**Object Oriented Programming MCQ (Multiple Choice Questions)**

[Here are 1000 MCQs on Object Oriented Programming (Chapterwise)](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#opp-chapters).

1. Who invented OOP?  
a) Andrea Ferro  
b) Adele Goldberg  
c) Alan Kay  
d) Dennis Ritchie  
View Answer

Answer: c  
Explanation: Alan Kay invented OOP, Andrea Ferro was a part of SmallTalk Development. Dennis invented C++ and Adele Goldberg was in team to develop SmallTalk but Alan actually had got rewarded for OOP.

2. Which is not a feature of OOP in general definitions?  
a) Efficient Code  
b) Code reusability  
c) Modularity  
d) Duplicate/Redundant data  
View Answer

Answer: d  
Explanation: Duplicate/Redundant data is dependent on programmer and hence can’t be guaranteed by OOP. Code reusability is done using inheritance. Modularity is supported by using different code files and classes. Codes are more efficient because of features of OOP.

3. Which was the first purely object oriented programming language developed?  
a) Kotlin  
b) SmallTalk  
c) Java  
d) C++  
View Answer

Answer: b  
Explanation: SmallTalk was the first programming language developed which was purely object oriented. It was developed by Alan Kay. OOP concept came into the picture in 1970’s.

4. When OOP concept did first came into picture?  
a) 1980’s  
b) 1995  
c) 1970’s  
d) 1993  
View Answer

Answer: c  
Explanation: OOP first came into picture in 1970’s by Alan and his team. Later it was used by some programming languages and got implemented successfully, SmallTalk was first language to use pure OOP and followed all rules strictly.

5. Which feature of OOP indicates code reusability?  
a) Abstraction  
b) Polymorphism  
c) Encapsulation  
d) Inheritance  
View Answer

Answer: d  
Explanation: Inheritance indicates the code reusability. Encapsulation and abstraction are meant to hide/group data into one element. Polymorphism is to indicate different tasks performed by a single entity.

advertisement

6. Which header file is required in C++ to use OOP?  
a) OOP can be used without using any header file  
b) stdlib.h  
c) iostream.h  
d) stdio.h  
View Answer

Answer: a  
Explanation: We need not include any specific header file to use OOP concept in C++, only specific functions used in code need their respective header files to be included or classes should be defined if needed.

7. Why Java is Partially OOP language?  
a) It allows code to be written outside classes  
b) It supports usual declaration of primitive data types  
c) It does not support pointers  
d) It doesn’t support all types of inheritance  
View Answer

Answer: b  
Explanation: As Java supports usual declaration of data variables, it is partial implementation of OOP. Because according to rules of OOP, object constructors must be used, even for declaration of variables.

8. Which among the following doesn’t come under OOP concept?  
a) Data hiding  
b) Message passing  
c) Platform independent  
d) Data binding  
View Answer

Answer: c  
Explanation: Platform independence is not feature of OOP. C++ supports OOP but it’s not a platform independent language. Platform independence depends on the programming language.

9. Which is the correct syntax of inheritance?  
a) class base\_classname :access derived\_classname{ /\*define class body\*/ };  
b) class derived\_classname : access base\_classname{ /\*define class body\*/ };  
c) class derived\_classname : base\_classname{ /\*define class body\*/ };  
d) class base\_classname : derived\_classname{ /\*define class body\*/ };  
View Answer

Answer: b  
Explanation: Firstly, keyword class should come, followed by the derived class name. Colon is must followed by access in which base class has to be derived, followed by the base class name. And finally the body of class. Semicolon after the body is also must.

10. Which feature of OOP is indicated by the following code?

class student{ int marks; };

class topper:public student{ int age; topper(int age){ this.age=age; } };

a) Encapsulation and Inheritance  
b) Inheritance and polymorphism  
c) Polymorphism  
d) Inheritance  
View Answer

Answer: a  
Explanation: Encapsulation is indicated by use of classes. Inheritance is shown by inheriting the student class into topper class. Polymorphism is not shown here because we have defined the constructor in the topper class but that doesn’t mean that default constructor is overloaded.

11. The feature by which one object can interact with another object is \_\_\_\_\_\_\_\_\_\_\_\_\_  
a) Message reading  
b) Message Passing  
c) Data transfer  
d) Data Binding  
View Answer

Answer: b  
Explanation: The interaction between two object is called the message passing feature. Data transfer is not a feature of OOP. Also, message reading is not a feature of OOP.

