

Object Oriented Programming through C++

1 Introduction

1. What is Procedure Oriented Programming? What are its characteristics?
2. What is Object Oriented Programming? How does it differ from Procedure Oriented Programming? What are the benefits of OOP? [CU 2006 2011 SU 2012]
3. Differentiate between – Object and Class, Data Abstraction and Encapsulation. [SU 2012]
4. What do you mean by message passing?
5. Differentiate between Object Oriented Programming and Object Based Programming?
6. What do you mean by Polymorphism? Illustrate with suitable example. [SU 2012 2014]
7. Describe the purpose of Inheritance as a feature of OOP.
8. What are the basic data types available in C++? Explain the following -
 - Type compatibility
 - Implicit conversion [CU 2013]
9. What do you mean by cascading of I/O operators in C++?
10. What is a reference variable? What are its major uses?
11. What do you mean by Dynamic Initialization of Variables?
12. What are the advantages of using new operator as compared to *malloc()*?
13. How does a constant defined by *const* differ from constant defined by *#define*?
14. What do you mean by manipulators in C++? Explain with suitable examples.
15. What do you mean by Chained, Embedded Assignment and Compound Assignment?
16. Differentiate between the following statements

```
char * const p;  
char const * p;
```

[SU 2012]
17. What do you mean by dangling pointer? Explain with an example. [SU 2014]
18. “In C++, a variable can be declared anywhere within its scope” – Justify the significance of this feature.
19. What is inline function? When do you make a function inline and why? How does it differ from macro? Differentiate between a function and an inline function. [CU 2014 SU 2012]
20. When do we need to use default arguments in a function? Illustrate with example.
21. What do you mean by function overloading? When should we use this concept? [SU 2012]
22. What are the manipulators used in C++? [SU 2013]

2 Class & Objects, Constructors & Destructors

1. How does structure in C and C++ differ?
2. What is a class? How does it differ from a structure? How does it accomplish data hiding? [CU 2008 2009]
3. How does a C++ structure differ from a C++ class? [CU 2004 SU 2015]
4. What is data hiding? What are the different methods for protecting data from external users? [SU 2011]
5. Define an *instance variable* and *member function*. Give example. [CU 2008]
6. What is the utility of scope resolution operator? [SU 2014 2015]
7. How a member function of a class is defined? Can we use the same function name for a member function of a class and an outside function in the same program file? If yes, how are they distinguished? If no, give reasons. [CU 2009]
8. Describe the mechanism of accessing data members and member functions in the following situations.
 - (a) Inside the main() program.
 - (b) Inside the member function of the same class.
 - (c) Inside the member function of some other class.
9. What will happen if a member function is declared as private? How to resolve the issue?
10. State the different characteristics of a static data member and a static member function of a class.
11. What is a friend function? State the different characteristics that friend functions possess. List the merits and demerits of using a friend function. [CU 2009 2011 SU 2013 2015]
12. Describe the objective of declaring a *friend class* through an example. [CU 2013 2014]
13. State the utility of declaring a member function as *const*.
14. What are empty classes? Can instances of empty classes be created? Give reasons. [SU 2011]
15. What is constructor? What is the role of a constructor. What are the properties of a constructor? [CU 2006 2007 2011]
16. Is it mandatory to use a constructor in a class?
17. What do you mean by parameterized constructor? Can we have more than one constructor in a class? If yes, explain the need. [CU 2008 2012]
18. What are the differences between default and parameterized constructors? [SU 2011]
19. What are read only objects? What is the role of constructors in creating such object? [SU 2011]
20. What do you mean by constructor with default arguments?
21. Can we have more than one constructor in a class? If yes, explain the need for such a situation with suitable example. [CU 2006 2007 2011]
22. Distinguish between the following two statements
Time T2(T1);
Time T2 = T1;
where T1 and T2 are the objects of the class Time
23. What do you mean by default constructor? [CU 2008 2013]
24. What is copy constructor? Illustrate with suitable example. [CU 2007 2008 SU 2013]
25. Differentiate between *assignment operator* and *copy constructor*. [SU 2014]
26. What is default copy constructor? [CU 2007 2008]
27. What do you mean by dynamic copy constructor?
28. Describe the importance of destructor with a suitable example. [CU 2006 2007 2011 2013 SU 2012]

29. "Destructors cannot be overloaded" – justify. [SU 2013]
30. What do you mean by dynamic initialization of objects? Why do we need to do this? How is dynamic initialization of objects achieved? [SU 2013]
31. Write a class declaration name *Circle* with a private member variable named *radius*. Write *set* and *get* functions to access the radius variable and a function named *Getarea()* that returns the area of the circle. Add a default constructor to the circle class where the constructor should initialize the radius member to zero. [CU 2013]

