## Third Semester Examination – 2007

OBJECT ORIENTED PROGRAMMING USING C++
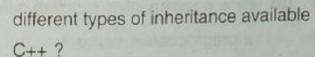
Full Marks – 70

Time : 3 Hours

*Answer Question No. 1 which is compulsory
and any five from the rest.*

*The figures in the right-hand margin
indicate marks.*

1. Answer the following questions :          2×10

   (a) What is a C++ template ?

   (b) What's the best way to declare and define global variables ?

   (c) What is inheritance and what are the different types of inheritance available in C++ ?

P.T.O.

(d) What is a constructor ? What is a default constructor ?

(e) What is the difference between char a[ ] = "string" ; and char *p = "string" ; ?

(f) How do you decide which integer type to use ?

(g) How do you declare an array of N pointers to functions returning pointers to functions returning pointers to characters ?

(h) What is "abstraction" and why do we use it ?

(i) What is an "invariant" ?

(j) What is the difference among **public, protected** and **private** members of a class ?

2. (a) How does an inline function differs from a preprocessor macro ? 2

(b) How is polymorphism is achieved at compile time and run time ? Give an example of a program that uses polymorphism. 3

(c) Define a class for a complex number. Write a program to read and print the complex number. 5

3. (a) The voltage gain of an amplifier is given by the formula

$$voltage\_gain = \left[ \frac{275}{\sqrt{23^2 + (0.5f)^2}} \right]^n$$

where f is the frequency in Hertz and n is the number of stages in the amplifier. Write a **complete C++ program** that asks the user to input values for n and f, calculates the value of the voltage gain using the formula and produces the following display on the terminal screen :

At a frequency of X hertz, the voltage gain is Y

where X is replaced by the frequency and Y is replaced by the voltage gain.

5

(b) Write a function that returns the mean of the n elements of an array of type double.

5

4. (a) List the three ways of passing a parameter (or return value) in C++. For each indicate :

- Whether the method makes a copy of the object passed.

- Whether the method allows passing of const objects.

- Whether the method supports poly-morphism and virtual dispatch.

IWL

- Whether the method allows implicit type conversions to be performed.

5

(b) Write a template function "max" that returns the largest element in an array of N elements. You can assume that N > 1. Select a reasonable and simple interface.

5

5. (a) What does the reference operator do ? What is the difference between passing an argument by reference and passing it by value ?

4

(b) Consider the following class *Foo*, (for which one constructor is written). Write a destructor, a copy constructor and an assignment operator that would be appro-priate for the class.

6

```
class Foo {
public:
int* p;
Foo(void) {
   p = new int[10];
   for (int k = 0; k < 10; k += 1) {
      p[k] = k;
   }
}
}
```

6. (a) Declare a C++ **structure** to contain the following five pieces of information about cars on a used car lot :   4

   (1)   the manufacturer of the car,

   (2)   the model name of the car,

   (3)   the number of miles on the odometer,

   (4)   the asking price for the car.

   (b) Write a template function **alloc** that takes two parameters :

   n : the size of the array to allocate

   val : a value of type T

The **alloc** function should allocate an array of type T with n elements and set all elements in the array to the value val. A pointer to the array is returned.   6

7. (a) Give two different ways that an alias can be created for an integer variable *i* in a C++ program.   4

   (b) Write a program that uses a 2-dimensional m x n double array A. Declare the variable A and write the C++ code required to *allocate* and *deallocate* the array (assume that **m** and **n** are declared and their values are known).   6

8. (a) Distinguish between virtual member functions and non-virtual member functions.   2.5

   (b) Why are internal data members of a base class declared protected instead of private ?   2.5

(c) Define the terms *static scoping* and *dynamic scoping* and give a very simple example of each.     2.5

(d) Explain what the following three lines "do" when executed ?     2.5

```
Tmp = new int(10) ;
*Tmp = 65 ;
delete [] Tmp ;
```

IWL