12. Which among the following, for a pure OOP language, is true?  
a) The language should follow at least 1 feature of OOP  
b) The language must follow only 3 features of OOP  
c) The language must follow all the rules of OOP  
d) The language should follow 3 or more features of OOP  
View Answer

Answer: c  
Explanation: The language must follow all the rules of OOP to be called a purely OOP language. Even if a single OOP feature is not followed, then it’s known to be a partially OOP language.

13. How many types of access specifiers are provided in OOP (C++)?  
a) 4  
b) 3  
c) 2  
d) 1  
View Answer

Answer: b  
Explanation: Only 3 types of access specifiers are available. Namely, private, protected and public. All these three can be used according to the need of security of members.

14. In multilevel inheritance, which is the most significant feature of OOP used?  
a) Code efficiency  
b) Code readability  
c) Flexibility  
d) Code reusability  
View Answer

Answer: d  
Explanation: The classes using multilevel inheritance will use the code in all the subsequent subclasses if available. Hence the most significant feature among the options given is code reusability. This feature is generally intended to use the data values and reuse the redundant functions.

15. What is encapsulation in OOP?  
a) It is a way of combining various data members and member functions that operate on those data members into a single unit  
b) It is a way of combining various data members and member functions into a single unit which can operate on any data  
c) It is a way of combining various data members into a single unit  
d) It is a way of combining various member functions into a single unit  
View Answer

Answer: a  
Explanation: It is a way of combining both data members and member functions, which operate on those data members, into a single unit. We call it a class in OOP generally. This feature have helped us modify the structures used in C language to be upgraded into class in C++ and other languages.

16. Which of the following is not true about polymorphism?  
a) Helps in redefining the same functionality  
b) Increases overhead of function definition always  
c) It is feature of OOP  
d) Ease in readability of program  
View Answer

Answer: b  
Explanation: It never increases function definition overhead, one way or another if you don’t use polymorphism, you will use the definition in some other way, so it actually helps to write efficient codes.

17. Which constructor will be called from the object created in the below C++ code?

class A

{

int i;

A()

{

i=0; cout&lt;&lt;i;

}

A(int x=0)

{

i=x; cout&lt;&lt;I;

}

};

A obj1;

a) Parameterized constructor  
b) Default constructor  
c) Run time error  
d) Compile time error  
View Answer

Answer: d  
Explanation: When a default constructor is defined and another constructor with 1 default value argument is defined, creating object without parameter will create ambiguity for the compiler. The compiler won’t be able to decide which constructor should be called, hence compile time error.

18. What is an abstraction in object-oriented programming?  
a) Hiding the implementation and showing only the features  
b) Hiding the important data  
c) Hiding the implementation  
d) Showing the important data  
View Answer

Answer: a  
Explanation: It includes hiding the implementation part and showing only the required data and features to the user. It is done to hide the implementation complexity and details from the user. And to provide a good interface in programming.

19. Which among the following can show polymorphism?  
a) Overloading &&  
b) Overloading <<  
c) Overloading ||  
d) Overloading +=  
View Answer

Answer: b  
Explanation: Only insertion operator can be overloaded among all the given options. And the polymorphism can be illustrated here only if any of these is applicable of being overloaded. Overloading is type of polymorphism.

20. In which access should a constructor be defined, so that object of the class can be created in any function?  
a) Any access specifier will work  
b) Private  
c) Public  
d) Protected  
View Answer

Answer: c  
Explanation: Constructor function should be available to all the parts of program where the object is to be created. Hence it is advised to define it in public access, so that any other function is able to create objects.

21. Which among the following is correct for the class defined below?

class student

{

int marks;

public: student(){}

student(int x)

{

marks=x;

}

};

main()

{

student s1(100);

student s2();

student s3=100;

return 0;

}

a) Program will give compile time error  
b) Object s3, syntax error  
c) Only object s1 and s2 will be created  
d) Program runs and all objects are created  
View Answer

Answer: d  
Explanation: It is a special case of constructor with only 1 argument. While calling a constructor with one argument, you are actually implicitly creating a conversion from the argument type to the type of class. Hence you can directly specify the value of that one argument with assignment operator.

22. The copy constructors can be used to \_\_\_\_\_\_\_\_  
a) Copy an object so that it can be passed to another primitive type variable  
b) Copy an object for type casting  
c) Copy an object so that it can be passed to a function  
d) Copy an object so that it can be passed to a class  
View Answer

Answer: c  
Explanation: When an object is passed to a function, actually its copy is made in the function. To copy the values, copy constructor is used. Hence the object being passed and object being used in function are different.

