Q.1 Why Coding Standards?

Ans: Coding standards are a set of guidelines, rules, and best practices that developers follow while writing code. They are not enforced by the compiler but help make code readable, maintainable, and error-free.

1. Readability: Makes code easier for others (and yourself) to read and understand. Example: Consistent naming of variables and methods (`camelCase` for variables, `PascalCase` for classes).
2. Maintainability:

   Standardized code is easier to modify or debug in the future.

   Teams can quickly identify where changes are needed.

3. Consistency:

   Multiple developers working on the same project produce uniform code.

   Prevents confusion caused by different coding styles.

4. Reduces Errors:

   Following naming conventions, indentation, and formatting reduces logical and syntactical mistakes.

5. Improves Collaboration:

   When everyone follows the same rules, team collaboration becomes smoother.

6. Ease of Code Review:

   Standardized code is easier to review and audit.

   Makes spotting bugs faster.

7. Supports Tools & Automation:

Tools like SonarQube, Checkstyle, or IDE linters work better when code follows standards.

2) If we don’t follow coding standard and Naming conventions will it lead to

Compile time error or Runtime error?

Ans: Not following coding standards or naming conventions will NOT cause compile-time or runtime errors.

They are guidelines, not strict rules enforced by the compiler.

3)What is base class of all classes?

Ans:Object class is the base class of all classes.

- Every class in Java directly or indirectly inherits from `java.lang.Object`.
- Even if you don’t write `extends Object`, the compiler automatically adds it.

4)Does java support multiple inheritance?

Ans: Java does NOT support multiple inheritance with classes

But Java DOES support multiple inheritance with interfaces.

5)Is array primitive data type?

Ans: No, array is NOT a primitive data type in Java.

Array is an object in Java.Even if it stores primitives, the array itself is stored as an object in the heap memory. Its class type is created at runtime (e.g., `int []` is a class).

Q6. What is difference between path and Class path?

Q7. What are local variables?

Ans: variables that are intilized inside the Block, constructor, Methods.

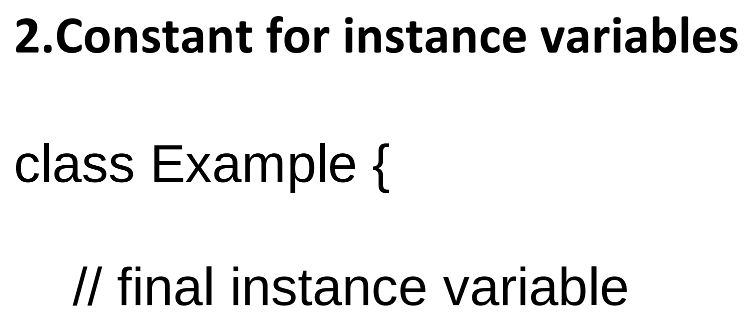
Q.8. How to define constant variable in Java?

Ans: A constant variable is a variable whose value cannot be changed once it is assigned. In Java, constants are usually written in uppercase letters with underscores _ between words.

1. Constant at class level (static constant)

```
class Example {
    // constant variable
    public static final double PI = 3.14159;
}
```

Constant for instance variables



2.Constant for instance variables

```
class Example {
    // final instance variable
    final int DAYS_IN_WEEK = 7;
}
```

3. **Local constant inside a method**

```
public class Example {
    public static void main (String [] args) {
        final int MAX_SCORE = 100;
        System.out.println("Max score: " +
MAX_SCORE);
    }
}
```

Q.9. **Should main () method be compulsory declared in all java classes?**

Ans: **No**, the `main ()` method is not compulsory in all Java classes.

Only **the class you want to run directly** must have a `main()` method.

Q10. **What is return type of main() method?**

Ans:The return type of main method is void because main method is used for execute the program.Not to perform any particular activity.

11)Why is main() method declared static?

Ans: Because the JVM (Java Virtual Machine**)** needs to call the `main ()` method without creating an object of the class. When we make main method as a static then we don't need to create an object.

12)What is argument of main method?

Ans: `args` is an array of Strings.

It is used to store command-line arguments that are passed to the program when it starts.

13)Can a main () method be overloaded?

Ans: Yes, main method can be overloade. You can overload `main ()` in Java, but the JVM only calls the version with `String [] args`.
All other overloaded `main ()` methods must be called manually.

14) Can a main method be declared as final?

Ans: Yes, we can declare main method as final in java. JVM does not care about it's final or normal method.

15)Does and static order of public and static declaration matter on main () method?

Ans: No. order of the public and static does not matter.

16) Can s source file contain more than one class declaration?

Ans: Yes, a single Java source file can contain more than one class declaration.

17) What is package?

Ans: A package in Java is a group of related classes, interfaces, and sub-packages.

It's like a folder in your computer that helps you organize your Java files.

18) Which package is imported by default?

Ans: The `java. Lang` package is automatically imported by the Java compiler.

It contains the most fundamental classes that are required in almost every Java program.

That's why you can use many classes without writing an import statement.

19)Can a class declared as protected?

Ans: You cannot declare a top-level class as `protected`, but you can declare an inner class as `protected`.