**Department of Computer Science & Engineering**

**SLIET Longowal**

# Question Bank

**of**

## Object Oriented Programming (CS-221/PCCS-202)

**for**

## Integrated Certificate & Diploma in Computer Science (ICD-CS) 4th Sem

**Submitted by**

**Er. Rahul Gautam (A.P., CSE)**

Title of the course : **Object Oriented Programming**

Subject Code : **CS-221**

Weekly load : 6 Hrs LTP 2-0-4

Credit : 4 (Lecture 2, Practical 2)

**Course Outcomes:** At the end of the course, the student will be able to:

| | |
|---|---|
| CO1 | Apply object-oriented approach to design the programs. |
| CO2 | Understand reusability of code using inheritance. |
| CO3 | Analyze polymorphic and virtual behaviour of functions. |
| CO4 | U Use stream classes in file-handling. |

| CO/PO Mapping : (Strong(S)/Medium(M)/Weak(W) indicates strength of correlation) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| COs | Programme Outcomes (POs) | | | | | | | | | |
| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 |
| CO1 | | S | S | M | | | | | | S |
| CO2 | | S | M | W | | | | | | S |
| CO3 | | S | S | W | | | | | | S |
| CO4 | | S | S | M | | | | | | S |

**Theory**

| Unit | Main Topics | Course outlines | Lecture(s) |
|---|---|---|---|
| **Unit-1** | 1. Introduction | Object-oriented programming, characteristics of object-oriented languages, C++ Programming Basics: Basic program construction, Pre-processor directives, variables, Operators, Library functions, manipulators. | 04 |
| | 2. Decision-making | Relational operators: loops; decisions; logical operators; other control Statements | 04 |
| | 3. Structures and Functions | Structure enumerated data types; functions; passing arguments to functions and returning values from functions, unions. | 03 |
| | 4. Classes and Objects | Creation, accessing class members, Private Vs Public, Constructor and Destructor Objects. | 03 |
| **Unit-2** | 5. Member Functions | Method definition, Inline functions implementation, Constant member functions Friend Functions and Friend Classes, Static functions Overloading Member Functions, Need of operator overloading, operator overloading | 05 |
| | 6.Inheritance | Definition of inheritance, protected data, private data, public data, inheriting constructors and destructors, constructor for virtual base classes, constructors and destructors of derived classes, size of a derived class, order of invocation, types of inheritance, single inheritance, hierarchical inheritance, multiple inheritance, hybrid inheritance, multilevel inheritance. | 05 |

| | | | |
|---|---|---|---|
| | 7. Polymorphism and Virtual Functions | Importance of virtual function, function call binding, implementing late binding, need for virtual functions, abstract base classes and pure virtual functions, virtual destructors | 04 |
| | 8. File and Streams | File and Streams components of a file, different operation of the file, creation of file streams, stream classes, header files, updating of file, opening and closing a file. | 04 |

**Total=32**

**Recommended Books:**


1. SB Lippman and J Lajoie,C++ Primer, Addison Wesley ,New Delhi
2. KR Venugopal , Mastering C++ , TMH Publishing
3. E. Balaguruswamy, Object Oriented Programming in C++, TMH Publishing Co. Ltd, New Delhi.
4. Robert Lafore, C++, Galgotia Publications Pvt. Ltd., Daryaganj, New Delhi.

# Object Oriented Programming

## MCQs

1. Which of the following is not a feature of OOP in C++?
   a. Encapsulation
   b. Inheritance
   c. Polymorphism
   **d. Compilation**
2. Which OOP concept refers to the ability to create a new class from an existing class?
   a. Encapsulation
   **b. Inheritance**
   c. Polymorphism
   d. Abstraction
3. Which of the following best describes a constructor in C++?
   a. A function that is called when an object is destroyed
   b. A function that is called when an object is accessed
   **c. A function that is called when an object is created**
   d. A function that is called when an object is inherited
4. What is the default access specifier for members of a class in C++?
   a. Public
   **b. Private**
   c. Protected
   d. None
5. What is Encapsulation in C++?
   a. The ability to inherit properties from another class
   b. The ability to overload functions
   c. The ability to create abstract classes
   **d. The wrapping of data and methods into a single unit**
6. Which OOP principle allows a derived class to override a method in the base class?
   a. Encapsulation
   **b. Polymorphism**
   c. Abstraction
   d. Inheritance
7. Which of the following type of class allows only one object of it to be created?
   a. Virtual class
   b. Abstract class
   **c. Singleton class**
   d. Friend class
8. If class A is a friend of class B and class B is a friend of class C, which of the following is true?
   a. Class A is a friend of class C
   b. Class B cannot be friend of any other class
   c. Class C is friend of Class A
   **d. None of the above**
9. Which of the following statements is correct?
   **a. Base class pointer cannot point to derived class.**

b. Derived class pointer cannot point to base class.
c. Pointer to derived class cannot be created.
d. Pointer to base class cannot be created.