23. Which constructor will be called from the object obj2 in the following C++ program?

class A

{

int i;

A()

{

i=0;

}

A(int x)

{

i=x+1;

}

A(int y, int x)

{

i=x+y;

}

};

A obj1(10);

A obj2(10,20);

A obj3;

a) A(int y, int x)  
b) A(int y; int x)  
c) A(int y)  
d) A(int x)  
View Answer

Answer: a  
Explanation: The two argument constructor will be called as we are passing 2 arguments to the object while creation. The arguments will be passed together and hence compiler resolves that two argument constructor have to be called.

24. Which among the following represents correct constructor?  
a) –classname()  
b) classname()  
c) ()classname  
d) ~classname()  
View Answer

Answer: b  
Explanation: The constructors must contain only the class name. The class name is followed by the blank parenthesis or we can have parameters if some values are to be passed.

25. What happens when an object is passed by reference?  
a) Destructor is called at end of function  
b) Destructor is called when called explicitly  
c) Destructor is not called  
d) Destructor is called when function is out of scope  
View Answer

Answer: c  
Explanation: The destructor is never called in this situation. The concept is that when an object is passed by reference to the function, the constructor is not called, but only the main object will be used. Hence no destructor will be called at end of function.

26. Which access specifier is usually used for data members of a class?  
a) Protected  
b) Private  
c) Public  
d) Default  
View Answer

Answer: b  
Explanation: All the data members should be made private to ensure the highest security of data. In special cases we can use public or protected access, but it is advised to keep the data members private always.

27. How to access data members of a class?  
a) Dot, arrow or direct call  
b) Dot operator  
c) Arrow operator  
d) Dot or arrow as required  
View Answer

Answer: d  
Explanation: The data members can never be called directly. Dot operator is used to access the members with help of object of class. Arrow is usually used if pointers are used.

28. Which feature of OOP reduces the use of nested classes?  
a) Inheritance  
b) Binding  
c) Abstraction  
d) Encapsulation  
View Answer

Answer: a  
Explanation: Using inheritance we can have the security of the class being inherited. The subclass can access the members of parent class. And have more feature than a nested class being used.

29. Which keyword among the following can be used to declare an array of objects in java?  
a) allocate  
b) arr  
c) new  
d) create  
View Answer

Answer: c  
Explanation: The keyword new can be used to declare an array of objects in java. The syntax must be specified with an object pointer which is assigned with a memory space containing the required number of object space. Even initialization can be done directly.

30. Which operator can be used to free the memory allocated for an object in C++?  
a) Unallocate  
b) Free()  
c) Collect  
d) delete  
View Answer

Answer: d  
Explanation: The delete operator in C++ can be used to free the memory and resources held by an object. The function can be called explicitly whenever required. In C++ memory management must be done by the programmer. There is no automatic memory management in C++.

31. Which of the following is not a property of an object?  
a) Properties  
b) Names  
c) Identity  
d) Attributes  
View Answer

Answer: b  
Explanation: The names are not property of an object. The identity can be in any form like address or name of object but name can’t be termed as only identity of an object. The objects contain attributes that define what type of data an object can store.

32. Which type of members can’t be accessed in derived classes of a base class?  
a) All can be accessed  
b) Protected  
c) Private  
d) Public  
View Answer

Answer: c  
Explanation: The private members can be accessed only inside the base class. If the class is derived by other classes. Those members will not be accessible. This concept of OOP is made to make the members more secure.

33. Which among the following best describes the Inheritance?  
a) Using the data and functions into derived segment  
b) Using already defined functions in a programming language  
c) Using the code already written once  
d) Copying the code already written  
View Answer

Answer: a  
Explanation: It can only be indicated by using the data and functions that we use in derived class, being provided by parent class. Copying code is nowhere similar to this concept, also using the code already written is same as copying. Using already defined functions is not inheritance as we are not adding any of our own features.

34. Single level inheritance supports \_\_\_\_\_\_\_\_\_\_\_\_\_ inheritance.  
a) Language independency  
b) Multiple inheritance  
c) Compile time  
d) Runtime  
View Answer

Answer: d  
Explanation: The runtime inheritance is done when object of a class is created to call a method. At runtime the function is searched if it is in class of object. If not, it will search in its parent classes and hierarchy for that method.

35. How to overcome diamond problem?  
a) Using seperate derived class  
b) Using virtual keyword with same name function  
c) Can’t be done  
d) Using alias name  
View Answer

Answer: b  
Explanation: To overcome the ambiguity and conflict we can use keyword virtual. This will help us to differentiate the functions with same name that came to last derived class in diamond problem.

36. Which keyword is used to declare virtual functions?  
a) virt  
b) virtually  
c) virtual  
d) anonymous  
View Answer

Answer: c  
Explanation: The virtual keyword is used to declare virtual functions. Anonymous keyword is used with classes and have a different meaning. The virtual functions are used to call the intended function of the derived class.

