

## **JAVA-OOP MCQS:**

1. Which of the following is not a principle of OOP?
  - a. Abstraction
  - b. Inheritance
  - c. Encapsulation
  - d. Normalization
  
2. Which keyword is used to create a new object in Java?
  - a. new
  - b. create
  - c. instantiate
  - d. object
  
3. Which of the following is an example of inheritance in Java?
  - a. A Car class inherits from a Vehicle class
  - b. A Vehicle class inherits from a Car class
  - c. A Motorcycle class inherits from a Car class
  - d. A Car class inherits from a Motorcycle class
  
4. Which of the following is not an access modifier in Java?
  - a. private
  - b. public
  - c. protected

d. global

5. What is the difference between an abstract class and an interface in Java?

a. An interface can have method implementations, while an abstract class cannot

b. An abstract class can have method implementations, while an interface cannot

c. An abstract class can only have abstract methods, while an interface can have both abstract and non-abstract methods

d. An interface can only have abstract methods, while an abstract class can have both abstract and non-abstract methods

6. What is the term used for creating an instance of a class in Java?

a. Instantiating

b. Constructing

c. Declaring

d. Initializing

7. What is a constructor in Java?

a. A method that is used to create objects

b. A method that is used to destroy objects

c. A method that is used to access private class members

d. A method that is used to override superclass methods

8. What is an interface in Java?

a. A class that defines implementation details

- b. A class that can be instantiated directly
- c. A blueprint for implementing a set of related methods
- d. A mechanism for hiding implementation details

9. Which keyword is used to create an interface in Java?

- a. interface
- b. class
- c. implements
- d. extends

10. What is the difference between composition and inheritance in Java?

- a. Composition allows a class to inherit behavior from another class, while inheritance allows a class to combine behavior from multiple classes.
- b. Inheritance allows a class to inherit behavior from another class, while composition allows a class to combine behavior from multiple classes.
- c. Composition and inheritance are two names for the same concept in Java.
- d. Composition and inheritance are not related concepts in Java.

11. What is method overloading in Java?

- a. Defining two methods in the same class with the same name but different parameters
- b. Defining two methods in the same class with the same name and same parameters
- c. Defining two methods in different classes with the same name and same parameters

d. Defining two methods in different classes with the same name but different parameters

12. What is method overriding in Java?

a. Defining a method in a subclass with the same name and signature as a method in the superclass

b. Defining a method in a subclass with the same name but different signature as a method in the superclass

c. Defining a method in a superclass with the same name and signature as a method in a subclass

d. Defining a method in a superclass with the same name but different signature as a method in a subclass

13. What is the keyword used to access the superclass from a subclass in Java?

a. super

b. this

c. extends

d. implements

14. What is the purpose of the final keyword in Java?

a. To indicate that a variable or method cannot be changed after initialization

b. To indicate that a variable or method can only be accessed within the same class

c. To indicate that a class cannot be extended by other classes

d. To indicate that a method must be implemented by all subclasses

15. What is the purpose of the static keyword in Java?

- a. To indicate that a variable or method is specific to a single object
- b. To indicate that a variable or method is specific to a class and not to any object of the class
- c. To indicate that a variable or method is protected from outside access
- d. To indicate that a variable or method can be changed at runtime

16. Which of the following is an example of encapsulation in Java?

- a. A public variable in a class
- b. A private variable in a class with public getter and setter methods
- c. A private variable in a class with no accessor methods
- d. A public variable in a class with no accessor methods

17. Which of the following is not a valid primitive data type in Java?

- a. int
- b. double
- c. boolean
- d. string

18. Which of the following is not a valid keyword in Java?

- a. try
- b. catch
- c. throw
- d. define

19. Which of the following is a keyword used to define a class in Java?

- a. class
- b. object
- c. type
- d. struct

20. Which of the following is an example of polymorphism in Java?

- a. A Car class has a method named drive() that takes a single parameter
- b. A Car class has two methods named drive() with different parameter types
- c. A Car class has two methods named drive() with the same parameter types but different return types
- d. A Car class has two fields named make and model

21. Which of the following is not a valid way to declare an array in Java?(all options are valid.)

- a. `int[] arr = new int[5];`
- b. `int[] arr = {1, 2, 3};`
- c. `int arr[] = new int[5];`
- d. `int arr[] = {1, 2, 3};`

22. Which of the following is an example of a wrapper class in Java?

- a. int
- b. float
- c. String
- d. Integer

23. Which of the following is not a valid way to create an object in Java?

- a. `Car myCar = new Car();` (valid)
- b. `Car myCar = Car();` (no new keyword used)
- c. `Car myCar = new Car;` (compilation error)
- d. `Car myCar;`

24. Which of the following is not a valid way to define a method in Java?

- a. `public static void myMethod()`
- b. `private void myMethod()`
- c. `static void myMethod()`
- d. `public int myMethod()`

In Java, a method can be defined using the access modifiers (public, private, protected), the non-access modifiers (static, final, abstract), the return type, the method name, and the parameter list. The method signature includes the method name and the parameter list, but not the return type.