10. Which of the following concepts means determining at runtime what method to invoke?
    a. Data hiding
    b. Dynamic Typing
    **c. Dynamic binding**
    d. Dynamic loading
11. Which of the following concepts of OOP means exposing only necessary information to the client?
    a. Encapsulation
    **b. Abstraction**
    c. Data hiding
    d. Data binding
12. Which of the following is correct about function overloading?
    a. The types of arguments are different
    b. The number of arguments is same
    c. The order of argument is different
    **d. Both A and C**
13. How is "Late binding" implemented in C++?
    a. Using C++ Tables
    b. **Using virtual Tables**
    c. Using indexed virtual tables
    d. Using polymorphic tables

14. Which inheritance type is used in the class given below?

```
class A : public X, public Y {};
```

    a. **Multiple Inheritance**
    b. Multilevel Inheritance
    c. Hierarchical Inheritance
    d. Hybrid Inheritance
15. Which one of the following is the correct way to declare a pure virtual function?
    a. virtual void Display(void){0};
    b. virtual void Display = 0;
    c. void Display(void) = 0;
    **d. virtual void Display(void) = 0;**
16. Which of the following statements regarding inline functions is correct?
    a. Its speed's up execution
    b. It increases code size
    **c. Both A and B**
    d. None of the above
17. What happens if the base and derived class contains a definition of a function with the same prototype?
    a. Compiler reports an error on compilation.
    b. Only base class function will get called irrespective of object.
    c. Only derived class function will get called irrespective of object.

**d. Base class object will call base class function and derived class object will call derived class function.**

18. What is the main difference between overloading and overriding in C++?
    **a. Overloading occurs when two or more methods in the same class have the same name but different parameters; overriding occurs when a derived class has a definition for one of the member functions of the base class.**
    b. Overloading occurs in the same class, overriding occurs between a base class and a derived class
    c. Overloading requires the use of the 'virtual' keyword, overriding does not.
    d. Overloading and overriding are the same in C++.
19. Which is not a feature of OOP in general definitions?
    a. Efficient Code
    b. Code reusability
    c. Modularity
    **d. Duplicate/Redundant data**
20. How can we restrict dynamic allocation of objects of a class using new?
    a. By overloading new operator
    b. By making an empty private new operator.
    **c. By making an empty private new and new[] operators**
    d. By overloading new operator and new[] operators
21. Which of the following operators are overloaded by default by the compiler in every user defined classes even if user has not written?
    1) Comparison Operator ( == )
    2) Assignment Operator ( = )
    a. only 1
    **b. only 2**
    c. both 1 and 2
    d. none of the above

22. Which of the following is not a member access specifier in C++?
    a. public
    b. private
    c. protected
    **d. global**
23. What is the output of the given code fragment?

```
int f = 1, i = 2;
while(++i < 5)
{ f *= i;}
 cout << f;
```

    **a. 12**
    b. 24
    c. 6
    d. 7
24. Identify the correct extension of the user-defined header file in C++.
    **a.** .cpp  b. .hg  **c. .h**  d. .hf

25. Identify the incorrect constructor type.

a. **Friend constructor** c. parametrized constructor
b. Default constructor d. copy constructor

26. Which of the following is “address of operator”?.
a. * c. []
b. **&** d. &&

27. Which keyword is used to declare a destructor in C++?
a. ~ b. ! c. # d. $

28. Which of the following operators cannot be overloaded?
a. . (Member Access or Dot operator) c. :: (Scope Resolution Operator)
b. ?: (Ternary or Conditional Operator ) **d. All of the above**

29. Find the output:
```
class Test { static int i; int j; };
int Test::i;
int main() { cout << sizeof(Test); return 0; }
```
a. 10 b. 20 c. 0 **d. 4**

