

**Object Oriented Programming Tutorial – Interface**

1. Interface looks like a class but it is not a class. An interface can have me\_\_\_\_\_ and va\_\_\_\_\_ just like the class but the me\_\_\_\_\_ declared in interface are by default abs\_\_\_\_\_ (i.e. only method signature, no body). Also, the va\_\_\_\_\_ declared in an interface are p\_\_\_\_\_, s\_\_\_\_\_ and f\_\_\_\_\_ by default.

Answer:

Interface looks like a

1. State if the each of the following statement is true or false.
2. True/false: Methods in interfaces do not have body.
3. True/false: The class that implements interface must implement all the methods of that interface.
4. True/false: Java allows a class to extend more than one class.
5. True/false: Java allows a class to implement more than one interface.
6. True/false: An interface can implement another interface.
7. True/false: An interface can extend another interface.
8. True/false: An interface can be instantiated.
9. True/false: The object of an interface cannot be created.
10. Write the output of the following program.

interface MyInterface

{

public void method1();

public void method2();

}

public class Demo implements MyInterface

{

public void method1()

{

System.out.println("implementation of method1");

}

public void method2()

{

System.out.println("implementation of method2");

}

public static void main(String arg[])

{

MyInterface obj = new Demo();

obj.method1();

}

}

Answer:

1. Differentiate between abstract class and interface.

|  |  |  |
| --- | --- | --- |
|  | **Abstract Class** | **Interface** |
| 1 | Abstract class can have abstract and non-abstract methods | Interface can have only abstract methods |
| 2 | Abstract class can provide the implementation of interface | Interface can’t provide the implementation of abstract class |
| 3 | The abstract keyword is used to declare abstract class | The interface keyword used to declare interface |
| 4 | An abstract class can extend another Java class and implement multiple Java interfaces | An interface can extend another Java interface only |
| 5 | A java abstract class can have class members like private and protected | Members of a Java interface are public by default. |

1. Explain why the following declaration of interface will throw a compilation error.

interface Try

{

int x; //Compile-time error

}

Answer:

The value of the variable x is not initialized at the time of declaration. By default x is public, final and static therefore it must have a value when declared

1. Identify the error in the following program.

interface Try

{

int x = 10;

}

public class Sample implements Try

{

public static void main(String args[])

{

x = 20;

}

}

Answer:

The variable x is public, static and final by default. As final variables cannot be re-initialized, the x = 10 statement in the main method will throw a compilation error

1. Write the output of the following program.

interface A

{

int x = 10;

}

interface B

{

int x = 100;

}

public class Hello implements A, B

{

public static void main(String args[])

{

System.out.println(A.x);

System.out.println(B.x);

}

}

Answer:

10

100

1. The two types of binding in Java are static binding and dynamic binding.

Answer:

1. \_\_\_\_\_ binding is determined during runtime.

Answer:

1. Compare dynamic polymorphism to static polymorphism in terms of the binding types.

Answer:

Dynamic polymorphism is a process in which a call to an overridden method is resolved through dynamic binding. Static polymorphism is a process in which a call to an overload method is resolved through static binding.

**END**