Ref No: EX/IT/PC/B/T/214/2024 (S)

B.E in INFORMATION TECHNOLOGY

2ND YEAR, 1ST SEMESTER SUPPLEMENTARY EXAMINATION, 2024

OBJECT ORIENTED PROGRAMMING

Time: 3 hours **Full Marks:** 100

<table>
<tr><td>CO1

[20 MARKS]</td><td>1. a) Assume a function with signature void sum(int a, int b, int c). Show three proper ways of supplying default parameter values in the given prototype and their corresponding function call statements.

b) Distinguish among each of the following triples. (Any 3)
i) malloc() vs. calloc() vs. new
ii) Local variables vs. reference variables vs. pointer variables
iii) Array vs. structure vs. class
iv) Normal function vs. inline function vs. macro

c) Write suitable code to execute the following instructions as specified in the main() method. Predict the output with proper reasons.

int main()
{
int a=40, b=50;
min(a,b)=-1;
cout<<“a=”<<a<<“b=”<<b;
swap(&a, &b);
min(a,b)=-2;
cout<<“a=”<<a<<“b=”<<b;
}

[5+(3+3+3)+6=20]</td></tr>
<tr><td>CO2

[20 MARKS]</td><td>2. a) Assume a class Money as the following definition. Now complete the class definition in order to properly execute the statements in the main() method. Finally discuss the output.

class Money
{
int rupees, paise;
};

int main()
{
Money m1(450,90), m2(500,70),m4;
Money m3=m1;
m3.add(m2);
m3.show();
m4=subtract(m2,m1);
m4.show();
}

Or,
Consider the following code snippet having the definition of a class Number. Keeping the class definition unchanged, define all the member methods and constructor outside the class definition and show how they are invoked from main() method. Discuss the outputs. Point out the drawback of the code and state how this drawback can be overcome.</td></tr>
</table>

	class Number *{* *int a;* *public:* *Number(int=0);* *friend void swap(Number, Number);* *void add(Number);* *void display();* *};* b) Justify the truth/falsity of each of the following statements. Provide suitable arguments in support of your answers along with suitable code snippets where/if required. ***(Any 4)*** i) Two functions having prototypes ***void sum(int, int)*** and ***int sum(int, int)*** cannot be overloaded. ii) A friend function hampers the data abstraction property of OOP. iii) A copy constructor of a class takes the photocopy of object of that class. iv) A non-static member method of a class can access only the non-static member variables of that class. v) A constant member method of a class can only read the class member variables, but can't modify them. [8+(3x4)=20]
CO3 **[20 MARKS]**	3. a) Suppose a class ***Employee*** has a private member variable ***basic_sal*** and a member function ***get_sal().*** Two subclasses ***Manager*** and ***Clerk*** have been derived from it. These two types of employees receive 40% and 30% allowances respectively on their basic monthly salary. Show with a suitable code how to compute the total salary of these two types of employees. What type of inheritance does it refer? b) Distinguish between the following pairs: ***(Any 4)*** i) Derived class and friend class ii) Normal virtual function and pure virtual function iii) Normal destructor and virtual destructor iv) Early binding and late binding v) Multiple inheritance and hybrid inheritance [(7+1)+(3x4)=20]
CO4 **[20 MARKS]**	4. a) Consider the following ***Sample*** class. Complete it with appropriate code to perform the operations as directed in the ***main()*** function. Discuss the output. *class Sample* *{ int s;* *//define suitable constructors and functions* *};* *int main()* *{Sample s1(10), s2(5),s3;* *s1+=s2;* *s3=s1*s2;* *int x=s1;* *int y=s2;* *Sample s4(x+y);* *cout<<s4;* *Sample s4=s3+s4;* *cout<<s4;* *}*

	b) Fill up the blanks with appropriate phrases. Hence justify the validity of each of the completed sentences. (Use code snippets if/where required for justification) i) For [ ] operator overloaded function, it is better to return _____ because _____. ii) For overloading input and output operators, we always have to use _____ functions because _____. iii) To distinguish between pre-increment (++ob) and post-increment (ob++) operator overloading, we have to _____. iv) For converting a basic data type to an object of a class, we need _____. However, for the reverse we need _____. [8+(3x4)=20]
CO5 **[20 MARKS]**	5. a) Answer the following questions in brief. Provide suitable code snippets where required. ***(Any 5)*** i) If we are not sure about the exact type of exception being thrown, then how can we handle such scenario? ii) When and how an exception needs to be rethrown? iii) Why does the need of specialization of template arise? iv) What are the possible modes in which a file can be opened? State the significance of each of them. v) Show how a custom exception class object can be created, thrown and handled. vi) Discuss the functionalities of *tellp(), seekg()* functions along with their proper syntax. b) Write a complete C++ program to open a file and print its contents to the console. Or, How can you supply default parameter/s in template functions? Show with suitable code snippet. Illustrate non-type parameters in templates. [(3x5)+5=20]

Course outcomes:

CO1: Recognise and **illustrate** the procedural enhancements of object-oriented programming languages over procedural languages.

CO2: Explain, illustrate and **recognise** the basic features of classes and objects.

CO3: Illustrate the extended features of OOP (Inheritance, Polymorphism) and **apply** them in practical problem solving.

CO4: Explain and **illustrate** RTTI, Namespace and Operator overloading.

CO5: Demonstrate I/O, exception handling and generic programming.