OOPS MCQ

1) Which of the following language was developed as the first purely object programming language?

a. SmallTalk
b. C++
c. Kotlin
d. Java

Hide Answer **Workspace**

Answer: a. SmallTalk

Explanation: This programming language was invented as the first pure OOPS (object-oriented) language. This language was designed by Alan Kay in the early 1970s.

2) Who developed object-oriented programming?

a. Adele Goldberg
b. Dennis Ritchie
c. Alan Kay
d. Andrea Ferro

Hide Answer **Workspace**

Answer: c. Alan Kay

Explanation: In the year 1970, Alan Kay gave Object-Oriented programming. He coined the concept of OOPS at a grad school in the year 1966 or 1967. Alan kay, Adele Goldberg, Dan Ingalls and others developed the first Smalltalk programming language, which follows the OOPS concept.

3) Which of the following is not an OOPS concept?

a. Encapsulation

b. Polymorphism

c. Exception

d. Abstraction

Hide Answer **Workspace**

Answer: c. Exception

Explanation: None.

4) Which feature of OOPS described the reusability of code?

a. Abstraction

b. Encapsulation

c. Polymorphism

d. Inheritance

Hide Answer **Workspace**

Answer: d. Inheritance

Explanation: Inheritance is the feature of OOPS, which allows the users of OOPS to reuse the code which is already written. This OOPS feature inherits the features of another class in the programs. This mechanism actually inherits the fields and methods of the superclass.

5) Which of the following language supports polymorphism but not the classes?

a. C++ programming language

b. Java programming language

c. Ada programming language

d. C# programming language

Hide Answer **Workspace**

Answer: c. Ada programming language

Explanation: It is a programming language that disapproves of the concept of polymorphism but supports the concept of classes. It is an object-based language. So, it does not follow the Object-oriented programming concepts.

6) Which among the following feature is not in the general definition of OOPS?

a. Modularity
b. Efficient Code
c. Code reusability
d. Duplicate or Redundant Data

Hide Answer **Workspace**

Answer: d. Duplicate or Redundant Data

Explanation: Duplicacy or Redundancy of data is a feature which totally dependent on the programmers. So, it cannot be created by the OOPS.

7) Which feature of OOPS derives the class from another class?

a. Inheritance
b. Data hiding
c. Encapsulation
d. Polymorphism

Hide Answer **Workspace**

Answer: a. Inheritance

Explanation: Inheritance is an important OOPS feature which derives the class from the base class or superclass. This OOPS feature inherits the features of another class in the programs. This mechanism actually inherits the fields and methods of the superclass.

8) Define the programming language, which does not support all four types of inheritance?

a. Smalltalk

b. Kotlin

c. Java

d. C++

Hide Answer **Workspace**

Answer: c. Java

Explanation: Java is a programming language that disapproves of the concept of 'multiple inheritance'. So, it does not agree with all types of inheritance. But, we can implement 'multiple inheritance' in Java language using the interface concept.

9) A single program of OOPS contains _______ classes?

a. Only 1

b. Only 999

c. Only 100

d. Any number

Hide Answer **Workspace**

Answer: d. Any number

Explanation: We can define any number of classes with different names in a single program of OOPS.

10) Which operator from the following can be used to illustrate the feature of polymorphism?

a. Overloading <<
b. Overloading &&
c. Overloading ||
d. Overloading +=

Hide Answer **Workspace**

Answer: a. Overloading <<

Explanation: << is an insertion operator which is used for overloading (polymorphism).

11) Which two features of object-oriented programming are the same?

a. Abstraction and Polymorphism features are the same
b. Inheritance and Encapsulation features are the same
c. Encapsulation and Polymorphism features are the same
d. Encapsulation and Abstraction

Hide Answer **Workspace**

Answer: d. Encapsulation and Abstraction

Explanation: Encapsulation and Abstraction are the same OOPS concepts. Encapsulation hides the features of the object and binds all the properties inside a single class. And abstraction is a feature that shows the required data to the user.

12) Which header file is required by the C++ programming language to use the OOPS concept?

a. stdio.h
b. iostream.h

c. stdlib.h

d. We can easily use the OOPS concepts in c++ programs without using any header file.