25. Which of the following is not a valid way to declare a variable in Java?

- a. `int myVar = 5;`
- b. `String myVar = "Hello";`
- c. `float myVar = 3.14;`
- d. `double myVar = "3.14";`

26. Which of the following is an example of method overloading in Java?

- a. A Car class has two methods named start() and stop()
- b. A Car class has two methods named start() with different parameter types
- c. A Car class has two methods named start() with different return types
- d. A Car class has two fields named make and model

27. Which of the following is an example of method overriding in Java?

- a. A Car class has a method named drive() that takes a single parameter
- b. A Car class has two methods named drive() with different parameter types
- c. A Car class inherits a method named drive() from a Vehicle class and overrides it with a new implementation
- d. A Car class has two fields named make and model

28. Which of the following is an example of a checked exception in Java?

- a. ArithmeticException
- b. NullPointerException
- c. IllegalArgumentException
- d. IOException

IOException is a checked exception in Java, which means that it must be handled in a try-catch block or declared in the method signature using the 'throws' keyword. The other options are all unchecked exceptions.

29. Which of the following is not a valid way to handle an exception in Java?

- a. Using try-catch block
- b. Using throws keyword



c. Using final keyword

d. Using finally block

30. Which of the following is an example of a non-static method in Java?

a. A method that can be called without creating an object of the class

b. A method that can only be called on an object of the class

c. A method that can be called from any class in the program

d. A method that can be called from within the same class only

31. Which of the following is not a valid type of loop in Java?

a. for

b. while

c. until

d. do-while

32. Which of the following is not a valid comparison operator in Java?

a. ==

b. !=

c. <>

d. <=

33. Which of the following is an example of a constructor in Java?

a. A method that is called when an object of the class is created

b. A method that can be called without creating an object of the class

- c. A method that is called automatically when the program starts
- d. A method that is used to destroy an object of the class

34. Which of the following is an example of a final variable in Java?

- a. A variable whose value cannot be changed after it is initialized
- b. A variable that can be accessed from any class in the program
- c. A variable that is automatically initialized with a default value
- d. A variable that is used to store a reference to an object

35. Which of the following is not a valid way to implement an interface in Java?

- a. Implementing the interface using a class
- b. Implementing the interface using an abstract class
- c. Implementing the interface using a concrete class
- d. Implementing the interface using an anonymous inner class

36. Which of the following is an example of an anonymous inner class in Java?

- a. A class that is defined inside another class without a name
- b. A class that is defined outside of any other class
- c. A class that is defined inside another class with a name
- d. A class that is used to create objects of another class

37. Which of the following is not a valid way to define a constant in Java?

- a. Using the final keyword
- b. Using the const keyword

- c. Using the static keyword
- d. Using the final and static keywords together

38. Which of the following is an example of a method call in Java?

- a. `myMethod();`
- b. `myMethod;`
- c. `myMethod[];`
- d. `myMethod{};`

39. Which of the following is not a valid way to initialize a variable in Java?

- a. `int x = 5;`
- b. `int y = new int();`
- c. `int z = x + y;`
- d. `int w = (int)3.14;`

40. Which of the following is an example of a type cast in Java?

- a. `double x = 3.14;`  
`int y = x;`
- b. `int x = 5;`  
`double y = x;`
- c. `String x = "3.14";`  
`double y = (double)x;`
- d. `boolean x = true;`  
`int y = (int)x; ( you can't cast a boolean to an int directly.)`

41. Which of the following option leads to the portability and security of Java?

- a. Bytecode is executed by JVM
- b. The applet makes the Java code secure and portable
- c. Use of exception handling
- d. Dynamic binding between object

42. Which of the following is not a Java features?

- a. Dynamic
- b. Architecture Neutral
- c. Use of pointers
- d. Object-oriented

43. What should be the execution order, if a class has a method, static block, instance block, and constructor, as shown below?

```
public class First_C {  
    public void myMethod()  
    {  
        System.out.println("Method");  
    }  
  
    {  
        System.out.println(" Instance Block");  
    }  
  
    public void First_C()  
    {  
        System.out.println("Constructor ");  
    }  
}
```

```

static {
    System.out.println("static block");
}
public static void main(String[] args) {
    First_C c = new First_C();
    c.First_C();
    c.myMethod();
}
}