37. What happens if non static members are used in static member function?  
a) Executes fine  
b) Compile time error  
c) Executes if that member function is not used  
d) Runtime error  
View Answer

Answer: b  
Explanation: There must be specific memory space allocated for the data members before the static member functions uses them. But the space is not reserved if object is not declared. Hence only if static members are not used, it leads to compile time error.

38. What is friend member functions in C++?  
a) Non-member functions which have access to all the members (including private) of a class  
b) Member function which doesn’t have access to private members  
c) Member function which can modify any data of a class  
d) Member function which can access all the members of a class  
View Answer

Answer: a  
Explanation: A non-member function of a class which can access even the private data of a class is a friend function. It is an exception on access to private members outside the class. It is sometimes considered as a member functions since it has all the access that a member function in general have.

39. Where is the memory allocated for the objects?  
a) Cache  
b) ROM  
c) HDD  
d) RAM  
View Answer

Answer: d  
Explanation: The memory for the objects or any other data is allocated in RAM initially. This is while we run a program all the memory allocation takes place in some RAM segments. Arrays in heap and local members in stack etc.

40. Which of the following best describes member function overriding?  
a) Member functions having the same name in derived class only  
b) Member functions having the same name and different signature inside main function  
c) Member functions having the same name in base and derived classes  
d) Member functions having the same name in base class only  
View Answer

Answer: c  
Explanation: The member function which is defined in base class and again in the derived class, is overridden by the definition given in the derived class. This is because the preference is given more to the local members. When derived class object calls that function, definition from the derived class is used.

41. Encapsulation and abstraction differ as \_\_\_\_\_\_\_\_\_\_\_\_  
a) Hiding and hiding respectively  
b) Binding and Hiding respectively  
c) Hiding and Binding respectively  
d) Can be used any way  
View Answer

Answer: b  
Explanation: Abstraction is hiding the complex code. For example, we directly use cout object in C++ but we don’t know how is it actually implemented. Encapsulation is data binding, as in, we try to combine a similar type of data and functions together.

42. Which feature of OOP is exhibited by the function overriding?  
a) Polymorphism  
b) Encapsulation  
c) Abstraction  
d) Inheritance  
View Answer

Answer: a  
Explanation: The polymorphism feature is exhibited by function overriding. Polymorphism is the feature which basically defines that same named functions can have more than one functionalities.

43. How to access the private member function of a class?  
a) Using class address  
b) Using object of class  
c) Using object pointer  
d) Using address of member function  
View Answer

Answer: d  
Explanation: Even the private member functions can be called outside the class. This is possible if address of the function is known. We can use the address to call the function outside the class.

44. Which keyword should be used to declare static variables?  
a) const  
b) common  
c) static  
d) stat  
View Answer

Answer: c  
Explanation: The keyword used to declare static variables is static. This is must be used while declaring the static variables. The compiler can make variables static if and only if they are mentioned with static keyword.

45. Which is correct syntax for declaring pointer to object?  
a) \*className objectName;  
b) className\* objectName;  
c) className objectName();  
d) className objectName;  
View Answer

Answer: b  
Explanation: The syntax must contain \* symbol after the className as the type of object. This declares an object pointer. This can store address of any object of the specified class.

46. Which class/set of classes can illustrate polymorphism in the following C++ code?

abstract class student

{

public : int marks;

calc\_grade();

}

class topper:public student

{

public : calc\_grade()

{

return 10;

}

};

class average:public student

{

public : calc\_grade()

{

return 20;

}

};

class failed{ int marks; };

a) Only class student and topper together can show polymorphism  
b) Only class student can show polymorphism  
c) Class failed should also inherit class student for this code to work for polymorphism  
d) All class student, topper and average together can show polymorphism  
View Answer

Answer: d  
Explanation: Since Student class is abstract class and class topper and average are inheriting student, class topper and average must define the function named calc\_grade(); in abstract class. Since both the definition are different in those classes, calc\_grade() will work in different way for same input from different objects. Hence it shows polymorphism.

47. If data members are private, what can we do to access them from the class object?  
a) Private data members can never be accessed from outside the class  
b) Create public member functions to access those data members  
c) Create private member functions to access those data members  
d) Create protected member functions to access those data members  
View Answer

Answer: b  
Explanation: We can define public member functions to access those private data members and get their value for use or alteration. They can’t be accessed directly but is possible to be access using member functions. This is done to ensure that the private data doesn’t get modified accidentally.

48. Which among the following is not a necessary condition for constructors?  
a) Its name must be same as that of class  
b) It must not have any return type  
c) It must contain a definition body  
d) It can contains arguments  
View Answer

Answer: c  
Explanation: Constructors are predefined implicitly, even if the programmer doesn’t define any of them. Even if the programmer declares a constructor, it’s not necessary that it must contain some definition.

