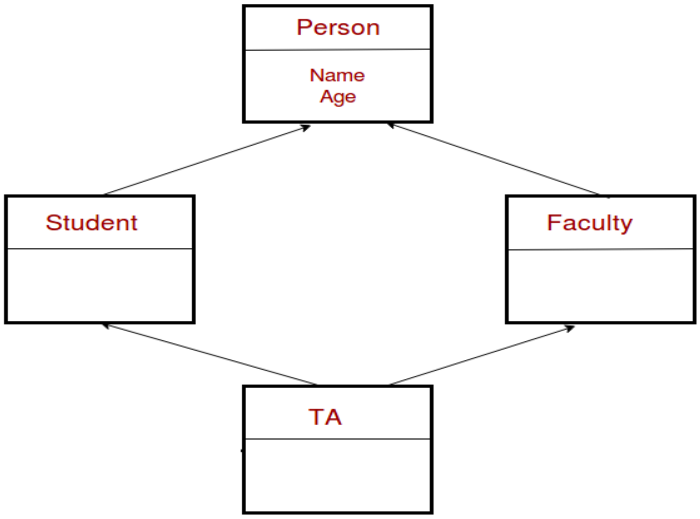
5. **Static and non-static members:** Create a vehicle having a non-static data member registration number and a static data member count. Non-static member functions setregno() and getregno() are used to get and set the registration number. A static member function getVehiclecount() is used to return the number of vehicles in the garage. Use a constructor to increment the vehicle count when a vehicle is created and the destructor to decrement the count when the vehicle is destroyed.

6. **Operator Overloading:** Develop a C++ program to create a class called MATRIX using a two-dimensional array of integers. Illustrate == operator overloading which checks the

compatibility of two matrices M1 and M2 for addition and subtraction. Find the sum and difference of matrices by overloading the operators + and – respectively. Display the results (sum matrix M3 and difference matrix M4).

7. **Inheritance and Constructors:** Design a class named PersonData with the following member variables: lastName, firstName, address, city, state and phone. Write the appropriate constructor, accessor and mutator functions. Next, design a class named CustomerData, which is derived from the PersonData class. The CustomerData class has member variables: customerNumber and email id. The customerNumber variable holds a unique integer for each customer. Write appropriate constructors, accessor and mutator functions. Demonstrate an object of the CustomerData class in retrieving individual customer data.

8. **Inheritance:** Implement the following relationship using appropriate member variables and member functions.



9. **Overriding functions:** Design a class called Ship with member variables name and year with appropriate constructors, accessors and mutators.

   - Illustrate virtual **print** function that displays the ship's **name** and the **year** it was built.
   - Design a **CruiseShip** class that is derived from the Ship class. The CruiseShip class has the member variable that holds maximum number of **passengers** and appropriate accessors, mutators and constructor. Create an overriding **print** function. The CruiseShip class's **print function should display only the ship's name and the maximum number of passengers.**
   - Design a **CargoShip** class that is derived from the Ship class. The CargoShip class should have the following members: a member variable for the **cargo capacity**, constructor, accessors and mutators. Define a print function that overrides the print function in the base class. The **CargoShip class's print function should display only the ship's name and the ship's cargo capacity.** Demonstrate the classes in a program that has an **array of Ship pointers**. The array elements should be initialized with the addresses of dynamically

allocated Ship. The program should then step through the array, calling each object's print function.

10. **Templates:**

    a) Illustrate class templates in a C++ program for the following operations: Adding two arrays, finding the max and min in an array.
    b) Develop a C++ program to sort using bubble sort by applying function templates.

11. **Exception Handling:** Demonstrate exceptional handling in C++ program by creating a class called EXPRESSION. Using appropriate member functions to accept a valid arithmetic expression. Evaluate the expression and divide the result by another variable 'd'. The programs should throw an exception "divide by zero" if the value of d is 0 and allows the users to read the 'd' value again. Rethrow the exception if d still contains 0.

12. **File Handling:** Develop a C++ program that reads a file containing real numbers and displays the minimum, maximum and average of the numbers. The program should accept the name of the file from the user.

Faculty in charge: Dr. VSR, RIS