30. The programming language that has the ability to create new data types is called_____.
a. encapsulation b. polymorphism **c. extensible** d. reprehensible

31. Inline functions are useful when
a. Function is large with many nested loops
**b. Function is small and we want to avoid function call overhead**
c. Function has many static variables
d. None of these

32. Find the output:
```
class P {public: void print(){ cout <<" Inside P::"; } };
class Q : public P {public: void print(){ cout <<" Inside Q"; }};
class R: public Q {};
int main(void){R r; r.print(); return 0;}
```
a. Inside P b. Inside P:: **c. Inside Q** d. no output

33. The C++ language is _____ object-oriented language.
a. Pure object oriented **c. semi/partial object oriented**
b. Not object oriented d. none of the above

34. Polymorphism means
**a). Single name multiple form** b). Multiple name single form
c). Object d). Class

35. In compile time polymorphism
**a. An object is bound to its function at compile time**
b. An object is not bound its function.

c. An object is bound to its function at run time.
d. Virtual function use.

36. Out of following which operator cannot overload
    a. +=
    b. ==
    c. *
    d. **::**

37. Operator overloading is
    **a. a concept by which we can give special meaning to an operator of language.**
    b. an object of a language.
    c. a concept by which we can design new object.
    d. a class

38. The reusability is advantage of inheritance.
    **a. True**
    b. False
    c. Can't say

39. Constructor is a
    a**). Member function** b). Member data c). Object d). Class

40. Constructor overloading means
    a). **Multiple constructor in a class** b). Single constructor in a class.
    c). One constructor and one destructor in a class d). No constructor is used in a class.

41. Copy Constructor are used
    **a). to copy one object to other one** b). to change the name of data member.
    c). to change the name of member function. d). to verify object.

42. Struct is a
    **a). keyword in *C*++** b). Object in *C*++ c). Class in *C*++ d). function in *C*++

43. File is closed by *close()* function.
    a. True
    b. False
    c. Can't say

44. A **do-while** loop is useful when we want that the statements within the loop must be executed

    a) only once. **c) at least once.**
    b) more than once. d) for a fixed number of iterations.

45. A **continue** statement causes execution to skip to the

    a) **r**eturn 0; statement. c) next iteration of the loop.
    b) first statement after the loop. **d) statement after continue statement.**

46. Which of the following is not a comparison operator in C++?
    a. == c. <
    **b.** >= **d. ++**

47. Which of the following is not a decision-making statement in C++?
    a. if
    b. switch
    **c) for**
    d) Ternary operator

48. What will be the values of **x**, **m**, and **n** after the execution of the following statements:
    int x, m, n;
    m = 10;
    n = 15;
    x = ++m + n++;

    a) x = 25, m = 10, n = 15
    **b) x = 26, m = 11, n = 16**
    c) x = 27, m = 11, n = 16
    d) x = 27, m = 10, n = 15

49. Which of the following is a valid way to set the width of the output field using manipulators in C++?

    a) cout << setwidth(10)
    **b) cout << setw(10)**
    c) cout << setfield(10)
    d) cout << setlength(10)

50. What is the difference between a for loop and a while loop in C++?

a) There is no difference between a for loop and a while loop.
b) A for loop requires an initial statement and a condition, while a while loop only requires a condition.
c) A for loop can only be used with integer variables, while a while loop can be used with any variable type.
**d) A for loop executes a set number of times, while a while loop executes as long as the condition is true.**

51. What is the purpose of the **break** statement in a loop in C++?

    a) It break the iteration and continue to the next iteration.
    **b) It exits the loop and continues with the code following the loop.**
    c) It returns a value to the function that called the loop.
    d) It continues the loop with the next value in the loop condition.

52. Which of the following is true about constructors. 1) They cannot be virtual. 2) They cannot be private. 3) They are automatically called by new operator.
    a. All 1,2 and 3
    b. Only 1 and 2
    **c. Only 1 and 3**
    d. Only 2 and 3

53. Constructors have _____ return type.
    a. Void
    b. Char
    c. Int
    **d. No**

54. What is the right way to declare a copy constructor of a class if the name of the class is MyClass?

a. MyClass (constant MyClass *arg)
**b. MyClass (constant MyClass &arg)**
c. MyClass (MyClass arg)
d. MyClass (MyClass *arg)

