-------------------------------2.16

61. What is an interface in Java?

An interface in Java is **a blueprint of a class**. It has static constants and abstract methods. The interface in Java is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not method body.

62. Is it allowed to mark an interface method as static?

No, we cannot declare interface methods as static because static methods cannot be overridden.

63. Why an Interface cannot be marked as final in Java?

Interface only have abstract methods. methods. interface method cannot be final. **cannot be declared final**.

64. What is a marker interface?

A marker interface is **an interface that doesn't have any methods or constants inside it**. It provides run-time type information about objects, so the compiler and JVM have additional information about the object. A marker interface is also called a tagging interface.

65. What can we use instead of Marker interface?

Instead of marker interface, Java 5 provides the **annotations** to achieve the same results. It allows flexible metadata capability. Therefore, by applying annotations to any class, we can perform specific action.

66. How Annotations are better than Marker Interfaces?

It seems annotation is a better choice than the marker interface as the same effect can be achieved by the annotations. **It can mark variables, methods, and/or classes**. It can mark any class specifically, or via inheritance. A marker interface will mark all subclasses of the marked class.

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

An interface cannot have state, whereas the abstract class can have state with instance variables.

68. Does Java allow us to use private and protected modifiers for variables in interfaces?

No, it is not possible to define private and protected modifiers for the members in interfaces in Java. As we know that, the members defined in interfaces are implicitly public or in other words, we can say the member defined in an interface is by default public.

69. How can we cast to an object reference to an interface reference?

If you implement an interface and provide body to its methods from a class. You can **hold object of the that class using the reference variable of the interface** i.e. cast an object reference to an interface reference.

70. How can you change the value of a final variable in Java?

Final variable in Java cannot be changed. Once if we have assigned the final variable, it cannot be changed it is fixed. but if you have declared a blank final variable then you can assign value to it only in constructor. also using hierarchical instances can come handy.

71. Can a class be marked final in Java?

You can declare some or all a class's methods final. You use the final keyword in a method declaration to indicate that the method cannot be overridden by subclasses. The Object class does this—several its methods are final.

72. How can we create a final method in Java?

You can declare some or all a class's methods final. You **use the final keyword in a method declaration to indicate that the method cannot be overridden by subclasses**. The Object class does this—several its methods are final.

73. How can we prohibit inheritance in Java?

To prevent inheritance, **use the keyword "final" when creating the class**. The designers of the String class realized that it was not a candidate for inheritance and have prevented it from being extended.

74. Why Integer class in final in Java?

In Java, we can use final keyword with variables, methods, and classes. When the final keyword is used with a variable of primitive data types such as int, float, etc), **the value of the variable cannot be changed**

75. What is a blank final variable in Java?

A blank final variable in Java is **a final variable that is not initialized during declaration**.