49. Object being passed to a copy constructor \_\_\_\_\_\_\_\_\_\_\_  
a) Must not be mentioned in parameter list  
b) Must be passed with integer type  
c) Must be passed by value  
d) Must be passed by reference  
View Answer

Answer: d  
Explanation: This is mandatory to pass the object by reference. Otherwise, the object will try to create another object to copy its values, in turn a constructor will be called, and this will keep on calling itself. This will cause the compiler to give out of memory error.

50. If in multiple inheritance, class C inherits class B, and Class B inherits class A. In which sequence are their destructors called if an object of class C was declared?  
a) ~A() then ~B() then ~C()  
b) ~C() then ~A() then ~B()  
c) ~C() then ~B() then ~A()  
d) ~B() then ~C() then ~A()  
View Answer

Answer: c  
Explanation: The destructors are always called in the reverse order of how the constructors were called. Here class A constructor would have been created first if Class C object is declared. Hence class A destructor is called at last.

51. Instance of which type of class can’t be created?  
a) Parent class  
b) Abstract class  
c) Anonymous class  
d) Nested class  
View Answer

Answer: b  
Explanation: Instance of abstract class can’t be created as it will not have any constructor of its own, hence while creating an instance of class, it can’t initialize the object members. Actually the class inheriting the abstract class can have its instance because it will have implementation of all members.

52. \_\_\_\_\_\_\_\_\_\_\_ underlines the feature of Polymorphism in a class.  
a) Virtual Function  
b) Inline function  
c) Enclosing class  
d) Nested class  
View Answer

Answer: a  
Explanation: Virtual Functions can be defined in any class using the keyword virtual. All the classes which inherit the class containing the virtual function, define the virtual function as required. Redefining the function on all the derived classes according to class and use represents polymorphism.

53. Which feature in OOP is used to allocate additional functions to a predefined operator in any language?  
a) Function Overloading  
b) Function Overriding  
c) Operator Overloading  
d) Operator Overriding  
View Answer

Answer: c  
Explanation: The feature is operator overloading. There is not a feature named operator overriding specifically. Function overloading and overriding doesn’t give addition function to any operator.

54. Which feature can be implemented using encapsulation?  
a) Polymorphism  
b) Overloading  
c) Inheritance  
d) Abstraction  
View Answer

Answer: d  
Explanation: Data abstraction can be achieved by using encapsulation. We can hide the operation and structure of actual program from the user and can show only required information by the user.

**Chapterwise Multiple Choice Questions on Object Oriented Programming**

Our 1000+ MCQs focus on all topics of the Object Oriented Programming subject, covering 100+ topics. This will help you to prepare for exams, contests, online tests, quizzes, viva-voce, interviews, and certifications. You can practice these MCQs chapter by chapter starting from the 1st chapter or you can jump to any chapter of your choice.

1. [OOPs Concept & Features](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#oops-concept-features)
2. [Constructors and Destructors](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#constructors-destructors)
3. [Access Specifiers](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#access-specifiers)
4. [Class Members & Types](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#class-members-types)
5. [Object](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#object)
6. [Classes](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#classes)
7. [Class Components (Using Java)](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#class-components-java)
8. [Types of Classes (Using Java)](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#types-classes-java)
9. [Inheritance & its Types](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#inheritance-types)
10. [Inheritance (Using Java)](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#inheritance-java)
11. [Polymorphism (Using Java)](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#polymorphism-java)
12. [Member Functions & its Types](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#member-functions-types)
13. [Exception Handling & Static Class Members](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#exception-handling-static-class-members)
14. [Assigning Object, Pointer to Objects, Passing and Returning Object](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#assigning-object-pointer-objects-passing-returning-object)
15. [Default Arguments vs Overloading, Upcasting and Downcasting](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#default-arguments-overloading-upcasting-downcasting)
16. [Memory Allocation & Scope of Variable](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#memory-allocation-scope-variable)
17. [Inbuilt Classes](https://www.sanfoundry.com/1000-object-oriented-programming-oops-questions-answers/#inbuilt-classes)

**1. MCQ on OOPs Concept & Features**

The section contains multiple choice questions and answers on oops basic concepts, objects, classes, oops features, polymorphism, encapsulation and abstraction.

