

2022
(July)
MCA
Paper Code: MCA-0816
(Object Oriented Programming and C++ Lab)
Full Marks: 40
Time: 3 Hours

The figures in the margin indicate full marks for the questions

Answer *any two* from the following questions

1. Design and test two classes namely **Number** and **Evaluate**. The Evaluate class is a friend of the Number class. The Number class will contain two floating point data members namely *a* and *b*. The class will also contain the following member functions: (2)
 - i. A parameterized constructor which initializes the data members (1)
 - ii. *get* and *set* methods for each data member (4)
 - iii. A friend function *ShowData()* that can display all the data members (1)

The Evaluate class will be responsible for performing operations on the Number class and hence it contains three functions as follows:

 - i. A function *CubeSum()* to compute the sum of cubes, (4)
i.e., $a^3 + b^3 = (a + b)(a^2 - ab + b^2)$
 - ii. A function *CubeDiff()* to compute the difference of cubes, (4)
i.e., $a^3 - b^3 = (a - b)(a^2 + ab + b^2)$
 - iii. A function *SquareDiff()* to compute the difference of squares, (4)
i.e., $a^2 - b^2 = (a + b)(a - b)$
2. Create a class template for a class named **GeneralPriorityQ** to implement priority queue. The class should contain the following members:
 - i. Data members (3)
 - a. A variable to track the size of an array,
 - b. Two pointers (for dynamic memory allocation), one for data item and the other for priority,
 - c. Front of the Queue index,
 - d. Rear of the Queue index.
 - ii. Member functions
 - a. *enqueue(item, priority)* to insert an element *item* into the priority queue by specifying the priority, (4)
 - b. *getHighestPriority()* for returning the highest priority item, (3)
 - c. *deleteHighestPriority()* for removing the highest priority item, (4)
 - d. *printPQueue()* to print the elements in the priority queue. (3)
 - iii. Exception handling for the following errors (1.5+1.5=3)
 - a. Priority queue full
 - b. Priority queue empty.

Use this class template in a program to create integer, character and floating point GeneralPriorityQ objects.

3. Define and test a class **SET** with two data members: an array of integers and its size. It should also include the member functions: (2)
 - a. To read elements of a SET object. (3)
 - b. To print elements of a SET object. (3)
 - c. To find union of two SET objects by overloading + operator. (4)
(Hint: $S3 = S1 + S2$)
 - d. To find intersection of two SET objects by overloading * operator. (4)
(Hint: $S4 = S1 * S2$)
 - e. To find difference of two SET objects by overloading - operator. (4)
(Hint: $S5 = S1 - S2$)

4. Create a base class called **Shape** with two data members- *radius* and *height* of double data types. (2)
 - a. Provide constructors and member function *setData()* to initialize and set the data members of **Shape** class respectively. Provide another member function *calcVolume()* which is a pure virtual function to compute the volume of shapes. (5)
 - b. Derive three classes called **Cylinder**, **Cone** and **Sphere** from the base class **Shape**. (6)
 - c. Using pointers to **Cylinder**, **Cone** and **Sphere** objects, design a program that will accept dimensions of a **Cylinder**, **Cone** and **Sphere**. Declare an array of pointers to the base class **Shape**. Place these objects in the array of pointers pointing to the base class **Shape** and demonstrate runtime polymorphism by displaying the volumes of the shapes. (7)

Evaluation Criteria:

Logic of the solution – 30%

Debugging skill – 20%

Syntax and Completion – 25%

Overall impact-25%