```

- a. Instance block, method, static block, and constructor
- b. Method, constructor, instance block, and static block
- c. Static block, method, instance block, and constructor
- d. Static block, instance block, constructor, and method

44. What will be the output of the following program?

```

public class MyFirst {
    public static void main(String[] args) {
        MyFirst obj = new MyFirst(n);
    }
    static int a = 10;
    static int n;
    int b = 5;
    int c;
    public MyFirst(int m) {
        System.out.println(a + ", " + b + ", " + c + ", " + n + ", " + m);
    }
    // Instance Block
    {
        b = 30;
        n = 20;
    }
}

```

```
// Static Block
```

```
static
```

```
{  
    a = 60;  
}  
}
```

- a. 10, 5, 0, 20, 0
- b. 10, 30, 20
- c. 60, 5, 0, 20
- d. 60, 30, 0, 20, 0

45. Evaluate the following Java expression, if x=3, y=5, and z=10:

**++z + y - y + z + x++**

- a. 24
- b. 23
- c. 20
- d. 25

46. What will be the output of the following program?

```
public class Test {
```

```
    public static void main(String[] args) {
```

```
        int count = 1;
```

```
        while (count <= 15) {
```

```
            System.out.println(count % 2 == 1 ? "***" : "+++++");
```

```

        ++count;

    }    // end while

}    // end main

}

```

- a. 15 times \*\*\*
- b. 15 times +++++
- c. 8 times \*\*\* and 7 times +++++
- d. Both will print only once

47. What is the output of the following program?

```

public class MyClass {

    private int num;

    public MyClass(int num) {

        this.num = num;

    }

    public int getNum() {

        return this.num;

    }

    public void setNum(int num) {

        this.num = num;

    }
}

```

```
}
```

```
public class Main {  
  
    public static void main(String[] args) {  
  
        MyClass obj1 = new MyClass(10);  
  
        MyClass obj2 = new MyClass(20);  
  
        obj1.setNum(obj2.getNum());  
  
        System.out.println(obj1.getNum());  
  
    }  
}
```

a. 10

b. 20

c. 30

d. Compilation error

48. What will be the output?

```
class A {  
  
    int num = 10;  
  
}
```



```
class B extends A {
```

```
    int num = 20;
```

```
}
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        A a = new B();
```

```
        System.out.println(a.num);
```

```
    }
```

```
}
```

a. 10

b. 20

c. Compilation error

d. Runtime error

49. what will be the output?

```
class A {
```

```
    static void print() {
```

```
        System.out.println("A");
```

```
    }
```

```
}
```

```
class B extends A {  
    static void print() {  
        System.out.println("B");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        A a = new B();  
        a.print();  
    }  
}
```

a. A

b. B

c. Compilation error

d. Runtime error

50. what will be the output?

```
class Animal {  
  
    void makeSound() {  
  
        System.out.println("Animal is making a sound.");  
  
    }  
  
}  
  
class Cat extends Animal {  
  
    void makeSound() {  
  
        System.out.println("Meow!");  
  
    } }  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        Animal myAnimal = new Cat();  
  
        myAnimal.makeSound();  
  
    } }
```

a. "Animal is making a sound."

b. "Meow!"

c. Compilation error

d. Runtime error

51. what will be the output?

```
class A {  
  
    int i;  
  
    A() {  
  
        i = 10;  
  
    }  
}  
  
class B extends A {  
  
    int j;  
  
    B() {  
  
        j = 20;  
  
    }  
}  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        A a = new B();  
  
        System.out.println(a.i + " " + ((B)a).j);  
  
    }  
}
```

a. "10 20"

b. Compilation error

c. Runtime error

d. "20 10"

52. what will be the output?

```
class A {  
    public void print() {  
        System.out.println("A");  
    }  
}  
  
class B extends A {  
    public void print() {  
        System.out.println("B");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        A a = new B();  
        B b = (B) a;
```

```
        b.print();  
    }  
}
```

a. "A"

b. "B"

c. Compilation error

d. Runtime error

53. What will be the output?

```
class A {  
    int i = 10;  
    A() {  
        System.out.println("A constructor");  
    }  
}
```

```
class B extends A {  
    int j = 20;  
    B() {  
        System.out.println("B constructor");  
    }  
}
```

```
}  
}
```

```
public class Main {  
  
    public static void main(String[] args) {  
  
        B b = new B();  
  
        System.out.println(b.i + " " + b.j);  
  
    }  
}
```

a. "A constructor", "B constructor", "10 20"

b. "B constructor", "A constructor", "10 20"

c. "A constructor", "B constructor", "0 20"

d. "B constructor", "A constructor", "0 20"