55. Which of the followings is/are automatically added to every class, if we do not write our own.
a. Copy Constructor
b. Assignment operator
c. A constructor without any parameter
**d. All of the Above**

# Object Oriented Programming

## Short Answer Questions

1. What is Object Oriented Programming (OOPs)?
2. Enlist main features of object-oriented programming
3. Explain the essence of class in OOPs.
4. How is object related to a class?
5. Differentiate compile time polymorphism and run time polymorphisms.
6. What is the essence of inheritance in OOPs.
7. How is abstraction implemented in OOPs?
8. What are access specifiers? What is their significance in OOPs?
9. What is the need of friend function in object-oriented programming?
10. How static data member is different from other data members?
11. How static member functions behave in a class?
12. What is inheritance? Enlist different types of inheritance?
13. How are constructors invoked in C++ programming?
14. Discuss the role of constructor and destructor in OOP.
15. In what sequence constructor and destructors are called?
16. Differentiate the role of break and continue statement in a program.
17. What are the different kinds of manipulators?
18. Differentiate function overloading and overriding in brief.
19. What is the difference between structured programming and object oriented programming?
20. Is it always necessary to create objects from class? justify your answer.
21. What is the role of pure virtual functions in C++ programming?
22. Differentiate structure and union.
23. How class is different from structure?
24. What is difference between while loop and do-while loop.
25. What are the different file opening modes?
26. Discuss the essence of const member functions.
27. What is the role of virtual destructors?
28. How multilevel inheritance is different from multiple inheritance?
29. Can we overload constructor in a class? justify your answer.
30. Can we overload destructor in a class? justify your answer.
31. Write a class definition that creates a class called **CDE** with one **private** data member **number_of_students**, of type **int** and one **public** function whose declaration is void **show_students_data().**
32. Write the declaration for a friend function called **cse()** that returns type **void** and takes one argument of class **sliet**.
33. Write the function declaration for overloading the binary **+** operator.
34. Explain the essence of inline functions in object-oriented programming.
35. Illustrate recursive calls and recursive function.

# Object Oriented Programming

## Descriptive type Questions

1. Distinguish between the following terms:
   a. Objects and classes
   b. Data abstraction and data encapsulation
   c. Inheritance and polymorphisms
   d. Dynamic binding and message passing
2. Write a C++ program using class that will ask for a temperature in Fahrenheit and display it in Celsius.
3. Write a function using reference variables as arguments to swap the value of a pair of integers.
4. What do you mean by the term 'constructor'? How is it created in a program? What is the difference between a default constructor and a parameterized constructor? Explain with the help of a suitable example.
5. WAP to write 'Name' & 'Reg_No.' of a student in a file named "DCS" and then display it on screen by reading from the file. Explain each step of the program clearly.
6. Create a class called **time** that has separate **int** member data for hours, minutes, and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. Another member function should display it, in HH:MM:SS format. Write a **main()** function to exercise this class.
7. Write a program in C++ using the concept of class & object to check whether a given number is prime or not?
8. What is operator overloading? How it is useful in oops? Explain the concept of operator overloading by writing a program to overload any arithmetic operator.
9. What is ambiguity in inheritance? How can we solve this problem? Explain the concept by writing a suitable program.
10. Write a program in *C++* to write name, registration no. and roll no. in a file and read the same from the file.
11. What is object-oriented programming? How it is different from procedure-oriented programming? Explain the various features of oops.
12. What is the need of virtual base class? Explain the use of virtual base class by writing a program in C++.
13. How constructor works in inheritance. Demonstrate the use of copy constructor in a program.
14. Write a program in *C++* which get the name, roll no. and marks of five subjects of a student from the keyboard, calculate the percentage and then print the result. (use the concept of classes & objects)
15. Write a program in *C++* which calculates factorial of a number with and without recursion.
16. Discuss importance of inheritance in OOP. Also explain the working of different types of inheritance.
17. What do you mean by the term streaming? Discuss different types of stream classes and header files.
18. Demonstrate the use of break and continue statement by writing suitable programs(s).
19. Differentiate switch-case statement and If-else-if ladder by writing suitable program(s).

20. Discuss the role of friend functions and Virtual functions by writing suitable program(s).
21. How is polymorphism implemented using compile time and run-time? Discuss polymorphism in inheritance.
22. Create a class FLOAT that contains one float data member. Overload all the four arithmetic operators so that they operate on the objects of FLOAT.