|  |  |
| --- | --- |
|  [OOPs MCQ: Basics](https://www.sanfoundry.com/object-oriented-programming-questions-answers-online-test/)   [OOPs MCQ: Classes](https://www.sanfoundry.com/object-oriented-programming-questions-answers-classes/)   [OOPs MCQ: Objects](https://www.sanfoundry.com/object-oriented-programming-questions-answers-objects/)   [OOPs MCQ: Features](https://www.sanfoundry.com/object-oriented-programming-questions-answers-oops-features/) |  [OOPs MCQ: Polymorphism](https://www.sanfoundry.com/object-oriented-programming-questions-answers-polymorphism/)   [OOPs MCQ: Encapsulation](https://www.sanfoundry.com/object-oriented-programming-questions-answers-encapsulation/)   [OOPs MCQ: Abstraction](https://www.sanfoundry.com/object-oriented-programming-questions-answers-abstraction/) |

**2. OOPs MCQ on Constructors and Destructors**

The section contains questions and answers on constructors and its types, copy constructor, overloading constructors, destructors, execution of constructor or destructor.

|  |  |
| --- | --- |
|  [OOPs MCQ: Constructors](https://www.sanfoundry.com/object-oriented-programming-questions-answers-constructors/)   [OOPs MCQ: Types of Constructors](https://www.sanfoundry.com/object-oriented-programming-questions-answers-online-quiz/)   [OOPs MCQ: Copy Constructor](https://www.sanfoundry.com/object-oriented-programming-basic-questions-answers/) |  [OOPs MCQ: Overloading Constructors](https://www.sanfoundry.com/object-oriented-programming-questions-answers-overloading-constructors/)   [OOPs MCQ: Execution of Constructor or Destructor](https://www.sanfoundry.com/object-oriented-programming-questions-answers-experienced/)   [OOPs MCQ: Destructors](https://www.sanfoundry.com/object-oriented-programming-questions-answers-destructors/) |

**3. Object Oriented Programming MCQ on Access Specifiers**

The section contains MCQs on access specifiers, private, public and protected access Specifiers.

|  |  |
| --- | --- |
|  [OOPs MCQ: Access Specifiers](https://www.sanfoundry.com/object-oriented-programming-questions-answers-access-specifiers/)   [OOPs MCQ: Private Access Specifier](https://www.sanfoundry.com/object-oriented-programming-questions-answers-private-access-specifier/) |  [OOPs MCQ: Protected Access Specifier](https://www.sanfoundry.com/object-oriented-programming-questions-bank/)   [OOPs MCQ: Public Access Specifier](https://www.sanfoundry.com/object-oriented-programming-questions-answers-public-access-specifier/) |

**4. OOP MCQ on Class Members & Types**

The section contains multiple choice questions and answers on data members, member functions, local and nested class.

|  |  |
| --- | --- |
|  [Object Oriented Programming MCQ: Data Members](https://www.sanfoundry.com/object-oriented-programming-questions-answers-data-members/)   [Object Oriented Programming MCQ: Member Functions](https://www.sanfoundry.com/object-oriented-programming-questions-answers-member-functions/) |  [Object Oriented Programming MCQ: Local Class](https://www.sanfoundry.com/object-oriented-programming-questions-answers-local-class/)   [Object Oriented Programming MCQ: Nested Class](https://www.sanfoundry.com/object-oriented-programming-questions-answers-nested-class/) |

**5. OOPs MCQ on Object**

The section contains questions and answers on passing and returning object with functions, object reference and memory allocation, object array and usage.

|  |  |
| --- | --- |
|  [OOPs MCQ: Passing and Returning Object with Functions](https://www.sanfoundry.com/object-oriented-programming-interview-questions-answers/)   [OOPs MCQ: Object Reference](https://www.sanfoundry.com/object-oriented-programming-questions-answers-object-reference/)   [OOPs MCQ: Memory Allocation of Object](https://www.sanfoundry.com/object-oriented-programming-questions-answers-test/) |  [OOPs MCQ: Object Array](https://www.sanfoundry.com/object-oriented-programming-questions-answers-object-array/)   [OOPs MCQ: Object Use](https://www.sanfoundry.com/object-oriented-programming-questions-answers-mcqs/) |

**6. Object Oriented Programming MCQ on Classes**

The section contains MCQs on abstract class, template class, base class, derived class and class usage.

|  |  |
| --- | --- |
|  [Object Oriented Programming MCQ: Abstract Class](https://www.sanfoundry.com/object-oriented-programming-questions-answers-abstract-class/)   [Object Oriented Programming MCQ: Template Class](https://www.sanfoundry.com/object-oriented-programming-questions-answers-template-class/)   [Object Oriented Programming MCQ: Base Class](https://www.sanfoundry.com/object-oriented-programming-questions-answers-base-class/) |  [Object Oriented Programming MCQ: Derived Class](https://www.sanfoundry.com/object-oriented-programming-questions-answers-derived-class/)   [Object Oriented Programming MCQ: Class Use](https://www.sanfoundry.com/object-oriented-programming-questions-answers-class-use/) |

