

# Java: Interface

Quiz

# Question 1

```
interface X{static void doIt();}  
  
public class Test1 implements X{  
  
    public static void doIt(){ System.out.println("OK");}  
  
    public static void main(String st[]){  
  
        Test1.doIt();  
  
    }  
}
```

Which of the following is true about the code above?

- A. The code does not compile because of a problem in interface.
- B. The code does not compile of a problem in Test1 class
- C. The code compiles and prints “OK” at execution time.
- D. The code compiles but throws a runtime error.

# Question 1

```
interface X{static void doIt();}  
  
public class Test1 implements X{  
  
    public static void doIt(){ System.out.println("OK");}  
  
    public static void main(String st[]){  
  
        Test1.doIt();  
  
    }  
}
```

Which of the following is true about the code above?

- ☒ A. The code does not compile because of a problem in interface.
- ☐ B. The code does not compile of a problem in Test1 class
- ☐ C. The code compiles and prints “OK” at execution time.
- ☐ D. The code compiles but throws a runtime error.

## Question 2

```
interface X{
    int k=90;
    int doIt();}
public class Test1 implements X{
    int l=doIt();
    // insert method declaration for doIt()
    { return X.k;}
    public static void main(String st[]){
        Test1 t=new Test1();
        t.doIt();
        System.out.print(t.l);}}
```

Which of the following declaration(s) for doIt() in Test1 will compile the code?

- A. `int doIt()`
- B. `public int doIt()`
- C. `public static int doIt()`
- D. Code has syntax error and will not compile for any declaration of `doIt()`

## Question 2

```
interface X{
    int k=90;
    int doIt();}
public class Test1 implements X{
    int l=doIt();
    // insert method declaration for doIt()
    { return X.k;}
    public static void main(String st[]){
        Test1 t=new Test1();
        t.doIt();
        System.out.print(t.l);}}
```

Which of the following declaration(s) for doIt() in Test1 will compile the code?

- A. int doIt()
- ☒ B. public int doIt()
- C. public static int doIt()
- D. Code has syntax error and will not compile for any declaration of doIt()

# Question 3

```
1. interface X{ int k=90;}  
2. public class Test1 implements X{  
3. static{System.out.print(k) ;}  
4. public static void main(String st[]){  
5. }}
```

What will happen on compilation or execution of code?

- A. Compilation error at line 1
- B. Compilation error at line 3
- C. Code prints 90
- D. Code prints 0

# Question 3

```
1. interface X{ int k=90;}  
2. public class Test1 implements X{  
3. static{System.out.print(k) ;}  
4. public static void main(String st[]){  
5. }}
```

What will happen on compilation or execution of code?

- A. Compilation error at line 1
- B. Compilation error at line 3
- ☒ C. Code prints 90
- D. Code prints 0

# Question 4

Which of the following is true about interfaces?

- A. Interface is an abstract class
- B. Interface is always public
- C. Interface can have a package declaration
- D. Interface automatically inherits from Object class



# Question 4

Which of the following is true about interfaces?

- A. Interface is an abstract class
- B. Interface is always public
- ☒ C. Interface can have a package declaration
- D. Interface automatically inherits from Object class

# Question 5

```
public class Flower implements Cloneable{  
    // insert method declaration here  
    { return(Flower)  super.clone() ;}  
}
```

Which of the following is/are NOT right declaration for clone method?

- A. `public Object clone() throws CloneNotSupportedException`
- B. `protected Object clone() throws CloneNotSupportedException`
- C. `Object clone() throws CloneNotSupportedException`
- D. `public Flower clone() throws CloneNotSupportedException`

# Question 5

```
public class Flower implements Cloneable{  
    // insert method declaration here  
    { return(Flower) super.clone() ;}  
}
```

Which of the following is/are NOT right declaration for clone method?

- A. `public Object clone() throws CloneNotSupportedException`
- B. `protected Object clone() throws CloneNotSupportedException`
- ☒ C. `Object clone() throws CloneNotSupportedException`
- D. `public Flower clone() throws CloneNotSupportedException`

## Question 6

```
interface A {  
    int a = 1;                // 1  
    public int b = 2;         // 2  
    public static int c = 3;  // 3  
}
```

Which field declaration results in a compile-time error?

