Top of Form

11. Which of the following statements about constructors in inheritance is true?

 Constructors are inherited from the superclass to the subclass  
 Constructors are inherited, but they can be overridden  
 The subclass constructor can call the superclass constructor using super()  
 Constructors of the superclass are not called when creating an instance of the subclass

Bottom of Form

Top of Form

12. In Java, which keyword is used to prevent a class from being inherited?

 static  
 final  
 abstract  
 private

Bottom of Form

Top of Form

13. What is the output of the following program?  
class Base {  
public void foo() { System.out.println("Base"); }  
}  
class Derived extends Base {  
private void foo() { System.out.println("Derived"); }  
}  
public class Main {  
public static void main(String args[]) {  
Base b = new Derived();  
b.foo();  
}  
}

 Compiler Error  
 Derived  
 Base  
 Runtime Error

Bottom of Form

Top of Form

14. What will be the output of the following program?  
class One{  
public One(){  
System.out.print("One,");  
}  
}  
class Two extends One{  
public Two(){  
System.out.print("Two,");  
}  
}  
class Three extends Two{  
public Three(){  
System.out.print("Three");  
}  
}  
public class Test{  
  
public static void main(String[] args){  
Three three = new Three();  
}  
}

 Run-time error  
 Three  
 One  
 One,Two,Three

Bottom of Form

Top of Form

15. What is Inheritance in Java programming?

 It's a technique to create objects in Java.  
 It's a Java-specific term for importing packages.  
 It's a process of creating a new class using the main() method.  
 It's a process where one class acquires the properties (fields) and behaviors (methods) of another class.

Bottom of Form

Top of Form

16. // filename Main.java  
class Grandparent {  
public void Print() {  
System.out.println("Grandparent\'s Print()");  
}  
}  
class Parent extends Grandparent {  
public void Print() {  
System.out.println("Parent\'s Print()");  
}  
}  
class Child extends Parent {  
public void Print() {  
super.super.Print();  
System.out.println("Child\'s Print()");  
}  
}  
public class Main {  
public static void main(String[] args) {  
Child c = new Child();  
c.Print();  
}  
}

 Compiler Error in super.super.Print()  
 Runtime Error  
 Grandparent\'s Print()Parent\'s Print()Child\'s Print()  
 Compiler Error

Bottom of Form

Top of Form

17. What will be the output of the following program?  
class ClassOne  
{  
int variableOne = 10;  
}  
class ClassTwo extends ClassOne  
{  
int variableOne = 20;  
}  
public class Main  
{  
public static void main(String[] args)  
{  
ClassOne objOne = new ClassTwo();  
  
System.out.println(objOne.variableOne);  
}  
}

 Program will not run.  
 20  
 10  
 30

Bottom of Form

Top of Form

18. What is a multilevel inheritance in Java?

 Two or more classes extend the same class  
 A class extends two or more classes  
 A class extends another class which also extends another class  
 All of the above

Bottom of Form

Top of Form

19. Can a final class be inherited in Java?

 No  
 Yes, but only by classes in the same package  
 Yes, but only by other final classes  
 Yes, but only by abstract classes

Bottom of Form

Top of Form

20. What is the output of the following program?  
class Base {  
public void Print() {  
System.out.println("Base");  
}  
}  
class Derived extends Base {  
public void Print() {  
System.out.println("Derived");  
}  
}  
  
class Main{  
public static void DoPrint( Base o ) {  
o.Print();  
}  
public static void main(String[] args) {  
Base x = new Base();  
Base y = new Derived();  
Derived z = new Derived();  
DoPrint(x);  
DoPrint(y);  
DoPrint(z);  
}  
}

 Base, Base, Derived  
 Base, Derived, Derived  
 Base, Derived, Base  
 Compiler Error

Top of Form

31. class Base {  
public void show() {  
System.out.println("Base::show() called");  
}  
}  
class Derived extends Base {  
public void show() {  
System.out.println("Derived::show() called");  
}  
}  
public class Main {  
public static void main(String[] args) {  
Base b = new Derived();;  
b.show();  
}  
}

 Derived::show() called  
 Compiler Error  
 Base::show() called  
 Runtime Error

Bottom of Form

Top of Form

32. What is a multilevel inheritance in Java?

 A class extends two or more classes  
 A class extends another class which also extends another class  
 All of the above  
 Two or more classes extend the same class

Bottom of Form

Top of Form

33. What is the parent class of all classes in Java?

 System  
 Object  
 Class  
 String

Bottom of Form

Top of Form

34. Which keyword is used to declare an abstract method in a class?

 static  
 void  
 final  
 abstract

Bottom of Form

Top of Form

35. What will be the output of the following Java program?  
class A  
{  
int i;  
}  
class B extends A  
{  
int j;  
void display()  
{  
super.i = j + 1;  
System.out.println(j + " " + i);  
}  
}  
class inheritance  
{  
public static void main(String args[])  
{  
B obj = new B();  
obj.i=1;  
obj.j=2;  
obj.display();  
}  
}

 2 3  
 2 2  
 3 2  
 3 3

Bottom of Form

Top of Form

36. Which of these keywords cannot be used for a class which has been declared final?

 extends  
 none of the mentioned  
 abstract and extends  
 abstract

Bottom of Form

Top of Form

37. What would be the result if a class extends two interfaces and both have a method with same name and signature? Lets assume that the class is not implementing that method.