Hide Answer **Workspace**

Answer: d. We can easily use the OOPS concepts in c++ programs without using any header file.

Explanation: There is no need to use any particular header file for using the OOPS concept in the C++ programs. The C++ functions and variables have their respective header files, which should be defined in the program.

13) Which of the following definition is incorrect for polymorphism?

a. Polymorphism helps in redefining the same functionality

b. Polymorphism concept is the feature of object-oriented programming (OOP)

c. It always increases the overhead of function definition

d. Ease in the readability of the program

Hide Answer **Workspace**

Answer: c. It always increases the overhead of function definition

Explanation: This concept of OOPS never increases the overhead of function definition.

14) Which among the following cannot be used for the concept of polymorphism?

a. Static member function

b. Constructor Overloading

c. Member function overloading

d. Global member function

Hide Answer **Workspace**

Answer: a. Static member function

Explanation: These functions are not an object property. That's why they cannot be acceptable for overriding or overloading.

15) Which function best describe the concept of polymorphism in programming languages?

a. Class member function
b. Virtual function
c. Inline function
d. Undefined function

Hide Answer **Workspace**

Answer: b. Virtual function

Explanation: Only those functions are used to achieve the polymorphism, which are declared as 'virtual'. These functions let the OOPS programs decide at runtime which function is to be called by the pointer.

16) Which member function is assumed to call first when there is a case of using function overloading or abstract class?

a. Global function
b. Local function
c. Function with lowest priority
d. Function with the highest priority

Hide Answer **Workspace**

Answer: d. Function with the highest priority

Explanation: The member function with the highest priority is called first when there is function overloading, or abstract class is used.

17) Is it true to use polymorphism in the C programming language?

a. True
b. False

Hide Answer **Workspace**

Answer: a. True

Explanation: Yes, we can use the concept of polymorphism in the C programming language. Users can use structures and then declare pointers in C programming language, which in turn points to some function.

Here, he/she can easily simulate the functions which are not exactly member function. Hence, users can manipulate the concept of polymorphism in the C language.

18) Which of the following language uses the classes but not the polymorphism concept?

a. Procedure Oriented language
b. Object-based language
c. Class-based language
d. If classes are used, then the polymorphism concept will always be used in the programming languages.

Hide Answer **Workspace**

Answer: b. Object-based language

Explanation: Object-based languages are those languages which support the definition of classes but not the polymorphism. Inheritance is also another feature which is not supported by these languages.

19) Which of the following OOP concept is not true for the C++ programming language?

a. A class must have member functions
b. C++ Program can be easily written without the use of classes
c. At least one instance should be declared within the C++ program
d. C++ Program must contain at least one class

Hide Answer **Workspace**

Answer: d. C++ Program must contain at least one class

Explanation: Use of class in the C++ language is not mandatory. Programs can be easily written without the use of classes. C++ programs classes may or may not contain the member function. That's why it is not a necessary situation in the C++ language.

20) What is the extra feature in classes which was not in the structures?

a. Member functions
b. Data members
c. Public access specifier
d. Static Data allowed

Hide Answer **Workspace**

Answer: a. Member functions

Explanation: Member function is an extra feature which is allowed in class but not in the concept of structures.

21) How many types of polymorphism in the C++ programming language?

a. Three types of polymorphism
b. Two types of polymorphism
c. Five types of polymorphism
d. Four types of polymorphism

Hide Answer **Workspace**

Answer: b. Two types of polymorphism

Explanation: C++ programming language has two types of polymorphism: 1. Runtime Polymorphism 2. Compile-time Polymorphism

Runtime Polymorphism: It is meet by the function overriding. This polymorphism is also known as late or dynamic binding.

Compile-time Polymorphism: It is meet by the operator and function overloading. This polymorphism is also known as early or static binding.

22) Which of the following feature is also known as run-time binding or late binding?

a. Dynamic typing
b. Dynamic loading
c. Dynamic binding
d. Data hiding

Hide Answer **Workspace**

Answer: c. Dynamic binding

Explanation: Dynamic binding or runtime binding or late binding is that type of binding which happens at the execution time of the program or code. Function or method overriding is the perfect example of this type of binding. Virtual functions are used to achieve the concept of function overriding.

