Assign-6

Interview question Set 2

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

Ans:- • Method overloading in java means having two or more methods (or functions) in a class with the same name & different arguments (or parameters).
• It can be with a different no. of arguments or different data types of arguments or different data types of arguments.
• for instance: void function1(double a) {...}

for example,
if the parameters of method 1 are (String name, int roll_no) and the other method is (int roll_no, String name) but both have the same name, then these 2 methods are considered to be overloaded with different seq. of parameters.

Q.2 What are the rules for method overloading resolution in java? How does java determine which overloaded method to call?

Ans:- • The Method parameters must change; either the no. or the type of parameters must be different in the two methods.
• The return type can be freely modified.
• The return type can be freely modified
• The access modifier (public, private, & so on) can be freely modified.
• Thrown exceptions, if any, can be freely modified.

· Java determine method overloading based on the no. & types of parameters in the method signature.
· When you define multiple methods with the same name within a class, Java allows you to overload those methods by providing different parameter lists.

Q.3 What does the Static keyword mean in java? Explain the difference betn Static & non Static methods.

Ans: Static keyword in java indicates that a particular member is not an instance, but rather part of a type.
· The Static member will be shared among all instances of the class, so we will only create one instance of it.

Static Method:
A static method is a class method & belongs to class itself. This means you do no need an instance in order to use a static Method.

Non Static Method:
A non-Static Method is an instance method & belongs to each object i.e. generated from the class.

Q.4 Can Static methods be overloaded & overridden in java? How are Static variables shared across multiple instance of a class?

Ans: Yes, we can have 2 or more Static methods with the same name, but differences in

input parameters.

- To store information that is shared across instances of a class, we a static variable. All instances of the same class share a single copy of the static variable

Q.5 What is the role of the static keyword in the context of memory mgmt.

Ans:
- The static keyword in java is mainly used for mem mgmt.
- The static keyword in java is used to share the same variable or method of a given class.
- The users can apply static keywords with variables, methods, blocks & nested classes.
- The static keyword belongs to the class than an instance of the class.

Q.6 What is the significance of the final keyword in java?

Ans:
- The final keyword is a non-access modifier used for classes, attributes & methods, which makes them non-changeable (impossible to inherit or override).
- The final keyword is useful when you want a variable to always store the same value like PI (3.14159...).
- The final keyword is called a "modifier

Q.7 Can a final method be overriden in a subclass? How does the final keyword affe

variables, methods, & classes in java?

Ans: No, the methods that are declared as final cannot be overridden or hidden.

- In java, the keyword 'final' serves as a non-access modifier applicable to classes, methods, and variables.
- A 'final' class cannot be subclassed, a 'final' method cannot be overridden, and a 'final' variable cannot be reassigned once initialized.

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

Ans: The keyword 'this' in java serves a fundamental purpose: it refers to the current object. In other word, 'this' represents the instance of the class where it's used.

- It's commonly used to access or modify the fields of the current object, especially when field names are the same as local variable names.
- The this keyword refers to the current object in a method or constructor.
- The most common use of the this keyword is to eliminate the confusion bet$^n$ bet$^n$ attributes and parameters with the same name (because a class attribute is shadowed by a method or constructor parameter).

Q.9 What are narrowing & widening conversions in java?

Ans: Widening conversions preserve the source value

but can change its representation.

- This occurs if you convert from an integral type to Decimal, or from char to Single.
- A narrowing conversion changes a value to a data type that might not be able to hold some of the possible values.

Q.10 Provide Examples of narrowing & Widening Conversions both primitive data types.

Ans:

Widening Conversions :-

| Data types | Widens to data types |
|---|---|
| SByte | SByte, Short, Integer, Long, Decimal, Single, Doub |
| Byte | Byte, Short, UShort, Integer, UInteger, Long, ULong, Decimal, Single, Double. |
| Short | Short, Integer, Long, Decimal, Single, Double |
| UShort | UShort, Integer, UInteger, Long, ULong, Decimal, Single Double. |
| Integer | Integer, Long, Decimal, Single, Double |
| UInteger | UIntger, Long, ULong, Decimal, Single, Doubl |
| Long | Long, Decimal, Single, Double. |
| ULong | ULong, Decimal, Single, Double |
| Decimal | Decimal, Single, Double |
| Single | Single, Double |
| Double | Double. |

- Narrowing Conversions:
- The reverse directions of the Widening Conversions in the preceding table (except that every type widens to itself)
- Conversions in either dirn betn boolean & any numeric type.
- Conversions from any numeric type to any enumerated type (Enum)
- Conversions in either dirn betn String & numeric type, Boolean, or Date.
- Conversions from a data type or object type to a type derived from it.

11 How does java handle potential loss of precision during narrowing Conversions?

Lossy Conversion:

- Lossy Conversion is simply the loss of infon while handling data.
- In java, it corresponds to the possibility of losing the value or precision of a variable while Converting one type to another.
- When we try to assign a variable of larger sized type to smaller sized type, java will generate an error, incompatible types: possible lossy Conversion, while Compiling the code.

12 Explain the concept of automatic Widening Conversion in java.

In java, there are 2 types of Casting:
Widening Casting (automatically) - Converting a smaller type to a larger type size

byte -> Short -> Char -> int -> long -> float -> double.

- Narrowing Casting (manually) - Converting a larger type to a smaller size type

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

Ans:
- Widening Conversions preserve the source value but can change its representation.
- This occurs if you convert from an integral type to Decimal, or from char to String.
- A narrowing conversion changes a value to a data type that might not be able to hold some of the possible values