9. QB - Basic Functional interface and lambda questions

🔢 **Assignment Questions**

1. **Use a `Supplier<Integer>` to provide a number, a `Predicate` to check if it's positive, and a `Consumer` to print "Positive Number" if true.**
2. **Create a `Function<String, Integer>` to return the length of a string. Use it to get the length of a name provided by a `Supplier<String>`, and print the result using a `Consumer<Integer>`.**
3. **Use a `Supplier<Double>` to provide a product price. If the price is greater than 1000 (use `Predicate`), apply a 10% discount (use `Function`) and print the final price.**
4. **Use a `Supplier<Integer>` to generate an age, and use a `Predicate` to check if it's eligible for voting (>=18). Print "Eligible" or "Not Eligible" accordingly.**
5. **Create a `Predicate<String>` to check if a string starts with "A". Use it with a `Supplier<String>` that supplies a name. If it matches, print "Name Accepted".**
6. **Use a `Function<Integer, Integer>` to get the cube of a number. Apply it to a number supplied by `Supplier<Integer>` and print the result.**
7. **Create a program that uses `Predicate<Integer>` to check if a number is odd, and if it is, print its square using `Function` and `Consumer`.**
8. **Use `Predicate<String>` to check if a password length is at least 8. If yes, print "Strong Password". The password should be provided by `Supplier<String>`.**
9. **Use `Function<String, String>` to convert a string to uppercase. Provide the string using a `Supplier` and print the result using `Consumer`.**
10. **Create a `Predicate<Double>` to check if a temperature is above 37.5 (fever). Use `Supplier<Double>` for temperature, and if true, print "High Temperature".**
11. **Use a `Supplier<Integer>` to provide two numbers (separately), use `Function<Integer, Integer>` to double them, and print their sum.**

12. **Create a `Supplier<List<String>>` to provide a list of names. Use a `Consumer<List<String>>` to print each name using enhanced for-loop.**
13. **Use a `Function<String, Boolean>` (or `Predicate<String>`) to check if a string is a palindrome. Input is from `Supplier<String>`, output printed using `Consumer<Boolean>`.**
14. **Use `Predicate<Integer>` to check if a number is divisible by 3 or 5. Use `Function<Integer, String>` to return a message accordingly: "Fizz", "Buzz", or "FizzBuzz".**
15. **Create a program where `Supplier<String>` provides a username. If username starts with "admin" (`Predicate`), print "Access Granted", else "Access Denied".**

🔹 **BiFunction**

1. **Create a `BiFunction<Integer, Integer, Integer>` that takes two integers and returns their sum. Use it to add 5 and 10, and print the result.**
2. **Write a `BiFunction<String, Integer, String>` that repeats a string the given number of times. For example, input ("Hi", 3) → Output: "HiHiHi".**
3. **Use a `BiFunction<String, String, Integer>` to return the total length of two strings combined.**
4. **Write a `BiFunction<Double, Double, Double>` that calculates the area of a rectangle (length × breadth) and prints the result.**

🔹 **BiPredicate**

5. **Create a `BiPredicate<String, String>` that checks if two strings are equal (case-insensitive). Test it with inputs like ("hello", "HELLO").**
6. **Use a `BiPredicate<Integer, Integer>` to check if the first number is divisible by the second.**
7. **Write a `BiPredicate<String, Integer>` that returns true if the length of the string is greater than the given number.**

🔹 **BiConsumer**

8. **Create a `BiConsumer<String, Integer>` that prints a formatted message like: "Name: John, Age: 25".**
9. **Write a `BiConsumer<Integer, Integer>` that prints the result of addition, subtraction, multiplication, and division of the two numbers.**
10. **Create a `BiConsumer<String, String>` that prints a combined greeting message, like: "Hello Alice and Bob!".**

Document by ***Suyash*** 😇