23) Which among the following is not a member of the class?

a. Virtual function
b. const function
c. Static function
d. Friend function

Hide Answer **Workspace**

Answer: d. Friend function

Explanation: The public member functions of a class can easily access the private data members of the same class.

This is achieved by the "friend", which is a non-member function to the class. Its private data can be accessed. That's why the friend function is not a member of the class.

24) Which of the following class is known as the generic class?

a. Final class
b. Template class
c. Abstract class
d. Efficient code

Hide Answer **Workspace**

Answer: b. Template class

Explanation: Template classes are those classes which can be used for any value of data type. So, these are known as a generic class.

Template classes help in making the genetic classes and generate the objects of classes based on the parameters. This type of class also saves system memory.

25) Which operator overloads using the friend function?

a. *

b. ( )

c. ->

d. =

Hide Answer **Workspace**

Answer: a. *

Explanation: The operators (->, ( ), =) cannot be overloaded using the friend function because if they are overloaded, then the code will show the compilation error. That's why * (asterisk) is a symbol that can be overloaded using the friend function.

26) Which of the following OOP concept binds the code and data together and keeps them secure from the outside world?

a. Polymorphism

b. Inheritance

c. Abstraction

d. Encapsulation

Hide Answer **Workspace**

Answer: d. Encapsulation

Explanation: Encapsulation is an important concept of Object-oriented programming. This concept binds the data and methods in a single unit. It binds the methods which manipulate the data.

27) Which member of the superclass is never accessible to the subclass?

a. Public member
b. Protected member
c. Private member
d. All of the mentioned

Hide Answer **Workspace**

Answer: c. Private member

Explanation: Private is a member of a class which never be accessible to the derived class.

28) What is the size of a class?

a. Sum of the size of all inherited variables along with the variables of the same class
b. The size of the class is the largest size of the variable of the same class
c. Classes in the programming languages do not have any size
d. Sum of the size of all the variables within a class.

Hide Answer **Workspace**

Answer: c. Classes in the programming languages do not have any size

Explanation: The object size defines the size of the class. That's why classes do not have any size.

29) Which class cannot create its instance?

a. Parent class
b. Nested class
c. Anonymous class
d. Abstract class

Hide Answer **Workspace**

Answer: d. Abstract class

Explanation: The instance of the abstract class can't be created because it will not have any constructor of its own. Hence while creating an instance of a class, it can't initialize the object members.

30) Encapsulation adds the function in a user-defined structure.

a. True
b. False

Hide Answer **Workspace**

Answer: b. False

Explanation: The encapsulation concept does not add the function in a user-defined structure because if the member functions are involved, then the user cannot call the structures defined by it.

31) Which of the following variable violates the definition of encapsulation?

a. Array variables
b. Local variables
c. Global variables
d. Public variables

Hide Answer **Workspace**

Answer: c. Global variables

Explanation: Global variable is a variable which violates the definition of encapsulation. The definition of encapsulation defines that the data or variable should be accessed by the specified set of elements. But, the global variable is accessible everywhere in the code. This type of variable does not hide the internal performance of the code. That's global variable violates the encapsulation concept.

32) How can the concept of encapsulation be achieved in the program?

a. By using the Access specifiers

b. By using the concept of Abstraction

c. By using only private members

d. By using the concept of Inheritance

Hide Answer **Workspace**

Answer: a. By using the Access specifiers

Explanation: Users can achieve the concept of encapsulation by implementing the access specifiers in the code. It is not compulsory that the user use only private members.

33) The concept of encapsulation helps in writing which type of classes in the Java programming language?

a. Abstract classes

b. Wrapper classes

c. Mutable classes

d. Immutable classes

Hide Answer **Workspace**

Answer: d. Immutable classes

Explanation: The concept of encapsulation helps in writing immutable classes in the Java programming language because these classes are used for caching purposes.

34) Encapsulation is_____?

a. technique of combining more than one member functions into a single unit.
b. mechanism of combining more than one data member into a single unit.
c. mechanism of combining more than one data members and member functions that implement on those data members into a single unit
d. technique of combining more than one data members and member functions into a single unit, which can manipulate any data.

Hide Answer **Workspace**

Answer: c. mechanism of combining more than one data members and member functions that implement on those data members into a single unit

Explanation: Encapsulation is an OOPS mechanism that combines more than one data member and member function and implements the member functions on those data members into a single unit.

