Java Basics Coding Challenges

1. Primitive Data Types

1. **Challenge**: Declare and initialize variables of all primitive types and print their default values.
2. **Challenge**: Write a program to detect overflow when adding two `byte` variables.
3. **Challenge**: Use type casting to convert `double` to `int` and `float` to `byte`.
4. **Challenge**: Perform bitwise operations between `int` and `byte`.
5. **Challenge**: Accept input for all primitive types and display them formatted.

2. Variables

1. **Challenge**: Swap two numbers using a temporary variable.
2. **Challenge**: Swap two numbers without a temporary variable.
3. **Challenge**: Demonstrate variable shadowing within a class and method.
4. **Challenge**: Declare a constant and use it in calculations.
5. **Challenge**: Create a class with instance, static, and local variables and demonstrate scope.

3. Operators

1. **Challenge**: Demonstrate all arithmetic operators using two integers.
2. **Challenge**: Use relational operators to compare ages.
3. **Challenge**: Implement a basic calculator using switch and operators.
4. **Challenge**: Use bitwise AND, OR, XOR on two binary values.
5. **Challenge**: Demonstrate logical operators with Boolean expressions.

4. String Concatenation

1. **Challenge**: Concatenate first name and last name.
2. **Challenge**: Combine name, age, and address using string concatenation.
3. **Challenge**: Use concatenation inside a loop to build a pattern.
4. **Challenge**: Demonstrate precedence of concatenation and addition.

5. **Challenge**: Accept input strings and concatenate with formatting.

5. StringBuilder

1. **Challenge**: Reverse a string using `StringBuilder`.
2. **Challenge**: Append multiple strings using `StringBuilder` and print.
3. **Challenge**: Replace characters in a string using `StringBuilder`.
4. **Challenge**: Insert a word into a string at a specific position.
5. **Challenge**: Delete part of a string using `StringBuilder`.

6. String API

1. **Challenge**: Use `charAt()`, `length()`, and `substring()` methods.
2. **Challenge**: Count the number of vowels in a string.
3. **Challenge**: Check if a string is a palindrome.
4. **Challenge**: Convert a string to upper case and lower case.
5. **Challenge**: Remove spaces and special characters from a string.

7. Date, Time, and Numeric Objects

1. **Challenge**: Get current date and time using `LocalDateTime`.
2. **Challenge**: Calculate age given a birth date.
3. **Challenge**: Format date in dd-MM-yyyy format.
4. **Challenge**: Add 5 days to current date and print.
5. **Challenge**: Round a decimal to 2 places using `BigDecimal`.

8. Flow Control

1. **Challenge**: Use if-else to determine if a number is positive, negative, or zero.
2. **Challenge**: Implement nested if to find the largest among 3 numbers.
3. **Challenge**: Validate login with username and password.
4. **Challenge**: Categorize age groups using if-else ladder.

5. **Challenge**: Determine student grade using percentage.

9. Conditions

1. **Challenge**: Check if a number is even and divisible by 5.
2. **Challenge**: Validate a triangle (sum of angles = 180).
3. **Challenge**: Check if year is a leap year.
4. **Challenge**: Check character type (vowel/consonant/digit/special).
5. **Challenge**: Check eligibility for vote, driving, and job using conditions.

10. Switch

1. **Challenge**: Create a calculator using switch.
2. **Challenge**: Map number to month name using switch.
3. **Challenge**: Implement a simple menu using switch.
4. **Challenge**: Use enhanced switch (Java 14+) for better syntax.
5. **Challenge**: Implement day of the week based on integer input.

11. Loop & Branching

1. **Challenge**: Print multiplication table for a number.
2. **Challenge**: Use break and continue in loops.
3. **Challenge**: Find factorial of a number.
4. **Challenge**: Print Fibonacci series.
5. **Challenge**: Find sum of even numbers from 1 to 100.

12. Arrays

1. **Challenge**: Find the largest and smallest element in an array.
2. **Challenge**: Sort an array in ascending order.
3. **Challenge**: Calculate average of numbers in an array.
4. **Challenge**: Count occurrence of an element.
5. **Challenge**: Reverse elements of an array.

13. Enum

1. **Challenge**: Define an enum for days of the week.
2. **Challenge**: Use enum in switch case.
3. **Challenge**: Iterate over enum values.
4. **Challenge**: Assign properties to enum constants.
5. **Challenge**: Create an enum to represent traffic light states.