 First called method is executed successfully  
 Code runs successfully  
 Runtime error  
 Compile time error

Bottom of Form

Top of Form

38. Which type of inheritance is NOT supported in Java?

 Hierarchical inheritance  
 Multiple inheritance (using classes)  
 Multilevel inheritance  
 Single inheritance

Bottom of Form

Top of Form

39. What is the keyword used to indicate that a class is inheriting from another class in Java?

 inherits  
 implements  
 derives  
 extends

Bottom of Form

Top of Form

40. What is the primary benefit of using polymorphism in Java?

 Reusing parent class logic  
 The ability to define methods with the same name in different classes  
 Code readability  
 Memory management

Bottom of Form

Top of Form

21. Which of the following is true about method overriding in Java?

 The method name must be different in the subclass  
 Only private methods can be overridden  
 The return type must be the same or a subtype in the overridden method  
 Only abstract methods can be overridden

Bottom of Form

Top of Form

22. What is the primary benefit of using inheritance in Java?

 Code reusability and method overriding  
 Preventing method calls  
 Preventing class instantiation  
 Increased execution speed

Bottom of Form

Top of Form

23. When two String objects in Java need to be compared for equality, which method should be used?

 ==  
 compareTo()  
 hashCode()  
 equals()

Bottom of Form

Top of Form

24. What pattern is typically used when a class delegates the creation of objects to subclasses?

 Singleton pattern  
 Observer pattern  
 Factory method pattern  
 Adapter pattern

Bottom of Form

Top of Form

25. All classes in Java are inherited from which class?

 java.lang.class  
 java.class.object  
 java.class.inherited  
 java.lang.Object

Bottom of Form

Top of Form

26. Which of the following is true about constructors in a subclass?

 The subclass constructor can only call a parameterized superclass constructor  
 A subclass must define its own constructor  
 A subclass constructor cannot call the superclass constructor  
 The superclass constructor is called automatically if no constructor is defined in the subclass

Bottom of Form

Top of Form

27. What is not type of inheritance?

 Hierarchical inheritance  
 Multiple inheritance  
 Single inheritance  
 Double inheritance

Bottom of Form

Top of Form

28. What will be the output of the following program?  
class ClassAlpha  
{  
{  
System.out.println("Alpha");  
}  
}  
class ClassBeta extends ClassAlpha  
{  
{  
System.out.println("Beta");  
}  
}  
class ClassGamma extends ClassBeta  
{  
{  
System.out.println("Gamma");  
}  
}  
public class Main  
{  
public static void main(String[] args)  
{  
ClassGamma gamma = new ClassGamma();  
}  
}

 Gamma  
 Beta  
 Alpha Beta Gamma  
 Alpha

Bottom of Form

Top of Form

29. What method is used to check an object's type before casting in Java?

 getClass()  
 instanceof  
 checkType()  
 getType()

Bottom of Form

Top of Form

30. Can interfaces be used to achieve multiple inheritance in Java?

 Yes  
 No  
 Neither  
 None

Top of Form

21. Which of the following is true about method overriding in Java?

 The method name must be different in the subclass  
 Only private methods can be overridden  
 The return type must be the same or a subtype in the overridden method  
 Only abstract methods can be overridden

Bottom of Form

Top of Form

22. What is the primary benefit of using inheritance in Java?

 Code reusability and method overriding  
 Preventing method calls  
 Preventing class instantiation  
 Increased execution speed

Bottom of Form

Top of Form

23. When two String objects in Java need to be compared for equality, which method should be used?

 ==  
 compareTo()  
 hashCode()  
 equals()

Bottom of Form

Top of Form

24. What pattern is typically used when a class delegates the creation of objects to subclasses?

 Singleton pattern  
 Observer pattern  
 Factory method pattern  
 Adapter pattern

Bottom of Form

Top of Form

25. All classes in Java are inherited from which class?

 java.lang.class  
 java.class.object  
 java.class.inherited  
 java.lang.Object

Bottom of Form

Top of Form

26. Which of the following is true about constructors in a subclass?

 The subclass constructor can only call a parameterized superclass constructor  
 A subclass must define its own constructor  
 A subclass constructor cannot call the superclass constructor  
 The superclass constructor is called automatically if no constructor is defined in the subclass

Bottom of Form

Top of Form

27. What is not type of inheritance?

 Hierarchical inheritance  
 Multiple inheritance  
 Single inheritance  
 Double inheritance

Bottom of Form

Top of Form

28. What will be the output of the following program?  
class ClassAlpha  
{  
{  
System.out.println("Alpha");  
}  
}  
class ClassBeta extends ClassAlpha  
{  
{  
System.out.println("Beta");  
}  
}  
class ClassGamma extends ClassBeta  
{  
{  
System.out.println("Gamma");  
}  
}  
public class Main  
{  
public static void main(String[] args)  
{  
ClassGamma gamma = new ClassGamma();  
}  
}

 Gamma  
 Beta  
 Alpha Beta Gamma  
 Alpha

Bottom of Form

Top of Form

29. What method is used to check an object's type before casting in Java?

 getClass()  
 instanceof  
 checkType()  
 getType()

Bottom of Form

Top of Form

30. Can interfaces be used to achieve multiple inheritance in Java?

 Yes  
 No  
 Neither  
 None

Bottom of Form

Bottom of Form

Bottom of Form