**7. OOPs Multiple Choice Questions on Class Components (Using Java)**

The section contains multiple choice questions and answers on data members, member functions, constructors, types of constructors, constructor overloading, copy constructors, assigning objects, the ‘this’ keyword, and memory allocation in Java programming.

|  |  |
| --- | --- |
|  [Data Members](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-data-members/)   [Member Functions](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-member-functions/)   [Constructors](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-constructors/)   [Types of Constructors](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-types-constructors/)   [Constructor Overloading](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-constructor-overloading/) |  [Copy Constructors](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-copy-constructors/)   [Assigning Objects](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-assigning-objects/)   [this Keyword](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-this-keyword/)   [Memory Allocation](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-memory-allocation/) |

**8. OOPs Multiple Choice Questions on Types of Classes (Using Java)**

The section contains questions and answers on object class, base class, derived class, inner class, static class, final class, generic class, nested class, wrapper class, and collection classes using Java programming.

|  |  |
| --- | --- |
|  [Object Class](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-object-class/)   [Base Class](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-base-class/)   [Derived Class](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-derived-class/)   [Inner Class](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-inner-class/)   [Static Class](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-static-class/) |  [Final Class](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-final-class/)   [Generic Class](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-generic-class/)   [Nested Class](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-nested-class/)   [Wrapper Class](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-wrapper-class/)   [Collection Classes](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-collection-classes/) |

**9. OOPs Multiple Choice Questions on Inheritance & its Types**

The section contains multiple choice questions and answers on inheritance and its types, single and multilevel inheritance, multiple inheritance, hierarchical and hybrid inheritance.

|  |  |
| --- | --- |
|  [OOPs MCQ: Inheritance](https://www.sanfoundry.com/object-oriented-programming-questions-answers-inheritance/)   [OOPs MCQ: Types of Inheritance](https://www.sanfoundry.com/object-oriented-programming-questions-answers-types-inheritance/)   [OOPs MCQ: Single Level Inheritance](https://www.sanfoundry.com/object-oriented-programming-questions-answers-single-level-inheritance/)   [OOPs MCQ: Multilevel Inheritance](https://www.sanfoundry.com/object-oriented-programming-questions-answers-multilevel-inheritance/) |  [OOPs MCQ: Multiple Inheritance](https://www.sanfoundry.com/object-oriented-programming-questions-answers-multiple-inheritance/)   [OOPs MCQ: Hierarchical Inheritance](https://www.sanfoundry.com/object-oriented-programming-questions-answers-hierarchical-inheritance/)   [OOPs MCQ: Hybrid Inheritance](https://www.sanfoundry.com/object-oriented-programming-questions-answers-hybrid-inheritance/) |

**10. OOPs MCQ on Inheritance (Using Java)**

The section contains OOPs MCQs using Java on inheritance types, including single-level inheritance, multilevel inheritance, hierarchical inheritance, and hybrid inheritance.

|  |  |
| --- | --- |
|  [Types of Inheritance](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-types-inheritance/)   [Single Level Inheritance](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-single-level-inheritance/)   [Multilevel Inheritance](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-multilevel-inheritance/) |  [Hierarchical Inheritance](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-hierarchical-inheritance/)   [Hybrid Inheritance](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-hybrid-inheritance/) |

**11. OOPs MCQ on Polymorphism (Using Java)**

The section contains multiple choice questions and answers on method overloading and method overriding.

|  |  |
| --- | --- |
|  [Method Overloading](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-method-overloading/) |  [Method Overriding](https://www.sanfoundry.com/object-oriented-programming-using-java-questions-answers-method-overriding/) |

**12. OOP MCQ on Member Functions & its Types**

The section contains questions and answers on virtual and abstract functions, member function types, member operator function, overloading and overriding member functions, constant, private and public member functions.

