KLS Gogte Institute Of Technology

Udyambag Belagavi -590008

Department of computer Science & Engineering

and

Department of Information Science & Engineering.

List of Experiments for Object Oriented Programming with C++

Course Code: 18ACS/IS28

Academic Year:  2019-2020

1. Program on class and object
2. Program illustrating the use of reference types in C++
3. Program on function overloading.
4. Program on dynamic memory management in C++
5. Program on array of objects.
6. Program on constructors and destructors.
7. Program on operator overloading.
8. Program on overloading i/o operators.
9. Program on inheritance.
10. Program on friend functions and virtual functions.
11. Program on dynamic polymorphism.
12. Program on file stream.

**1) Program on class and object**

1.1) Develop a C++ program consisting of a class called Employee with name, ID, department and basic salary as data members. Include member functions to:

* 1. set values for data members
  2. compute gross salary where DA=70% of basic, HRA=20% of basic and deductions=5% of basic    and
  3. Display details of an employee

1.2) Create a menu driven C++ application that stores price list of 5 items. It also provides a facility to print the largest price among all the elements. It also computes the sum of all prices using class and objects in C++.

1.3) Create a class to represent a bank account. Include the following members:

Data members

            a. name of the depositor

            b. account number

            c. balance amount

Member functions

            a. To assign initial values

            b. To deposit an amount

            c. To withdraw an amount after checking the balance

            d. To display details of the account

Write a menu driven main function that calls the member functions until the user chooses to terminate the program.

1.4) Develop a C++ program with a class called car having data members brand, model and year. Write the appropriate member functions to set the values of the data members using scope resolution operator and also display the details of the car using objects.

1. **Program illustrating the use of reference types in C++**

2.1) Write a C++ program to swap two variables using reference variables.

2.2) Write a C++ function power(int, int, int &) that takes two integer variables (say a and b) and an integer reference variable as parameter, computes a ^ b and stores the result in the reference variable. Write the associated main() to call this function.

2.3) Consider n students’ records, each consisting of name, USN and marks in three tests and average. Develop a C++ program to read name, USN and marks in three subjects. Write a function that updates the average field (using reference parameter) by computing average of best two marks. Display the updated student records.

2.4) Write a function exchange(int &, int &, int &) that uses reference variables to cyclically exchange values contained in three variables. Write the associated main to exercise this function.

1. **Program on function overloading**

3.1) Write a C++ program to calculate the area of circle (given the radius), rectangle (given the two sides) and triangle (given the three sides) using function overloading.

3.2) Develop a menu driven C++ program with overloaded functions to sort arrays of integers, floats and characters.

3.3) Write a C++ program to calculate maturity value of investment using function overloading. One function computes the maturity value using simple interest formula and the other function computes the maturity value using compound interest formula.

Formula for maturity value using simple interest calculation: p + (p \* t \* r / 100)

Formula for maturity value using compound interest: p((1 + r / n) ^ (n \* t))

where,

    p is the principal amount

    r is the rate of interest

    t is the time duration of investment

    n is the number of times interest is compounded per year

3.4)  Write the following overloaded functions:

    i. strcopy(s1, s2) that copies all the characters of s2 to s1

    ii. strcopy(s1, s2, n) that copies the first n characters of s2 to s1

   Write the associated main to exercise these functions.

1. **Program on dynamic memory management in C++**

4.1) Create an array using dynamic memory allocation. Write functions to perform the following:

            a. Find the minimum element in the array

            b. Find the maximum element in the array

            c. Find the mean of the elements in the array

 4.2) Develop a C++ program to illustrate dynamic allocation and deallocation of memory using new and delete operators for an array of n integers. Include functions to

a. read the elements

b. find the number of even and odd integers in the array and

c. display the elements

 4.3) Create a C++ application that receives the test scores from the user. The application computes the average of test scores. It also displays the scores in ascending order. The application does dynamic allocation of memory for storing the scores.

4.4) Develop a C++ program to read records of n items as per the following structure:

struct Item{

            int itemCode;

            string name;

            float price;

};

Create the array of structures using dynamic memory allocation. Write functions

a.    to read the records and

b.    display item having highest price.

