H Roll No. .....

TBC–203/TBI–203/TBS–203

B. C. A./B. Sc. (IT)/B. Sc. (CS)

(SECOND SEMESTER)

END SEMESTER

EXAMINATION, June, 2023

OBJECT ORIENTED PROGRAMMING USING C++

Time : Three Hours

Maximum Marks : 100

Note : (i) All questions are compulsory.

(ii) Answer any ***two*** sub-questions among (a), (b) and (c) in each main question.

(iii) Total marks in each main question are **twenty**.

(iv) Each sub-question carries 10 marks.

(a) State the important features of object-oriented programming. Compare the object-oriented programming with procedure-oriented programming. (CO1)

P. T. O.

(b) Write short notes on the following with example : (CO1)

(i) Message passing

(ii) Access specifiers

(iii) Encapsulation

(iv) Scope resolution operator

(c) Explain the role of abstraction in object languages. Compare the features of Classes and Structure in C++. (CO1)

2. (a) Describe, what do you mean by nesting of classes. Also explain briefly how Friend Function is important in C++. (CO2)

(b) What happens if we have the following two functions ?

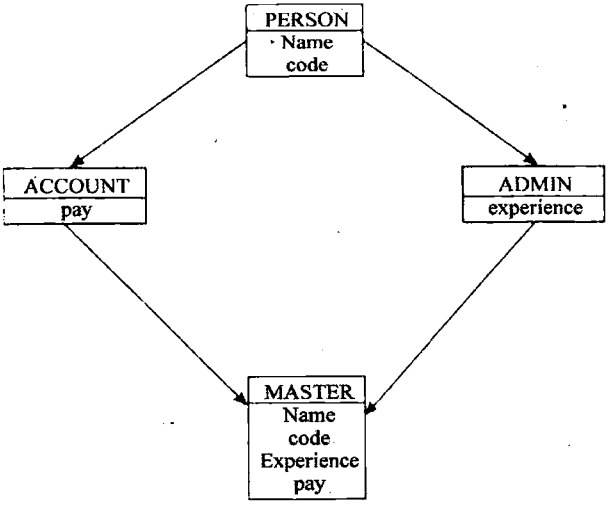
Int Area (int width, int length = 1);

Int Area (int Size);

Will these overloads? There are a different number of parameters, but the first one has a default value. (CO2)

(c) Define inline functions. Mention the situations where declaration of a function as inline is not recommended. Write a C++ program with an inline function to add 3 numbers. (CO2)

3. (a) A shopkeeper wants to maintain the stock database category wise (electronics appliances, food items, clothing, milk product, kitchen product). Specify all the classes and functions as per the relation between different products. (CO3)



Does this type inheritance lead to ambiguity? If yes, discuss by putting up a suitable scenario how it can be removed.

P. T. O.

(b) Consider the following C++ declaration and answer the questions given below : (CO3)

```
class Alpha
{
      protected:
int x, y;
 public:
void PutValA();
 void GetValA();
};
class Beta: private Alpha
{
 int m, n;
void PutValB();
public:
 void GetValB();
class Gamma: protected Beta
{
     int a:
     public:
        void GetData();
        void ShowData();
};
```

(i) Write names of member functions, which are accessible from the objects of class Gamma.

(ii) Write the name of the data members, which are accessible from the member function of class Beta.

(iii) Name the base class and the derived class of class Beta.

(iv) Name the base class and derived class of class Gamma.

(c) Create a class named complex with two members real and imaginary of float data type. Show how we can perform the following operation : (CO3)

$$C_3 = C_1 + C_2$$

using operator overloading.

4. (a) Justify the following statement as a suitable example : (CO4)

"In a class hierarchy of several levels, if we want a function at any level to be called through a base class pointer, then the function must be declared as virtual in the base class."

(b) (i) What is Abstract Class ? What is the significance of pure virtual function in a class ? How are they different from normal functions ? (CO4)

(ii) Write a program to illustrate runtime polymorphism using virtual function.

(c) With the help of an example program, differentiate between the following :(CO4)

(i) Overloading *vs.* Overriding

(ii) Early binding *vs.* Late binding

5. (a) What is Template ? Differentiate between function template and class template. Write a function template for finding the maximum value in an array. (CO5)

(b) What is Stream ? Explain the hierarchy of C++ Stream classes. Write a C++ program to get employee id and salary of an employee of a Water Treatment Chemical Company from the user and store these details into a file called 'employee.dat'. Also read the details from the file and display the data. (CO5)

(c) What are exceptions ? How is an exception handled in C++ ? What is the need of exception handling ? Write a program that throws an arithmetic exception as and when a number input is a negative number. (CO5)

OR

What is STL? What are its components ? Differentiate between Map and Vector.