|  |  |
| --- | --- |
|  [Object Oriented Programming MCQ: Virtual Functions](https://www.sanfoundry.com/object-oriented-programming-questions-answers-entrance-exams/)   [Object Oriented Programming MCQ: Abstract Function](https://www.sanfoundry.com/object-oriented-programming-questions-answers-abstract-function/)   [Object Oriented Programming MCQ: Types of Member Functions](https://www.sanfoundry.com/object-oriented-programming-problems/)   [Object Oriented Programming MCQ: Member Operator Function](https://www.sanfoundry.com/object-oriented-programming-questions-answers-aptitude-test/)   [Object Oriented Programming MCQ: Overloading Member Functions](https://www.sanfoundry.com/object-oriented-programming-questions-answers-freshers/) |  [Object Oriented Programming MCQ: Overriding Member Functions](https://www.sanfoundry.com/object-oriented-programming-interview-questions-answers-freshers/)   [Object Oriented Programming MCQ: Constant Member Functions](https://www.sanfoundry.com/object-oriented-programming-questions-answers-quiz/)   [Object Oriented Programming MCQ: Private Member Functions](https://www.sanfoundry.com/object-oriented-programming-multiple-choice-questions-answers/)   [Object Oriented Programming MCQ: Public Member Functions](https://www.sanfoundry.com/object-oriented-programming-questions-answers-public-member-functions/) |

**13. OOPs MCQ on Exception Handling & Static Class Members**

The section contains MCQs on exception handling, catching class types, static data members and static member functions.

|  |  |
| --- | --- |
|  [OOPs MCQ: Exception Handling](https://www.sanfoundry.com/object-oriented-programming-questions-answers-exception-handling/)   [OOPs MCQ: Catching Class Types](https://www.sanfoundry.com/object-oriented-programming-questions-answers-catching-class-types/) |  [OOPs MCQ: Static Data Members](https://www.sanfoundry.com/object-oriented-programming-questions-answers-static-data-members/)   [OOPs MCQ: Static Member Functions](https://www.sanfoundry.com/object-oriented-programming-questions-answers-static-member-functions/) |

**14. OOPs MCQ on Assigning Object, Pointer to Objects, Passing and Returning Object**

The section contains multiple choice questions and answers on passing objects to functions, returning and assigning objects, pointers to objects and this pointer.

|  |  |
| --- | --- |
|  [OOPs MCQ: Passing Object to Functions](https://www.sanfoundry.com/object-oriented-programming-interview-questions-answers-experienced/)   [OOPs MCQ: Returning Objects](https://www.sanfoundry.com/object-oriented-programming-questions-answers-returning-objects/)   [OOPs MCQ: Assigning Objects](https://www.sanfoundry.com/object-oriented-programming-questions-answers-assigning-objects/) |  [OOPs MCQ: Pointer to Objects](https://www.sanfoundry.com/object-oriented-programming-assessment-questions-answers/)   [OOPs MCQ: This Pointer](https://www.sanfoundry.com/object-oriented-programming-questions-answers-pointer/) |

**15. OOPs MCQ on Default Arguments vs Overloading, Upcasting and Downcasting**

The section contains questions and answers on default arguments, constructors overloading, upcasting and downcasting.

|  |  |
| --- | --- |
|  [OOPs MCQ: Default Arguments](https://www.sanfoundry.com/object-oriented-programming-questions-answers-default-arguments/)   [OOPs MCQ: Constructors Overloading](https://www.sanfoundry.com/object-oriented-programming-questions-answers-constructors-overloading/) |  [OOPs MCQ: Upcasting](https://www.sanfoundry.com/object-oriented-programming-questions-answers-upcasting/)   [OOPs MCQ: Downcasting](https://www.sanfoundry.com/object-oriented-programming-questions-answers-downcasting/) |

**16. OOPs MCQ on Memory Allocation & Scope of Variable**

The section contains MCQs on new and delete operators, automatic and extern variables.

|  |  |
| --- | --- |
|  [OOPs MCQ: New Operator](https://www.sanfoundry.com/object-oriented-programming-questions-answers-new-operator/)   [OOPs MCQ: Delete Operator](https://www.sanfoundry.com/object-oriented-programming-questions-answers-delete-operator/) |  [OOPs MCQ: Automatic Variable](https://www.sanfoundry.com/object-oriented-programming-questions-answers-automatic-variable/)   [OOPs MCQ: Extern Variable](https://www.sanfoundry.com/object-oriented-programming-questions-answers-campus-interviews/) |

**17. Object Oriented Programming MCQ on Inbuilt Classes**

The section contains multiple choice questions and answers on inbuilt classes, io and string class.

|  |  |
| --- | --- |
|  [OOPs MCQ: Inbuilt Classes](https://www.sanfoundry.com/object-oriented-programming-questions-answers-inbuilt-classes/)   [OOPs MCQ: IO Class](https://www.sanfoundry.com/object-oriented-programming-questions-answers-io-class/) |  [OOPs MCQ: String Class](https://www.sanfoundry.com/object-oriented-programming-questions-answers-string-class/) |

If you would like to learn "Object Oriented Programming" thoroughly, you should attempt to work on the complete set of 1000+ MCQs - multiple choice questions and answers mentioned above. It will immensely help anyone trying to crack an exam or an interview.