35) Which of the following statement of a program is not right?

a. class teacher{ }; teacher s[5];
b. class teacher{ }s;
c. class teacher{ }; teacher s;
d. class teacher{ }s[];

Hide Answer **Workspace**

Answer: d. class teacher{ }s[];

Explanation: This statement is false because it must be compulsory to specify the array with its size.

36) Which of the following syntax is incorrect for the class definition?

a. student class{ };

b. class student{ student(int a){} };

c. class teacher{ public: teacher(int a){ } };

d. None of the mentioned

Hide Answer **Workspace**

Answer: a. student class{ };

Explanation: In this syntax, the keyword 'class' is present after the student. That's why this syntax is incorrect.

37) The object cannot be________?

a. passed by copy

b. passed as function

c. passed by value

d. passed by reference

Hide Answer **Workspace**

Answer: b. passed as function

Explanation: The object cannot be passed as the function because it is an instance of a class. Only the objects passed by the value, reference or copy.

38) Which among the following feature does not come under the concept of OOPS?

a. Data binding
b. Data hiding
c. Platform independent
d. Message passing

Hide Answer **Workspace**

Answer: c. Platform independent

Explanation: Platform independence is a feature which does not come under the OOPS concepts. This feature depends on the programming language. C++ is an object-oriented programming language which is not a platform-independent language.

39) Which of the following feature may be breaked if the user does not use the classes in the code?

a. Object must be used violated
b. Only the encapsulation concept is violated
c. Inheritance cannot be implemented
d. Basically, all the features of OOPS get violated

Hide Answer **Workspace**

Answer: d. Basically, all the features of OOPS get violated

Explanation: If the user does not use the classes in the code, then all the OOP features get violated. And, the encapsulation and inheritance cannot be implemented without the use of classes.

40) Which of the following feature interacts one object with another object?

a. Message reading
b. Message passing
c. Data transfer
d. Data binding

Hide Answer **Workspace**

Answer: b. Message passing

Explanation: Message passing is a feature of OOP that connects one object with another object.

It is a concept of communication for passing the messages between the two objects. It allows objects to send and receive information among them.

41) Which definition best defines the concept of abstraction?

a. Hides the important data
b. Hides the implementation and showing only the features
c. Hiding the implementation
d. Showing the important data

Hide Answer **Workspace**

Answer: b. Hides the implementation and showing only the features

Explanation: It hides the implementation part and showing only the data and features which are required to the end-user.

This technique is used to hide the complexity of code and its details from the user. It is also used to define a good interface in coding.

42) The combination of abstraction of the data and code is viewed in_______.

a. Inheritance

b. Object

c. Class

d. Interfaces

Hide Answer **Workspace**

Answer: b. Object

Explanation: Object is a technique which can be viewed as an abstraction of the combination of data and code. This OOPS concept uses the class data members and their functioning as the data abstraction. Code abstraction is used as the object of an inbuilt class.

43) The principle of abstraction__________

a. is used to achieve OOPS.

b. is used to avoid duplication

c. Use abstraction at its minimum

d. is used to remove longer codes

Hide Answer **Workspace**

Answer: b. is used to avoid duplication

Explanation: The principle of abstraction is used to avoid the code duplicacy. In this technique, the program does not contain any redundant functions. And it also makes the program effective.

44) Which among the following concept is correct if a user using the concept of encapsulation in a code?

a. The modification of the code can be additional overhead

b. Member functions can be used for modifying the data type of data members

c. The data type of data member cannot be modified.

d. The data type of the data member can be easily modified without modifying any other code

Hide Answer **Workspace**

Answer: d. The data type of the data member can be easily modified without modifying any other code

Explanation: Data type of a data member can be easily modified without modifying any code. Member functions can never modify the data type of the data members in the same class.

45) Using the concept of encapsulation security of the data is __________

a. Ensured to some extent

b. Purely ensured

c. Not ensured

d. Very low

Hide Answer **Workspace**

Answer: a. Ensured to some extent

Explanation: The concept of encapsulation can only ensure data security to some extent.