- A. 1
- B. 2
- C. 3
- D. None of the above

## Question 6

```
interface A {  
    int a = 1;                // 1  
    public int b = 2;         // 2  
    public static int c = 3;  // 3  
}
```

Which field declaration results in a compile-time error?

A. 1

B. 2

C. 3

☒ D. None of the above

# Question 7

```
interface I1 {String name = "I1"; }  
interface I2 {String name = "I2"; }  
class X implements I1, I2 {           //line 1  
public static void main(String[] args) {  
System.out.print(name);               //line 2  
}}
```

What is the result of attempting to compile and run the program?

- A. I1
- B. I2
- C. Compilation error at line 1
- D. Compilation error at line 2

# Question 7

```
interface I1 {String name = "I1"; }  
interface I2 {String name = "I2"; }  
class X implements I1, I2 {           //line 1  
public static void main(String[] args) {  
System.out.print(name);               //line 2  
}}
```

What is the result of attempting to compile and run the program?

- A. I1
- B. I2
- C. Compilation error at line 1
- ☒ D. Compilation error at line 2

## Question 8

```
interface I {final void f();}  
  
abstract class X implements I{  
  
public final void f(){}  
  
public static void main(String[] args) {  
new X().f();}}
```

Spot the all the problems in the code .

- A. Interface I cannot have final method
- B. Class X cannot be declared as abstract since it implements interface
- C. Method f() cannot be overridden in class X
- D. Instance of X cannot be created



## Question 8

```
interface I {final void f();}  
  
abstract class X implements I{  
  
public final void f(){}  
  
public static void main(String[] args) {  
  
new X().f();}}
```

Spot the all the problems in the code .

- ☒ A. Interface I cannot have final method
- ☐ B. Class X cannot be declared as abstract since it implements interface
- ☒ C. Method f() cannot be overridden in class X
- ☒ D. Instance of X cannot be created

# Question 9

```
1. public interface X {  
2.     void f() ;}  
3. abstract class A implements X{  
4.     public final void f(){}  
5.     public static void main(String[] args) {  
6.     }}
```

Which lines will cause compilation error

- A. Line 1
- B. Line 3
- C. Line 4
- D. No compilation error

# Question 9

```
1. public interface X {  
2.     void f() ;}  
3. abstract class A implements X{  
4.     public final void f(){}  
5.     public static void main(String[] args) {  
6.     }}
```

Which lines will cause compilation error

- A. Line 1
- B. Line 3
- C. Line 4
- ☒ D. No compilation error

## Question 10

```
interface Y {}  
public class X{  
    static String s;  
    public static void main(String[] args) {  
        X a1= new X();  
        System.out.print(a1 instanceof X); ///Line 1  
        System.out.print(a1 instanceof Y); //Line 2  
        System.out.print(s instanceof Y); //Line 3  
        System.out.print(null instanceof Y); //Line 4  
    }  
}
```

Which of the following is true about the code?

- A. Line 1 prints true
- B. Line 2 prints false
- C. Line 3 gives compilation error
- D. Line 4 prints false

## Question 10

```
interface Y {}  
public class X{  
    static String s;  
    public static void main(String[] args) {  
        X a1= new X();  
        System.out.print(a1 instanceof X); ///Line 1  
        System.out.print(a1 instanceof Y); //Line 2  
        System.out.print(s instanceof Y); //Line 3  
        System.out.print(null instanceof Y); //Line 4  
    }  
}
```

Which of the following is true about the code?

- ☒ A. Line 1 prints true
- ☒ B. Line 2 prints false
- ☒ C. Line 3 gives compilation error
- ☒ D. Line 4 prints false

# Question 11

```
interface A {int i = 1; int m1();}  
  
interface B extends A {int i = 10; int m1();}  
  
class X implements B {  
  
    public int m1() {return i;}  
  
    public static void main(String[] args) {  
  
        System.out.print(new X().m1());  
  
    }  
}
```

What happens on compilation and execution of the code?

