No. of Printed Pages : **2**

Sl. No.

B4.3-R4 : OBJECT ORIENTED DATABASE MANAGEMENT SYSTEM

NOTE :

1. **Answer question 1 and any FOUR from questions 2 to 7.**
2. **Parts of the same question should be answered together and in the same sequence.**

Time : 3 Hours **Total Marks : 100**

1. (a) What is the use of signatures in object oriented databases ? Briefly discuss.

(b) What is virtual function ? Briefly discuss. In context of C++, define the set of rules for virtual functions.

(c) Through an example, discuss specialization in database.

(d) Justifying the need of mediators and wrappers, discuss them briefly.

(e) Through an example in C++, discuss multilevel inheritance.

(f) What is relationship integrity in object oriented databases ? Discuss.

(g) Object and relational databases (OODBMS and RDBMS) take fundamentally different approaches to the idea of identity. Discuss, how ? **(7x4)**

2. (a) You have been given a class "Book" with following attributes : "Title" (indicating the title/name of the book), "Page" (indicating the count of pages in the book), "Price" (indicating the price of the book) and "Year" (indicating the year on which the book got published). Assume that count of pages in all the books are unique. It is desired to perform following :

(i) Write the OQL expression that produces the set of all such books whose publication year is less than 2005.

(ii) Let us consider bookList as a C++ variable (whose type is List<Book>) of the class Book. Write a sketch of the C++ function extended with OQL which assigns the set of all books of the class Book into bookList in sorted order of count of pages and prints Title, Page, Price and Year of all the books in that set.

(b) XML tags are broadly categorised as, Start Tag, End Tag and Empty Tag. Briefly discuss each of them. Also discuss the rules that need to be followed to use XML tags. **(9+9)**

3. (a) There are different *type constructors* in ODL (Object Definition Language). Briefly discuss those *type constructors* in ODL which are called as collection types.

(b) What is semi-structured data ? Highlighting the characteristics, discuss its advantages, disadvantages and the problems faced in storing semi-structured data. Also list out the sources of semi-structured data. **(6+12)**

4. (a) Majorly, XML is designed to be used in two modes, Well-formed XML and Valid XML (involves a document type definition). Through examples and XML codes, explain both modes.

(b) Briefly discuss and compare RDBMS and OODBMS. **(9+9)**

5. (a) What is Object Relational Database ? Highlight the features of ORDBMS. Making comparison on different basis, discuss how ORDBMS is different than OODBMS.

(b) Highlight the major advantages of object oriented programming. Also discuss its key features. **(12+6)**

6. (a) There are two attributes, A1 and A2 (data types of A1 and A2 are "int" and "float" respectively). Define the class XYZ in C++ with access mode of the attributes as private and default constructor to initialise the values in A1 and A2 as 10 and 54.75 respectively. In main function, create following : (i) an object of the class XYZ, (ii) an array of ten objects of the class XYZ and (iii) an array of ten object pointers of the class XYZ.

(b) Control classes, Entity classes, Interface classes and Container classes are the four major type of classes used in object oriented program. Briefly discuss them.

(c) Object Modeling Technique (OMT) is one of the object oriented data models. Briefly discuss it. Also present the notations in OMT used for following relationships : "is a", "associations" and "aggregations". **(4+6+8)**

7. (a) A pair of inverse relationships in ODL classes can be classified as either many-many, many-one in either direction, or one-one. Briefly discuss the type declarations for the pair of mentioned inverse relationship. Further, class definition of three ODL classes (Book, Author and Publisher) and their relationships are given in following figure. Discuss the pair of inverse relationships in these classes.

```
class Book
{
    attribute string Title;
    relationship          Set<Author>
authors
     inverse  Author :  :  written
  relationship  Publisher ownedBy
     inverse  Publisher : : owns
}
```

```
class Author
{
    attribute string Name;
    relationship           Set<Book>
written
        inverse Author : : authors;
}
```

```
class Publisher
{
    Attribute string Publication
    relationship Set<Book> owns
          inverse  Movie : :
ownedBy;
}
```

(b) Briefly discuss the ORION system. Also highlight the issues involved **(9+9)**

- o O o -