**✅ 1. Which of the following is not a feature of OOP in Java?**

A. Encapsulation  
B. Inheritance  
C. Compilation  
D. Polymorphism

**Answer:** C

**✅ 2. Which of the following best defines Encapsulation?**

A. Hiding internal details and showing only functionality  
B. Binding code and data together into a single unit  
C. Using methods with the same name  
D. Inheriting properties from parent class

**Answer:** B

**✅ 3. What is the main purpose of inheritance in Java?**

A. To overload methods  
B. To implement encapsulation  
C. To achieve code reusability  
D. To hide data from users

**Answer:** C

**✅ 4. Which keyword is used for inheritance in Java?**

A. implements  
B. extends  
C. inherit  
D. super

**Answer:** B

**✅ 5. Which of these concepts allows the use of the same method name with different parameters?**

A. Inheritance  
B. Abstraction  
C. Method Overloading  
D. Encapsulation

**Answer:** c

**✅ 6. What is method overriding in Java?**

A. Writing two methods with the same name in the same class  
B. Changing the implementation of a method in subclass  
C. Using private methods in subclass  
D. Using final methods in subclass

**Answer:** b

**✅ 7. Which of the following is true about abstract classes in Java?**

A. Can be instantiated  
B. Cannot have any methods  
C. Must contain only abstract methods  
D. Can have both abstract and concrete methods

**Answer:** D

**✅ 8. Which access modifier makes members visible only within the same package?**

A. public  
B. private  
C. protected  
D. default (no modifier)

**Answer:** d

**✅ 9. Which of the following is NOT a type of polymorphism in Java?**

A. Compile-time polymorphism  
B. Run-time polymorphism  
C. Static polymorphism  
D. Constructor polymorphism

**Answer:** D

**✅ 10. Which of these is a correct example of abstraction?**

A. Using getter and setter methods  
B. Using interface or abstract class  
C. Creating objects  
D. Using loops

**Answer:** B

**✅ 11. What will be the output of the following code?**

class Parent {

void show() {

System.out.println("Parent");

}

}

class Child extends Parent {

void show() {

System.out.println("Child");

}

}

public class Test {

public static void main(String[] args) {

Parent obj = new Child();

obj.show();

}

}

A. Parent  
B. Child  
C. Compile-time error  
D. Runtime error

**Answer:** B. Child

**✅ 12. What is the output?**

abstract class Animal {

abstract void sound();

}

class Dog extends Animal {

void sound() {

System.out.println("Bark");

}

}

What happens if you write:

java

CopyEdit

Animal a = new Animal();

A. Bark  
B. Compile-time error  
C. Runtime error  
D. None of the above

**Answer:** b

**✅ 13. Which of the following can be used to achieve multiple inheritance in Java?**

A. Class  
B. Abstract class  
C. Interface  
D. Object

**Answer:** C

**✅ 14. Which concept is violated if a subclass breaks encapsulation?**

A. Inheritance  
B. Abstraction  
C. Polymorphism  
D. Encapsulation

**Answer:D**

**✅ 15. Which is true about constructors in Java?**

A. Can be overridden  
B. Can be static  
C. Can be inherited  
D. Cannot be inherited

**Answer:** d

**✅ 16. What does the super keyword do?**

A. Calls the child class method  
B. Refers to the current object  
C. Calls the parent class constructor or method  
D. None of the above

**Answer:** C

**✅ 17. Which of the following allows achieving abstraction and multiple inheritance?**

A. Interface  
B. Static class  
C. Final class  
D. Constructor

**Answer:A**

**✅ 18. If a method is final, which statement is true?**

A. It can be overridden  
B. It can be hidden  
C. It cannot be overridden  
D. It must be static

**Answer:** c

**✅ 19. What will happen if the base and derived class both have a static method with same signature?**

A. Runtime polymorphism  
B. Compile-time error  
C. Method overriding  
D. Method hiding

**Answer:d**

**✅ 20. Which of the following is used to restrict inheritance in Java?**

A. private keyword  
B. final keyword  
C. static keyword  
D. protected keyword

**Answer:** b