

1.What is method overloading in Java & explain with an example.?

Process of defining method with same name and different signature is called as method overloading. Methods which take part in method overloading are called as overloaded method.

```
private static int sum (int a,int b){  
    return a+b;  
}
```

```
private static float sum (int a,float b){  
    return a+b;  
}
```

2.How does Java determine which overloaded method to call?

Java determines which overloaded method to call based on the most specific matching parameters.If there's an exact match, Java selects that method. If not, it looks for the next closest match based on parameter types, preferring widening conversions over narrowing conversions.

3.What does the static keyword mean in Java? Explain the difference between static and non-static methods.

Static: Static members have a global scope and can be accessed from anywhere within the program, even without creating an instance of the class.

Non-Static: Non-static members have a local scope and can be accessed only through an instance of the class. They are not accessible without creating an object.

4.Can static methods be overloaded and overridden in Java?How are static variables shared across multiple instances of a class?

In Java, static methods can be overloaded but not overridden. They can have different parameters while having the same name in the same class or subclass. They cannot be overridden because they act on the class itself, not an object.To store information that is shared across instances of a class, use a static variable. All instances of the same class share a single copy of the static variable.

5.What is the role of the static keyword in the context of memory management?

The static keyword in Java is used to share the same variable or method of a given class. And it will create once per class.

6.What is the significance of the final keyword in Java?

The final keyword is called a "modifier".The final keyword is useful when you want a variable to always store the same value.

7.Can a final method be overridden in a subclass?How does the final keyword affect variables, methods, and classes in Java?

The Methods that are declared as final cannot be Overridden or hidden.

8.What does the this keyword represent in Java?How is the this keyword used in constructors and methods?

The this keyword in Java refers to the current instance of the class. It is primarily used within constructors and methods to refer to the current object. In constructors, this is used to differentiate between instance variables and parameters with the same name. In methods, this is used to call other constructors or methods of the same class.

9.What are narrowing and widening conversions in Java?

Narrowing conversion occurs when a data type with a larger range is converted to a data type with a smaller range, potentially losing information. Widening conversion is the opposite, where a data type with a smaller range is converted to a data type with a larger range, without loss of information.

10.Provide examples of narrowing and widening conversions between primitive data types.

```
int a;
```

```
double b;
```

```
b=a; ----> widening
```

```
a = (int) b; ---> narrowing
```

11.How does Java handle potential loss of precision during narrowing conversions?

Java handles potential loss of precision during narrowing conversions by truncating the extra bits without raising any errors or exceptions. It's the responsibility of the programmer to ensure that the loss of precision is acceptable for the application's requirements.

12.Explain the concept of automatic widening conversion in Java.

Automatic widening conversion occurs when a value of a smaller data type is assigned to a variable of a larger data type. Java automatically promotes the smaller data type to the larger data type to prevent loss

of information.

13.What are the implications of narrowing and widening conversions on type compatibility and data loss?

Narrowing and widening conversions affect type compatibility and potential data loss. Narrowing conversions may lead to loss of precision, while widening conversions are generally safe but can potentially lead to unexpected behavior if not handled properly. It's crucial to understand these implications when designing and implementing Java applications.