- A. Prints 1
- B. Prints 10
- C. Compilation error
- D. Runtime error

# Question 11

```
interface A {int i = 1; int m1();}  
  
interface B extends A {int i = 10; int m1();}  
  
class X implements B {  
  
    public int m1() {return i;}  
  
    public static void main(String[] args) {  
  
        System.out.print(new X().m1());  
  
    }  
}
```

What happens on compilation and execution of the code?

- A. Prints 1
- ☒ B. Prints 10
- C. Compilation error
- D. Runtime error

# Question 12

```
interface I1 { }  
interface I2 {}  
class C1 implements I1{}  
class C2 implements I2 {}  
class C3 implements I1,I2 {}
```

Assume the following declarations.

```
C1 obj1;      C2 obj2;      C3 obj3;
```

Which of the following are valid assignments

- A. `obj2 = obj1;`
- B. `I1 a = obj2;`
- C. `I2 a = obj2;`
- D. `I2 c = obj1;`



# Question 12

```
interface I1 { }  
interface I2 {}  
class C1 implements I1{}  
class C2 implements I2 {}  
class C3 implements I1,I2 {}
```

Assume the following declarations.

```
C1 obj1;      C2 obj2;      C3 obj3;
```

Which of the following are valid assignments

A. `obj2 = obj1;`

B. `I1 a = obj2;`

☒ C. `I2 a = obj2;`

D. `I2 c = obj1;`

# Question 13

```
interface A {}  
class Y{}  
class X implements A {  
public static void main(String str[]){  
X a = new X();  
A a0=(A)a;//Line 1  
Y a1=(Y)a;//Line 2  
Cloneable a2=(Cloneable)a;//Line 3  
}}
```

Which statement causes compilation error?

- A. Line 1
- B. Line 2
- C. Line 3
- D. None

# Question 13

```
interface A {}  
class Y{}  
class X implements A {  
public static void main(String str[]){  
X a = new X();  
A a0=(A)a;//Line 1  
Y a1=(Y)a;//Line 2  
Cloneable a2=(Cloneable)a;//Line 3  
}}
```

Which statement causes compilation error?

- A. Line 1
- ☒ B. Line 2
- C. Line 3
- D. None

# Question 14

Which of the following are marker interface?

- A. `Serializable`
- B. `Comparable`
- C. `Cloneable`
- D. `Comparator`

# Question 14

Which of the following are marker interface?

- ☒ A. Serializable
- ☐ B. Comparable
- ☒ C. Cloneable
- ☐ D. Comparator

## Question 15

```
import java.util.*;

public class A {
    static int[] b = { 10,30,5,20 };
    public static void main(String a[]){
        System.out.println(Arrays.binarySearch(b, 5));
    }
}
```

What will the code display?

- A. 0
- B. 1
- C. 2
- D. Result is undefined

## Question 15

```
import java.util.*;  
  
public class A {  
    static int[] b = { 10,30,5,20 };  
    public static void main(String a[]){  
        System.out.println(Arrays.binarySearch(b, 5));  
    }  
}
```

What will the code display?

- A. 0
- B. 1
- C. 2
- ☒ D. Result is undefined

## Question 16

```
final class Z{}  
public class X{  
    public static void main(String[] args) {  
        Z z= new Z();  
        System.out.print(z instanceof Object); //Line 1  
        System.out.print(z instanceof Comparable);  
        //Line 2  
    }  
}
```

Which of the following statements are true?

- A. Code will display true for Line 1 and false for Line 2
- B. Code will not compile for Line 2 because Z is final class
- C. Code will not compile for Line 1 because Z is final class
- D. Code will not compile for Line 2 because Comparable is not defined



## Question 16

```
final class Z{}  
public class X{  
    public static void main(String[] args) {  
        Z z= new Z();  
        System.out.print(z instanceof Object); //Line 1  
        System.out.print(z instanceof Comparable);  
        //Line 2  
    }  
}
```

Which of the following statements are true?

- A. Code will display true for Line 1 and false for Line 2
- ☒ B. Code will not compile for Line 2 because Z is final class
- C. Code will not compile for Line 1 because Z is final class
- D. Code will not compile for Line 2 because Comparable is not defined