†

For object T1 = 2 hours and 45 minutes
For object T2 = 3 hours and 30 minutes
For object T3 = 6 hours and 15 minutes

(b) Modify the class and rewrite the above program to display the total number of objects created by the class using a static member function. **[2]**

(c) Write down the output of the following code. **[2]**

```
#include <iostream>
using namespace std;
class item {
int a,b;
public:
item(int x,int y) {
a=x;
b=y;
}
void display() {
cout<<"a="<<a<<endl;
cout<<"b="<<b<<endl;
} };
int main(){
item A;
A=item(10,20);
A.display();
return 0;
}
```

End of Questions

MID SEMESTER EXAMINATION, April-2024

Object-Oriented Programming with C++
(CSE 3943)

Programme: B.Tech(CSIT) **Semester: 6th**
Full marks: 30 **Time: 2 hours**

Subject / Course Learning Outcome	*Taxonomy Level	Question Number	Marks
Design programs that show how to use the variables, data types, and operators with the basic concepts of Object-Oriented Programming.	L1, L2	1a, 1b, 1c, 2a, 2b	10
Understand the various concepts of functions such as return by reference, inline functions, default arguments and function overloading in C++.	L1, L2, L3	2c, 3b	4
Understand the concept of classes and objects in Object-Oriented Programming.	L1, L2	3a, 4a, 4b, 4c, 5b	10
Learn to apply the concept of constructors and destructors in the program.	L2, L3	3c, 5a, 5c	6
Demonstrate the use of operator overloading and type conversions in the development of the programs.	L2, L3		
Design programs that show how to use the concept of inheritance, pointers and virtual functions in C++.	L1, L2, L3		

*Bloom's taxonomy levels: Remembering (L1), Understanding (L2), Application (L3), Analysis (L4), Evaluation (L5), Creation (L6)

Answer all questions. Each questions carry equal marks.

1. (a) Distinguish between the following terms: **[2]**
   Data Abstraction and Encapsulation
   (b) Describe, with examples, the uses of enumeration data types. **[2]**
   (c) What are manipulators in C++? Write down the most commonly used manipulators with an example. **[2]**

2. (a) Explain the difference between constant pointer, pointer to a constant, and pointer constant to a constant with an example. **[2]**
   (b) Write down the difference between a pointer variable and a reference variable. **[2]**
   (c) Explain the concept of return by reference with an example. **[2]**

3. (a) Consider a shopping list of items for which a user can place an order with a dealer every month. The list includes details such as the item code number and price of each item. Different Operations which can be performed on the list are: **[2]**
   1. Adding an item
   2. Deleting an item
   3. Display total value of all items
   4. Display all items

   Write a C++ program using class and objects depicting the above scenario.
   (b) Write a program that defines a function that receives 4 arguments - char, int, float and double in that order. Make 3 calls to this function. In the second call, pass only 3 arguments; the last one should be taken as 6.28 by default. In the third call, pass only 2 arguments. In this case the third argument should be taken as 3.14 and fourth as 6.28 by default. **[2]**

   (c) Why is it important to distinguish between a default constructor and a default argument constructor? Explain with an example. **[2]**

4. (a) Define a class to represent a bank account. Include the following members: **[2]**

   Data members:
   1. Name of the depositor
   2. Account number
   3. Type of account (savings or current)
   4. Balance amount in the account

   Member functions:
   1. To assign initial values
   2. To deposit an amount
   3. To withdraw an amount after checking the balance
   4. To display name and balance

   Write a main program to test the program.
   (b) Modify the above program for handling 10 customers. **[2]**
   (c) Describe the mechanism of accessing data members and member functions of a class in the following cases: **[2]**
   (i) Inside the main function
   (ii) Inside a member function of the same class

5. (a) Create one class **Time**, which stores the value of time in hours and minutes format. Initialize the objects of the class by calling the constructor explicitly. The arguments of the constructor must be entered by the user. Write a program that adds one object with another object of class **Time**. Use two friend functions to carry out the addition operation and to display the result. The result display should be in the format below. Here, T1, T2 and T3 are the objects of class **Time**. **[2]**
   Sample Run: