

**SUBJECT : 13.403 OBJECT ORIENTED TECHNIQUES (FR)**

**MODEL QUESTION PAPER**

**PART A**

All questions are compulsory. Each question carries 4 marks

1. Distinguish between Inline functions and macros.
2. What is the purpose of copy constructor? Explain with an example.
3. When do we use the keyword **protected** with a data member?
4. Explain syntax of class template with an example showing class definition and object definition.
5. Distinguish between overloading and overriding.

(5 \* 4 marks = 20 marks)

**PART B**

Answer one question from each module. Each question carries 20 marks.

**MODULE I**

6. (a) Write a function called **Smaller()** that is passed with two int arguments by reference and then sets the smaller of the two arguments to 0. Write a **main()** program to test this function. (12)

(b) Distinguish between the following terms:

- (i) Inheritance and polymorphism (ii) Data abstraction and data encapsulation (8)

7. (a) Write an overloaded function **power()** to raise a number **m** to to **n**. The function takes **int** type as well as **double** type value for **m** and an int type value for n. Use a default value of 2 for n to make the function to calculate squares when this argument is omitted. Write a main program that takes the values of m and n from the user to test the function. (12)

(b) Distinguish between the following terms:

- (i) Objects and classes (ii) Dynamic binding and message passing (8)

**MODULE II**

8. (a) Write a class to represent a vector (a series of float values). Include member functions to perform the following tasks:
  - (i) To create the vector
  - (ii) Modify the value of a given element
  - (iii) Multiply by a scalar value
  - (iv) Display the vector

Write a main program to test your class. (14)

(b) When do we declare a member of a class static? (6)

9. (a) Define a class `String` that could work as a user defined string type. Include constructors that will enable to create an uninitialized string object and an object with a string constant initialized a member function to concatenate two strings and another member function to display string objects. Write a main program to test your class. (14)

(b) What is a friend function? What are the merits and demerits of using friend functions? (6)

### MODULE III

10. Write a C++ program to create a class called **Counter** with the following functionalities.

(i) Automatic Counter initialization (ii) Incrementing counter using ++ operator (use friend function)

(iii) Decrement using -- operator (iv) Comparison for equality using ==

(v) Display count value using operator <<.

Write a main program to test this class. (20)

11. Create a base class called `shape`. Use this class to store two double type values that could be used to compute the area of figures. Derive four specific classes called `triangle`, `rectangle`, `square` and `circle` from the base class `shape`. Add to the base class, a member function `get_data( )` to initialize base class data members and another member function `display_area ( )` to compute and display the area of figures. Make `display_area ( )` as a virtual function and redefine this function in the derived class to suit their requirements. Using these five classes design a program that will accept dimensions of a triangle, square or circle interactively and display the area. (20)

### MODULE IV

12. (a) Write a C++ program to implement a generic **stack** class. Include a default constructor, destructor and member functions **push( )** to put values into the **stack**, **pop( )** to remove values from **stack**, **isempty( )** and **isfull( )** for testing whether stack is empty or full. Use array implementation of stack. Write a main program to test your class for integer and character type data. (14)

(b) What is a stream? How is **cout** able to display various types of data without any special instructions? (6)

