1. Method Overloading in Java:
Method overloading in Java allows a class to have multiple methods with the same name but with different parameters. The methods can differ in the number of parameters or the data types of the parameters. Java determines which overloaded method to call based on the number and type of arguments provided during the method invocation.

Example:

java
Copy code
```
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }

    public double add(double a, double b) {
        return a + b;
    }
}
```

2. Rules for Method Overloading Resolution in Java:
Java determines which overloaded method to call based on the following rules:

The number of parameters.
Data types of the parameters.
Order of parameters if the data types are the same but the number of parameters is different.
Automatic type promotion.

3. Static Keyword in Java:

The static keyword in Java is used to define a class-level member. When a member is declared as static, it belongs to the class rather than to any specific instance of the class.
Static methods and variables can be accessed directly using the class name without creating an instance of the class.
Non-static methods and variables are associated with instances of the class and require an instance to be accessed.

4. Static Methods in Java:

Static methods can be overloaded but not overridden.
Static variables are shared across all instances of a class. They are initialized only once when the class is loaded into memory.

5. Role of the Static Keyword in Memory Management:

Static variables are allocated memory in the method area of the JVM, which is shared across all instances of the class.
Static methods are stored in the method area and can be accessed directly without creating an instance of the class.

6. Significance of the Final Keyword in Java:

The final keyword in Java is used to restrict the user from changing the value of a variable, overriding a method, or inheriting from a class.
It can be applied to variables, methods, and classes.

7. Final Method in Java:

A final method cannot be overridden by subclasses. Attempting to override a final method in a subclass will result in a compilation error.
Final variables cannot be reassigned once initialized.
Final classes cannot be subclassed.

8. This Keyword in Java:

The this keyword in Java is a reference to the current instance of the class.
It can be used to refer to instance variables and methods of the current object.
In constructors, this can be used to invoke other constructors in the same class.
It can also be used to differentiate between instance variables and local variables with the same name.

9.Narrowing and Widening Conversions in Java:

Narrowing conversion: Converting a larger data type to a smaller data type, which may result in loss of data.
Widening conversion: Converting a smaller data type to a larger data type, which does not result in loss of data.

10. Examples of Narrowing and Widening Conversions:

Widening conversion:
java
Copy code
```
int x = 10;
double y = x;  // Widening conversion from int to double
```
Narrowing conversion:
java
Copy code

```
double a = 10.5;
int b = (int) a;  // Narrowing conversion from double to int
```

11. Handling Loss of Precision during Narrowing Conversions:

During narrowing conversions, Java may truncate the fractional part of floating-point numbers or lose significant digits.
The programmer needs to be aware of potential loss of precision and handle it appropriately.

12. Automatic Widening Conversion in Java:

Java automatically performs widening conversions when the target data type can accommodate the source data type without loss of data.
For example, converting an int to a long, or a float to a double.

13. Implications of Narrowing and Widening Conversions:

Narrowing conversions may result in loss of data or precision.
Widening conversions do not result in loss of data.
Type compatibility must be considered when performing conversions to avoid data loss or unexpected behavior.