3 Inheritance

1. What does inheritance means in C++? What are the different forms of inheritance? Give an example for each of the forms. [CU 2008]
2. What do you mean by private, public and protected mode of inheritance? State the need for protected access specifier for data members of a class? [CU 2007]
3. We know that private members of a base class is not inheritable. Is it possible for the objects of a derived class to access the private members of the base class? If yes, how? Note that the base class can not be modified. [CU 2009]
4. Show with an example how constructors are called in derived classes? [CU 2014]
5. Differentiate between protected and private members of a class. [CU 2013]
6. Describe single, multiple and multilevel inheritance. Discuss the inherited access restrictions for base class members with the derived class. [CU 2008 SU 2013]
7. Explain hybrid inheritance with suitable example. [SU 2015]
8. Is it possible to use multilevel and multiple inheritances simultaneously? Is there any problem associated with this? If yes, how it is resolved?
or
State the role of virtual base class in inheritance ambiguity. [CU 2006 2007 SU 2011 2014]
9. What is orthogonal base class? [SU 2014]
10. Class D is derived from class B and class D does not contain any data member of its own. Does class D required constructor? Is yes, why?
11. How do the properties of the following two derived classes differ?
class D_1 : private B ;
class D_2 : public B ; where D_1 , D_2 & B are classes. [CU 2012]
12. In multiple inheritance what will happen if there are functions with same name and signature in both the base classes and object of derived class tries to call the function?
13. Design a class named Employee having members Employee name, Employee number and Hire date. Write necessary constructors. Also, write a class named Production worker that is derived from Employee class. This class should have the information about shift and hourly pay rate. The workday is divided into two shifts - day and night. Write one / more constructor(s) and the appropriate function for the class. Demonstrate the classes by writing a program that uses a production worker object. [CU 2013]

4 Polymorphism - Operator overloading, Virtual function

1. What is polymorphism? [SU 2011 2012]
2. How is polymorphism achieved at - (i) Compile time and (ii) Run time? [CU 2009 2012 2014]
3. What is operator overloading? Why is it necessary? [CU 2009 2013 SU 2011 2012]
4. Overload pre-increment, post-increment and subs-script operator.
5. How can we overload binary operator? Explain with example.

6. When is it necessary to use a friend function to overload an operator? [CU 2012 2014]
7. Is it possible to overload a binary operator using a friend function? If yes, illustrate with suitable example.
8. "A friend function cannot be used to overload '=' operator" – justify. [CU 2009]
9. List the operator that cannot be overloaded.
10. Why can't we overload ternary operator? [CU 2010 SU 2011]
11. What is function overloading? [CU 2008]
12. Explain with suitable example - static binding & dynamic binding. [CU 2007]
13. We have two classes X and Y. If A is an object of X and B is an object of Y and we want to say $A = B$, what type of conversion routine should be used and where?
14. What do you mean by conversion function? How is it created?
15. Define two classes Polar and Cartesian to represent points in Polar and Cartesian co-ordinates. Use conversion routine to convert one system to other.
16. Define a class string, use overloaded '==' operator to compare two strings and '+' operator to concatenate two strings.
17. Create a class Matrix of size $M \times N$. Define addition, subtraction and multiplication using operator overloading.
18. What do you mean by 'this' pointer? Give an example to show the necessity of using 'this'.
19. What is virtual function? Why do we need virtual functions? When do we make virtual functions pure? What are the implications of making a function pure virtual function? Differentiate between a virtual function & an overloaded function. [CU 2010 2014 SU 2013]
20. Write a program to create a class *Doctor* with a virtual function *Salary*. Derive class *Visiting-Doctor* and implement function *Salary* on it. [CU 2014]
21. What do you mean by an abstract class? [SU 2012 2014]
22. What is virtual constructor and virtual destructor? [SU 2011]

5 Exception Handling

1. What is an exception? How an exception is handled in C++? What are the advantages of using exception handling mechanism in a program? [CU 2011 SU 2012]
2. Write the functions of throw, catch and try blocks with respect to exception handling.

6 Template

1. What is generic programming? How is it implemented in C++? [CU 2012 2014]
2. What is a template class? Why is it used? [SU 2011]
3. What is the difference between a template and a macro? [CU 2012]
4. How are class templates with multiple parameters used in C++? [CU 2012]
5. Differentiate between overloaded functions and function templates.
6. Differentiate between the terms class template and template class. [SU 2014]
7. Explain function template and class template with an example. [SU 2012]
8. Write a function swap () that swaps two values of any given data type? [SU 2014]
9. Write a function template that will find the maximum of the elements of an array. [SU 2013]

7 Managing I/O & File Handling

1. What is the difference between opening a file with a constructor and with `open()` function? When is one method preferred over the other?
2. Describe different methods for detecting *end-of-file* in C++? [CU 2012]

8 Namespace

1. What is namespace? What is unnamed namespace? What is the requirement of namespace? [CU 2011 SU 2012]