AUTUMN END SEMESTER EXAMINATION-2015

3rd Semester B.Tech & B.Tech Dual Degree

OBJECT ORIENTED PROGRAMMING (IT-2001/IT-301)

(Regular-2014 & Back of Previous Admitted Batches)

Full Marks: 60 **Time: 3 Hours**

Answer any SIX questions including Question No.1 which is compulsory.

The figures in the margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.

1. Answer the following questions: [2 × 10

a) What is the difference between static and non static member of a class? Give example for each.

b) Can we pass class objects as function arguments? Explain with the help of an example.

c) What will be the output of the following code:

```
#include<iostream.h>
void main()
{
int i=0;
i=400*400/400;
cout<< i;
}
```

d) A friend function cannot be used to overload the assignment operator(=). Explain why?

e) Identify the error in the following code:

```
#include<iostream.h>
Class Space
{
int mcounter;
public:
Space()
{
mcounter=0;
}
Space operator ++()
{
mcounter++;
return Space(mcounter);
}
};
void main()
{
Space obj;
obj++;
}
```

f) How constructors and destructors are called in multiple and multilevel inheritance?

g) Differentiate between compile-time binding and run-time binding with example.

h) Consider this code segment:

```
class myclass {
public:
.....
        void show ();
.....
};
.....
int main () {
.....
myclass myobj;
myclass *ptr = &myobj;
}
```

Write an expression for invoking the member function show() of object myobj using the pointer ptr.

i) What is generic programming? How is it implemented in C++?

j) Explain the different types of throwing exceptions mechanisms allowed in C++.

2. a) Write a C++ program to implement the time class that contains three integer data members hour, minute and second and various member functions to perform the following task: [4

(i) To initialize data members.

(ii) To advance time of an existing object by a specified number of hours, minutes and seconds.

(iii) To reset the current time to initial time of the existing object.

(iv) To display the time in **hour : minute : second** format.

b) Discuss the different characteristics of OOP with examples in detail. [4

3. (a) Write a program to declare two classes Meter and Centimeter. Declare objects of both the classes. Convert Meter to Centimeter and vice versa. Perform the conversion using user defined conversion routines. [4

(b) Explain the syntax of binary operator overloading. How many arguments are required to perform binary operator overloading? [4

4. (a) Explain different types of constructors in C++ in details with appropriate syntax. [4

(b) An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in following figure A. The figure A also shows the minimum information required for each class. Specify all classes and define functions to create the database and retrieve individual information as and when required. [4

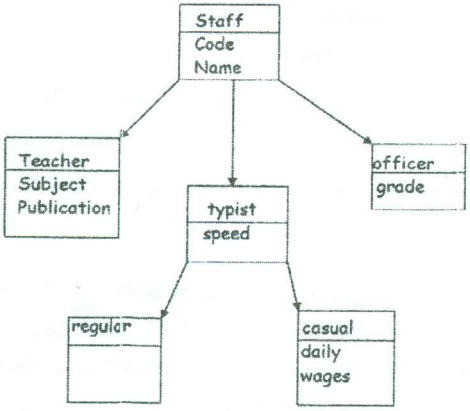

Figure A: Class Relationship

5. a) Write a program to display the details of employee (name, address) with higher salary using ***this*** operator. [4

   b) Write a program to swap the values of two data members using class template. [4

6. a) Describe briefly the C++ stream classes for I/O operations? Discuss the various forms of **get()** function supported by the input stream? [4

   b) What is the difference between opening a file with a constructor function and opening a file with open () function? When is one method preferred over the other? [4

7. a) Write a program that illustrates the application of multiple catch statements. [4

   b) Consider base class Base and derived class Derived. Assume bptr is a pointer to base class and dptr is a pointer to derived class. Differentiate between these pointers in terms of accessing the derived class object. [4

8. Write short notes on the following (any two) [4 × 2

   a) Object Delegation

   b) STL

   c) Pure Virtual Functions

X X X X X