OOPS MCQ Sample Questions

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

1. SmallTalk
2. C++
3. Kotlin
4. Java

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?

1. Adele Goldberg
2. Dennis Ritchie
3. Alan Kay
4. Andrea Ferro

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?

1. Encapsulation
2. Polymorphism
3. Exception
4. Abstraction

Answer: c. Exception

Explanation: None.

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

1. Abstraction
2. Encapsulation
3. Polymorphism
4. Inheritance

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

1. C++ programming language
2. Java programming language
3. Ada programming language
4. C# programming language

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?

1. Modularity
2. Efficient Code
3. Code reusability
4. Duplicate or Redundant Data

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?

1. Inheritance
2. Data hiding
3. Encapsulation
4. Polymorphism

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?

1. Smalltalk
2. Kotlin
3. Java
4. C++

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?

1. Only 1
2. Only 999
3. Only 100
4. Any number

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?

1. Overloading <<
2. Overloading &&
3. Overloading | |
4. Overloading +=

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?

1. Abstraction and Polymorphism features are the same
2. Inheritance and Encapsulation features are the same
3. Encapsulation and Polymorphism features are the same
4. Encapsulation and Abstraction

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?

1. stdio.h
2. iostream.h
3. stdlib.h
4. We can easily use the OOPS concepts in c++ programs without using any header file.

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?

1. Polymorphism helps in redefining the same functionality
2. Polymorphism concept is the feature of object-oriented programming (OOP)
3. It always increases the overhead of function definition
4. Ease in the readability of the program

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?

1. Static member function
2. Constructor Overloading
3. Member function overloading
4. Global member function

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?

1. Class member function
2. Virtual function
3. Inline function
4. Undefined function

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?

1. Global function
2. Local function
3. Function with lowest priority
4. Function with the highest priority

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?

1. True
2. False

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?

1. Procedure Oriented language
2. Object-based language
3. Class-based language
4. If classes are used, then the polymorphism concept will always be used in the programming languages.

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?

1. A class must have member functions
2. C++ Program can be easily written without the use of classes
3. At least one instance should be declared within the C++ program
4. C++ Program must contain at least one class

---

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

1. Member functions
2. Data members
3. Public access specifier

4. Static Data allowed

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?

1. Three types of polymorphism
2. Two types of polymorphism
3. Five types of polymorphism
4. Four types of polymorphism

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?

1. Dynamic typing
2. Dynamic loading
3. Dynamic binding
4. Data hiding

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?

1. Virtual function
2. const function
3. Static function
4. Friend function

---

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

1. Final class
2. Template class
3. Abstract class
4. Efficient code

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?

1. *
2. ( )
3. ->
4. =

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?

1. Polymorphism
2. Inheritance
3. Abstraction
4. Encapsulation

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

1. Public member
2. Protected member
3. Private member
4. All of the mentioned

28) What is the size of a class?

1. Sum of the size of all inherited variables along with the variables of the same class
2. The size of the class is the largest size of the variable of the same class
3. Classes in the programming languages do not have any size
4. Sum of the size of all the variables within a class.

29) Which class cannot create its instance?

1. Parent class
2. Nested class
3. Anonymous class
4. Abstract class

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

1. True
2. False

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

1. Array variables
2. Local variables
3. Global variables
4. Public variables

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

1. By using the Access specifiers
2. By using the concept of Abstraction
3. By using only private members
4. By using the concept of Inheritance

---

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

1. Abstract classes
2. Wrapper classes
3. Mutable classes
4. Immutable classes

---

34) Encapsulation is_____?

1. technique of combining more than one member functions into a single unit.
2. mechanism of combining more than one data member into a single unit.
3. mechanism of combining more than one data members and member functions that implement on those data members into a single unit
4. technique of combining more than one data members and member functions into a single unit, which can manipulate any data.

---

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

1. class teacher{ }; teacher s[5];
2. class teacher{ }s;
3. class teacher{ }; teacher s;
4. class teacher{ }s[];

---

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

1. student class{ };
2. class student{ student(int a){} };
3. class teacher{ public: teacher(int a){ } };
4. None of the mentioned

---

37) The object cannot be________?

1. passed by copy
2. passed as function
3. passed by value
4. passed by reference

---

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

1. Data binding
2. Data hiding
3. Platform independent
4. Message passing

---

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

1. Object must be used violated
2. Only the encapsulation concept is violated
3. Inheritance cannot be implemented
4. Basically, all the features of OOPS get violated

---

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

1. Message reading
2. Message passing
3. Data transfer
4. Data binding

---

41) Which definition best defines the concept of abstraction?

1. Hides the important data
2. Hides the implementation and showing only the features
3. Hiding the implementation
4. Showing the important data

---

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

1. Inheritance
2. Object
3. Class
4. Interfaces

---

43) The principle of abstraction___________

1. is used to achieve OOPS.
2. is used to avoid duplication
3. Use abstraction at its minimum
4. is used to remove longer codes

---

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

1. The modification of the code can be additional overhead
2. Member functions can be used for modifying the data type of data members
3. The data type of data member cannot be modified.
4. The data type of the data member can be easily modified without modifying any other code

---

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

1. Ensured to some extent
2. Purely ensured
3. Not ensured
4. Very low

---

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;
}
```

1. The above program violates the feature of encapsulation
2. The above program may result in undesirable conditions
3. The above program will generate an error
4. The above program is good to go

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

1. Private access specifier
2. Public access specifier
3. Protected access specifier
4. Depends on compiler

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

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

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

1. It is the ability to process the many messages and data in one way
2. It is the ability to process the undefined messages or data in at least one way
3. It is the ability to process the message or data in more than one form
4. It is the ability to process the message or data in only one form

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; };
```

1. Only the student_details class can show the concept of polymorphism
2. The class which is 'failed' should also inherit class student for this code to work for polymorphism
3. The student_details, topper and average classes together can show the concept of polymorphism
4. Only the student_details and topper class together can show the concept of polymorphism

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

Explanation: None.