**5) Program on array of objects**

5.1) Create a Book class containing data members viz., Book number, Title, Author and price. Write the main function that does the following:

            a. Creates an array of book objects and reads the book information using member function.

            b. Given the book number, searches and prints using friend function, the book details if found, an error message otherwise.

5.2) Develop a C++ program consisting of an array of n Movie objects with each Movie having name, year of release and rating as data members.

Include member functions to:

1. set values for data members
2. display movie details

Using separate friend functions:

1. search for a particular movie based on name and
2. display top performing movie based on rating (Assume that no two movies have the same rating).

5.3) Create a class called User that has name and age as data members. The class has member functions to set values for data members and display the details of Users. In main, create an array of objects of User class and invoke member functions to read and display. Write a friend function that counts and returns the number of users whose age is less than 18.

5.4) Create a class called Course that has following data members:

            courseName, CourseCode, Faculty, numStudRegistered

Write member function to read data members and a friend function that returns the course details where maximum registrations are done.

Create an array of Course class and exercise the above functions.

**6) Program on constructors and destructors**

6.1) Create a class called intArray with

Data members:

1. pointer to an integer array and
2. integer to hold the array length

Member functions:

1. A zero-arg constructor
2. A parameterized constructor with an array and its length as parameters
3. A copy constructor
4. Display array elements
5. Destructor

Write the corresponding main()

6.2) Create a C++ class for football player object with the following attributes: player no., name, number of matches and number of goals done in each match. The number of matches varies for each player. Write parameterized constructor which initializes player no., name, number of subjects and creates array for number of goals and number of matches dynamically. Write a destructor for the class player.

6.3) Create a class called List that uses constructor to create a list of n elements. Write member functions to read and display the List object. Write destructor to free up the memory allocated.

6.4) Create a class called Employee with id, name, designation, salary as the data members. Write the following methods/constructors with the main function.

1. Default constructor to initialize the values for manager (employee)
2. Parameterized constructor to initialize the values of employees with other designations.
3. A display\_detail function to display details of the employee.
4. Usage of destructor.

**7) Program on operator overloading**

7.1) Create a class called Complex with real and imaginary parts as data members.

Member functions:

1. A zero-argument constructor
2. A parameterized constructor
3. Overloaded + operator to add two complex numbers and return the sum
4. Display complex number

Write the corresponding main()

7.2) Create a class to represent Currency. The class has two instance variables: rupees and paise. Write default constructor and parameterized constructor to initialize a Currency object. Write member functions to add two Currency objects by overloading the addition operator and to display the currency.

7.3) Create a class to represent a shape named Box. Include the following members: length, breadth and height.

It includes the member functions to set length, breadth and height. Compute the volume of two boxes. Overload + operator to add two Box objects and store the result in Box3.

7.4) Design an invoice class of items with array of price as the data member. Overload ‘&’ operator to add the item price to the array and similarly overload ‘ ! ’ operator to remove an item’s price from the array. Exercise the same using the main function.

**8) Program on overloading i/o operators**

8.1) Create a class called Distance with feet and inches as data members.

Member functions:

1. A zero-argument constructor
2. A parameterized constructor

Friend functions:

1. Overloaded extraction operator to read a distance
2. Overloaded insertion operator to display a distance (Ex - 10 feet 5 inches)
3. Compare two distances

Write the corresponding main()

8.2) Create a class called Time having hours, minutes and seconds as data members. Overload extraction operator (>>) to read a Time object and insertion operator(<<) to display a Time object in the format: "hh:mm:ss".

8.3) Create a class called complex. It has data members as real and imag. It reads real and imag values by overloading << operator. It also displays the value of the complex object by overloading >> operator.

8.4) Create a class player (cricket) with player\_name , date of birth and ranking as data members. Read the array of values for the data members mentioned. Display the player details in Ascending order of ranking by overloading ‘>>’ operator and in descending order of date of birth by overloading ‘<<’ operator. Exercise the same using the main function.

**9) Program on inheritance**

9.1) Create a class called Manager with attributes: name, ID and basic salary. Demonstrate inheritance by deriving classes HR Manager and Sales Manager from Manager and compute gross salary as per the following:

  HR Manager - DA = 70% of basic, HRA = 20% of basic, deductions = 5% of basic

  Sales Manager - DA = 70% of basic, HRA = 10% of basic, TA = 5% of basic,

deductions = 5% of basic

9.2) Implement the following class hierarchy. Assume suitable data members.

