Q: What are the eight primitive data types in Java?

A: The eight primitive data types in Java are byte, short, int, long, float, double, char, and boolean.

Q: What is the default value of the byte data type in Java?

A: The default value of the byte data type is 0.

Q: What is the default value of the boolean data type in Java?

A: The default value of the boolean data type is false.

Q: What is the difference between a primitive data type and a reference data type in Java?

A: Primitive data types hold the actual values, while reference data types hold references to objects.

Q: What is the difference between int and Integer in Java?

A: int is a primitive data type, while Integer is a wrapper class that provides methods to operate on int values.

Q: What is autoboxing and unboxing in Java?

A: Autoboxing is the automatic conversion of a primitive type to its corresponding wrapper class, and unboxing is the reverse operation.

Q: What is a keyword in Java?

A: Keywords are reserved words that have a predefined meaning in the Java language and cannot be used as identifiers.

Q: Give an example of a keyword in Java.

A: Examples of keywords in Java are "public," "class," "static," "void," "if," "for," "while," etc.

Q: Can a variable name start with a digit in Java?

A: No, a variable name cannot start with a digit in Java.

Q: What is the scope of a variable in Java?

A: The scope of a variable defines its visibility, and it can be local, instance, or class level.

Q: What is a local variable in Java?

A: A local variable is a variable declared inside a method, constructor, or block and is accessible only within that block.

Q: What is an instance variable in Java?

A: An instance variable is a variable declared inside a class but outside any method, constructor, or block and is accessible to all methods of the class.

Q: What is a class variable in Java?

A: A class variable, also known as a static variable, is a variable declared with the static keyword and is shared among all instances of the class.

Q: What is the naming convention for variables in Java?

A: Variables should start with a lowercase letter and follow the camel case convention. For example, "myVariable" or "numberOfStudents."

Q: What is the difference between local and instance variables?

A: Local variables are declared inside a method and have a limited scope, while instance variables belong to an object and can be accessed by multiple methods.

Q: What is the difference between instance and class variables?

A: Instance variables belong to a specific object instance, while class variables are shared among all instances of a class.

Q: Can we change the value of a final variable in Java?

A: No, a final variable cannot be reassigned a new value once it has been initialized.

Q: What are the access modifiers in Java?

A: Java has four access modifiers: public, protected, private, and the default (no modifier).

Q: What is the default access modifier in Java?

A: The default access modifier (no modifier) allows access within the same package but not from outside the package.

Q: What does the public keyword mean in Java?

A: The public keyword is an access modifier that allows a class, method, or variable to be accessed from anywhere.

Q: What does the private keyword mean in Java?

A: The private keyword is an access modifier that restricts access to the member to within the same class.

Q: What does the protected keyword mean in Java?

A: The protected keyword is an access modifier that allows access within the same package or by subclasses in different packages.

Q: What does the static keyword mean in Java?

A: The static keyword is used to create class-level variables and methods that can be accessed without creating an instance of the class.

Q: Can the static keyword be used with a constructor in Java?

A: No, the static keyword cannot be used with a constructor in Java.

Q: What does the final keyword mean in Java?

A: The final keyword can be used with variables, methods, and classes. It indicates that the entity cannot be modified or extended.

Q: What is the difference between final, finally, and finalize in Java?

A: final is used to declare a constant or make an entity unchangeable, finally is a block used in exception handling, and finalize is a method called by the garbage collector before an object is destroyed.

Q: What does the abstract keyword mean in Java?

A: The abstract keyword is used to declare abstract classes and methods. Abstract classes cannot be instantiated, and abstract methods must be implemented by subclasses.

Q: Can we instantiate an abstract class in Java?

A: No, an abstract class cannot be directly instantiated. It can only be used as a superclass for other classes.

Q: What does the interface keyword mean in Java?

A: The interface keyword is used to declare interfaces, which define a contract that implementing classes must adhere to.

Q: Can a class implement multiple interfaces in Java?

A: Yes, a class can implement multiple interfaces by separating the interface names with commas.

Q: What is the this keyword in Java?

A: The this keyword refers to the current instance of a class and is used to differentiate between instance variables and method parameters with the same name.

Q: What is the super keyword in Java?

A: The super keyword refers to the superclass of a class and is used to access superclass members or invoke the superclass constructor.

Q: What is the difference between == and .equals() in Java?

A: The == operator checks for reference equality, while the .equals() method compares the content of objects for equality.

Q: What is the difference between == and .equals() for primitive types in Java?

A: For primitive types, == compares their values for equality since they are not objects.

Q: What is the difference between == and .equals() for strings in Java?

A: For strings, == compares the memory addresses of the string objects, while .equals() compares the actual contents of the strings.

Q: What is type casting in Java?

A: Type casting is the process of converting a value from one data type to another. It can be done implicitly (automatic) or explicitly (manual).

Q: What is implicit type casting in Java?

A: Implicit type casting, also known as widening or automatic type promotion, occurs when a smaller data type is automatically converted to a larger data type.

Q: What is explicit type casting in Java?

A: Explicit type casting, also known as narrowing or type conversion, occurs when a larger data type is manually converted to a smaller data type using parentheses and the desired type.

Q: Can you convert a string to an int in Java?

A: Yes, you can convert a string to an int using the Integer.parseInt() or Integer.valueOf() methods.

Q: Can you convert an int to a string in Java?

A: Yes, you can convert an int to a string using the String.valueOf() or Integer.toString() methods.

Q: What is the difference between StringBuilder and StringBuffer in Java?

A: StringBuilder is not thread-safe and more efficient, while StringBuffer is thread-safe but less efficient.

Q: What is the NaN value in Java?

A: NaN stands for "Not a Number" and represents an undefined or unrepresentable value in floating-point calculations.

Q: What is the null value in Java?

A: null is a special value that represents the absence of an object reference. It is often used to indicate that a variable does not refer to any object.

Q: What is the difference between null and an empty string ("") in Java?

A: null represents the absence of an object reference, while an empty string is a valid string object with zero length.

Q: What is the difference between a local variable and an instance variable?

A: Local variables are declared within a method or block and have a limited scope, while instance variables belong to an object and can be accessed by multiple methods.

Q: What is the difference between final and finally in Java?

A: final is used to declare a constant or make an entity unchangeable, while finally is a block used in exception handling to execute code regardless of whether an exception occurred or not.

Q: What is the difference between break and continue in Java?

A: break is used to terminate a loop or switch statement, while continue is used to skip the remaining code in a loop and move to the next iteration.

Q: What is the difference between do-while and while loops in Java?

A: A do-while loop executes the code block at least once, while a while loop checks the condition first and may skip the code block entirely.

Q: What is the difference between && and & in Java?

A: && is a short-circuit logical AND operator, which means the second operand is only evaluated if the first operand is true. & is a non-short-circuit logical AND operator that always evaluates both operands.

Q: What is the difference between || and | in Java?

A: || is a short-circuit logical OR operator, which means the second operand is only evaluated if the first operand is false. | is a non-short-circuit logical OR operator that always evaluates both operands.