



**I. Choose the correct answer**

[15 × 1 = 15]

1. The term is used to describe a programming approach based on classes and objects is  
(A) OOP                      (B) POP                      (C) ADT                      (D) SOP
2. The paradigm which aims more at procedures.  
(A) Object Oriented Programming                      (B) Procedural programming  
(C) Modular programming                      (D) Structural programming
3. Which of the following is a user defined data type?  
(A) class                      (B) float                      (C) int                      (D) object
4. The identifiable entity with some characteristics and behaviour is.  
(A) class                      (B) object                      (C) structure                      (D) member
5. The mechanism by which the data and functions are bound together into a single unit is known as  
(A) Inheritance                      (B) Encapsulation                      (C) Polymorphism                      (D) Abstraction
6. Insulation of the data from direct access by the program is called as  
(A) Data hiding                      (B) Encapsulation                      (C) Polymorphism                      (D) Abstraction
7. Which of the following concept encapsulate all the essential properties of the object that are to be created?  
(A) class                      (B) Encapsulation                      (C) Polymorphism                      (D) Abstraction
8. Which of the following is the most important advantage of inheritance?  
(A) data hiding                      (B) code reusability                      (C) code modification                      (D) accessibility
9. “Write once and use it multiple time” can be achieved by  
(A) Redundancy                      (B) reusability                      (C) modification                      (D) composition
10. Which of the following supports the transitive nature of data?  
(A) Inheritance                      (B) Encapsulation                      (C) Polymorphism                      (D) Abstraction
11. The variables declared inside the class are known as  
(A) data                      (B) inline                      (C) method                      (D) attributes
12. Which of the following statements about member functions are True or False?  
i) A member function can call another member function directly with using the dot operator.  
ii) Member function can access the private data of the class.  
(A) i)True, ii)True                      (B) i)False, ii)True                      (C) i)True, ii)False                      (D) i)False, ii)False

13. A member function can call another member function directly, without using the dot operator called as

- (A) Sub function (B) sub member  
(C) Nesting of member function (D) sibling of member function

14. The member function defined within the class behave like ..... functions

- (A) inline (B) Non inline (C) Outline (D) Data

15. Which of the following access specifier protects data from inadvertent modifications?

- (A) Private (B) Protected (C) Public (D) Global

**II. Answer any 5 of the following questions:**

**[5 × 2 = 10]**

16. What is polymorphism?

17. How encapsulation and abstraction is are interrelated?

18. Write the disadvantages of OOP.

19. What are called members?

20. Define a class.

21. Differentiate structure and class though both are user defined data type.

22. What is the difference between the class and object in terms of oop?

**III. Answer any 5 of the following questions:**

**[5 × 3 = 15]**

23. What do you mean by modularization and software reuse?

24. Define information hiding.

25. Write the general form of class definition.

26. Write short note on access specifiers.

27. Write the procedure to creating objects

28. Rewrite the following program after removing the syntax errors if any and underline the errors:

```
#include<iostream>
#include<stdio>
class mystud
{
    int studid =1001;
    char name[20];
public
    mystud() { }
    void register ( )
    {
        cin>>stdid;
```

```

        gets(name); \
    }
    void display ()
    {
        cout<<studid<<”: “<<name<<endl;}
    }
int main()
{
    mystud MS;
    register.MS( );
    MS.display( );
}

```

29. Write with example how will you dynamically initialize objects? Given the following C++ code, answer the questions (i) & (ii).

```

class TestMeOut
{
    public:
        ~TestMeOut() //Function 1
        {
            cout<<“Leaving the exam hall”<<endl;
        }
        TestMeOut() //Function 2
        {
            cout<<“Appearing for exam”<<endl;
        }
        void MyWork() //Function 3
        {
            cout<<“Answering”<<endl;
        }
};

```

- (i) In Object Oriented Programming, what is Function 1 referred as and when does it get invoked / called?
- (ii) In Object Oriented Programming, what is Function 2 referred as and when does it get invoked / called?

#### IV. Answer the following questions:

[2 × 5 = 10]

30. a) What are the advantages of OOPs?

[OR]

b) Write a note on the basic concepts that support OOPs?

31. a) Define a class RESORT with the following description in C++ :

Private members:

Rno // Data member to store room number

Name //Data member to store user name

Charges //Data member to store per day charge

Days //Data member to store the number of days

Compute( )/\*A function to calculate total amount as Days \* Charges and if the total amount exceeds 11000 then total amount is 1.02 \* Days \*Charges \*/

Public member:

GetInfo( ) /\* Function to Read the information like name , room no, charges and days\*/

DispInfo( )/\* Function to display all entered details and total amount calculated using COMPUTE function\*/

[OR]

b) Write the output of the following

```
#include<iostream>
using namespace std;
class student
{
    int rno, marks;
public:
    student(int r,int m)
    {
        cout<<"Constructor "<<endl;
        rno=r;
        marks=m;
    }
    void printdet()
    {
        marks=marks+30;
        cout<<"Name: Bharathi"<<endl;
        cout<<"Roll no : "<<rno<<"\n";
        cout<<"Marks : "<<marks<<endl;
    }
};
int main()
{
    student s(14,70);
    s.printdet();
    cout<< "Back to Main";
    return 0;
}.
```