Roll No. .....

Subject Code—2014

M.C.A. (Third Year) EXAMINATION

(Five Year Integrated Course)

OBJECT ORIENTED PROGRAMMING USING C++

MCA-302

Time : 3 *Hours* *Maximum Marks* : 100

Note : Attempt any *Five* questions.

1. (a) What does polymorphism in C++ ? How is polymorphism achieved at : **10**
   (i) Compile time
   (ii) Run time
   (b) What is data abstraction ? How is it implemented in C++ ? **10**

2. (a) What does inheritance means in C++ ? What are the different forms of inheritance ? Give an example for each. **15**

(b) Compare inheritance and containership. How does it enable code reusability ? 5

3. (a) Define a class of teacher. What will you define as private and public in that class ? 4

(b) Differentiate between :

(i) Class and Object

(ii) Function overloading and function overriding

(iii) Internal and external linkage of member function

(iv) Static data member and static member function. 4×4=16

4. (a) What is the 'this' pointer ? What is its advantages ? 5

(b) How a static member differ from normal member of a class ? Give an example of the requirements of a static member in the following classes : 15

(i) Teacher

(ii) Examiner.

5. (a) What is friend function ? What are the merits and demerits of using friend function ? 10

(b) What are constructors and deconstructors ? When are they called ? 5

(c) Explain the term composition with suitable example. 5

6. (a) What is converstion function ? How is it created ? Explain its syntax. 8

(b) What is the need for template function in C++ ? What are their advantages ? What is difference between template function and template class ? 12

7. (a) Differentiate between overload functions and function template. 5

(b) Explain the process of open, close, read and write for files supported by C++ with example. 15

8. (a) What will be the result of the following program argument ?

```
for (i = 0·25; i <>1·0; i = i+0·25)
{
    cout.precision(5);
    cout.width(7);
    cout<<i;
    cout.width(10);
    cout<<i*i<<*\n*;
}
```

8

(b) Explain the process of exception handling mechanism in a C++. 12