Total No. of printed pages = 6

Co-506/OOM/5th Sem/2017/N

OBJECT ORIENTED METHODOLOGY

Full Marks – 70

Time – Three hours

The figures in the margin indicate full marks for the questions.

PART – A

Marks – 25

Time – One hour

This part consists of 5 questions, each of which carries 5 marks equally. All the questions are compulsory.

1. Choose the appropriate options : 1×5=5

   (i) When a function is defined inside a class the function is called –

   (a) Inside function

   (b) Class function

   (c) Inline function

   (d) Interior function.

[Turn over

(ii) A static member function can access –

(a) both static and non-static member data

(b) static member data only

(c) non-static member data only

(d) neither static member data nor non-static member data.

(iii) Regarding a destructor which one of the following is true ?

(a) A destructor helps to create an object.

(b) A destructor must have a return data type.

(c) A destructor is called by the compiler.

(d) A destructor can be overloaded.

(iv) Regarding a inheritance which one of the following is true ?

(a) Inheritance provides the mechanism to reuse of existing codes.

(b) Object of sub-class cannot access the private members of the base class.

(c) Parametric constructor of sub-class cannot pass arguments to that of the base class

(d) None of the above.

(v) Regarding function overriding which one of the following is true ?

(a) It requires the concept of polymorphism.

(b) Member function of the base class to be overridden in the sub-class must be declared as a virtual member function.

(c) Base class constructor can be overridden.

(d) Overridden member function of the sub-class redefines the functionality.

2. Fill in the blanks : 1×5=5

(a) A class puts together ——— and ——— as a single entity.

(b) ——— members of a class are accessible by the outsider of the class.

(c) ——— members of a class are shareble by all the objects of that class.

(d) Name of destructor of a class is preceded by ———.

3. Answer the following in a single word/sentence each : 1×5=5

(a) Which feature of OOPS focuses on functionality of objects hiding details.

(b) Name the operator used with 'cout'.

(c) Which access specifier blocks the unauthorised access of members of a class ?

(d) Which non-member function of a class can access its private and protected members.

(e) What is the name of the parameter used in templating a function or a class ?

4. Write down whether the following are true or false : 1×5=5

(a) A pointer is a static data structure.

(b) A class is a data type.

(c) Constructors and destructors must be defined in a class by the programmer.

(d) Dynamic data structure are not supported in C++.

(e) Sub-class pointer can point to the object of the base class.

5. Match the following Columns : 1×5=5

Column A	Column B
Encapsulation	allocates space dynamically
Default constructor	an instance of a class
In function overloading	function signatures remain same
New operator	one of the features of C++
An object is	has no formal arguments

PART – B

Marks – 45

Time – 2 hours

This part consists of 6 questions, each of which carries 9 marks equally. Out of 6 answering any *five* questions are compulsory.

1. Explain briefly the main features of OOPS. Write a C++ program to overload a binary operator. 4+5=9

2. What do you mean by data type ? What is the importance of data type in programming ? Discuss about the basic and user defined data types used in C++. 2+2+5=9

3. (a) Discuss about the benefits of OOPS. 4

(b) Write a C++ program to find the largest of any three integer numbers implementing the concept of class and object. 5

4. Differentiate between early binding and late binding. Write a C++ program to realise the concept of pure virtual function. 3+6=9

5. What is template ? How does this help the programmer in programming ? Write a C++ program to add any two numbers of any type realising the concept of template function and template class. 2+2+5=9

6. Write short notes on any *three* : 3×3=9

(a) Friend function

(b) Constructor

(c) Scope resolution operator

(d) Multiple inheritance.