Employee

computeSalary()

printDetails()

SalesManager

computeSalary()

printDetails()

ProjectManager

computeSalary()

printDetails()

9.3) Implement a BANK class to demonstrate the inheritance in Java by implementing getRateOfInterest member function for three different banks, as shown below.

Bank

getRateOfInterest()

SBI

getRateOfInterest()

ICICI

getRateOfInterest()

AXIS

getRateOfInterest()

* 1. The class Cylinder inherits a superclass Circle. It further defines a variable called height, a method computeVolume() and its own constructors. Implement the hierarchy as shown below:

Circle

radius:double

Circle()

Circle(radius:double)

computeArea():double

Cylinder

height:double

Cylinder()

Cylinder(height:double)

Cylinder(height:double, radius:double)

computeVolume():double

**10) Program on virtual functions**

10.1) Create a base class called Student and derived classes UGStudent and PGStudent. Demonstrate polymorphism using virtual functions to compute the total score as follows:

    UGStudent - average of best two scores

    PGStudent - addition of two scores

10.2) Class Employee contains virtual functions namely, raiseSalary() and promote(). Classes Engineer and Manager inherit the Employee class and provide implementations for the virtual functions. Write associated main() that exercises the two derived classes.

10.3) Create three classes Person, Professor and Student. The class Person should have data members name and age. The classes Professor and Student should inherit from the class Person. The class Professor should have two integer members: publications and cur\_id. There will be two member functions: getdata and putdata. The function getdata should get the input from the user: the name, age and publications of the professor. The function putdata should print the name, age, publications and the cur\_id of the professor.

The class Student should have two data members: marks, which is an array of 3 integers and cur\_id. It has two member functions: getdata and putdata. The function getdata should get the input from the user: the name, age, and the marks of the student in  3 subjects. The function putdata should print the name, age, sum of the marks and the cur\_id of the student.

For each object being created of the Professor or the Student class, sequential id's should be assigned to them starting from 1.

Solve this problem using virtual functions, constructors and static variables.

10.4) Implement the following inheritance hierarchy to demonstrate pure virtual functions.

Circle

computeArea()

Triangle

computeArea()

Rectangle

computeArea()

Shape

computeArea()

x, y

**11) Program on dynamic polymorphism**

11.1) Create a base class called List that has virtual functions: CreateList and DispList. Create two classes IntList and CharList that inherit from List. Both classes override CreateList and DispList. The IntList class has an additional member function to return the sum of elements of the list of integers. The CharList has an additional member function to search for a character received as a parameter and returns the position of it’s first occurrence, if found in the list of characters; else returns -1. Demonstrate dynamic polymorphism by creating and upcasting objects of the two derived types in main.

11.2) Implement the following inheritance hierarchy and demonstrate runtime polymorphism. Display appropriate role in the displayRole() function.

Doctor

displayRole()

Psychiatrist

displayRole()

Nephrologist

displayRole()

Pediatrician

displayRole()

Cardiologist

displayRole()

11.3) Consider an example of book shop which sells books and video CDs. These two classes are inherited from the base class called BookShop. The BookShop class has data members such as title and price. The Book class has data members for storing number of pages in a book and the VideoCD class has playing time in minutes. Each class will have member functions such as read() and show(). In the base class, these members have to be defined as virtual functions. Write a program which models the class hierarchy for book shop and processes objects of these classes using pointers to base class.

11.4) Write a C++ program to create one superclass Animal and three subclasses, Herbivores, Carnivores and Omnivores. Subclasses inherits from the superclass Animal and override its eat() method. Demonstrate dynamic polymorphism to print data of Herbivores, Carnivores and Omnivores by upcasting objects of the three derived types in main.

**12) Program on file streams**

12.1) Write a program that creates a file by reading and storing user input text. The program further reads the file and creates another file that contains all the text of the input file converted into uppercase.

12.2) Write a C++ program to copy the contents of two files (first followed by second) into a third file.

12.3) Write a C++ program to count the number of alphabets, digits and other characters contained in a file.

12.4) Write a C++ program to encrypt the contents of a file using substitution cipher. In a Substitution cipher, any character of plain text from the given fixed set of characters is substituted by some other character from the same set depending on a key. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. Accept the key from the user to perform encryption.