46) Consider the following Java program and select the right option from the given options.

```
class marksofstudent
{
    int subjectmarks;
    public : int* fun()
    {
        return &subjectmarks;
    }
};
main()
{
   marksofstudent s;
   int *ptr = c.fun() ;
   return 0;
}
```

a. The above program violates the feature of encapsulation

b. The above program may result in undesirable conditions

c. The above program will generate an error

d. The above program is good to go

Hide Answer **Workspace**

Answer: a. The above program violates the feature of encapsulation

Explanation: The code in this question violates the definition of encapsulation.

47) The name of the default access specifier for the member functions or data members in the C++ programming language is________.

a. Private access specifier

b. Public access specifier

c. Protected access specifier

d. Depends on compiler

Hide Answer **Workspace**

Answer: a. Private access specifier

Explanation: If none of the access specifiers is used, then the data members and member functions are Private by default in C++ classes. This specifier is actually used to increase the privacy of information.

48) Which of the following option best illustrates a friend class?

a. This class can access and manipulate all the private members of that class which connects to a friend.
b. Friend class can only access and manipulate the protected data members of that class that connects to a friend.
c. Friend class can't access any data member of another class but can use its methods
d. Friend class don't have any implementation

Hide Answer **Workspace**

Answer: a. Friend class can access and manipulate all the private members of the class, of which it is a friend

Explanation: A friend class can easily access all the private data members of another class.

49) Which of the following definition best describes the concept of polymorphism?

a. It is the ability to process the many messages and data in one way
b. It is the ability to process the undefined messages or data in at least one way
c. It is the ability to process the message or data in more than one form
d. It is the ability to process the message or data in only one form

Hide Answer **Workspace**

Answer: c. It is the ability for a message or data to be processed in more than one form.

Explanation: None.

50) Which class/ or set of classes can describe the concept of polymorphism in the following code?

```
abstract class student_details
{
  public : int marksofstudent;
  calculate_grade();
}
class topper:public student_details
{
   public : calculate_grade()
   {
      return 15;
   }
};
class average:public student_details
{
    public : calculate_grade()
    {
       return 30;
    }
};
class failed{ int marksofstudent; };
```

a. Only the student_details class can show the concept of polymorphism

b. The class which is 'failed' should also inherit class student for this code to work for polymorphism

c. The student_details, topper and average classes together can show the concept of polymorphism

d. Only the student_details and topper class together can show the concept of polymorphism

Hide Answer **Workspace**

Answer: c. The student_details, topper and average classes together can show the concept of polymorphism

Explanation: None.

← Prev Next →

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share

Learn Latest Tutorials

Splunk	SPSS	Swagger	Transact-SQL
Tumblr	ReactJS	Regex	

Reinforcement Learning

R Programming

RxJS

React Native

Python Design Patterns

Python Pillow

Python Turtle

Keras

Preparation

Aptitude

Reasoning

Verbal Ability

Interview Questions

Company Questions

Trending Technologies

Artificial Intelligence

Artificial Intelligence

AWS Tutorial

AWS

Selenium tutorial

Selenium

Cloud Computing

Cloud Computing

Hadoop tutorial

Hadoop

ReactJS Tutorial

ReactJS

Data Science Tutorial

Data Science

Angular 7 Tutorial

Angular 7

Blockchain Tutorial

Blockchain

Git Tutorial

Git

Machine Learning Tutorial

Machine Learning

DevOps Tutorial

DevOps

B.Tech / MCA

DBMS tutorial

DBMS

Data Structures tutorial

Data Structures

DAA tutorial

DAA

Operating System

Operating System

Computer Network tutorial

Computer Network

Compiler Design tutorial

Compiler Design

Computer Organization and Architecture

Computer Organization

Discrete Mathematics Tutorial

Discrete Mathematics

Ethical Hacking

Ethical Hacking

Computer Graphics Tutorial

Computer Graphics

Software Engineering

Software Engineering

html tutorial

Web Technology

Cyber Security tutorial

Cyber Security

Automata Tutorial

Automata

C Language tutorial

C Programming

C++ tutorial

C++

Java tutorial

Java

.Net Framework tutorial

.Net

Python tutorial

Python

List of Programs

Programs

Control Systems tutorial

Control System

Data Mining Tutorial

Data Mining

Data Warehouse Tutorial

Data Warehouse