Dart & Flutter Fundamentals – Complete Practice & Assessment

This document contains **practical questions, MCQs, guess-the-output problems, coding tasks, and logical hard questions** covering all topics we have studied so far.

Topics Covered: - Variables - Data Types - Naming Rules - final & const - Nullable Variables - Null-Aware Operators - Dynamic Variables - String Methods & Interpolation - Set - Map - Flutter-based usage

---

Section A: Practical Real-Life Questions

1. In a login form, which fields should be nullable and why?
2. Why should a user ID be stored in a `final` variable?
3. How would you use a Map to store user profile information?
4. Where would you use a Set in a shopping app UI?
5. Why is `Map<String, dynamic>` used for API responses?
6. How would bad variable naming affect a team project?
7. Where can `??` prevent app crashes?
8. Why is `dynamic` risky in production apps?
9. How would you store cart items using a Map?
10. Why is a Set better for unique email storage?

Section B: MCQs

Choose the correct answer.

1. Which keyword makes a variable nullable? a) var b) final c) ? d) dynamic

2. Which operator provides a default value if null? a) ! b) ?? c) ?. d) ==

3. Which data structure does not allow duplicates? a) List b) Map c) Set d) String

4. What happens when you assign a new value to an existing Map key? a) Error b) Ignored c) App crashes d) Old value is replaced

5. Which is safer? a) name! b) name?.length c) name.length d) dynamic

---

Section C: Guess the Output

6.

```
String? name;
print(name ?? "Guest");
```

7.

```
int x = 10;
print(x.isEven && x.isOdd);
```

8.

```
Set<int> data = {1, 2, 3};
data.add(3);
print(data.length);
```

9.

```
Map<String, int> cart = {"Apple": 2};
cart["Apple"] = 5;
print(cart["Apple"]);
```

10.

```
String name = "dart";
print(name.toUpperCase());
```

Section D: Coding Questions

1. Write a program to swap two variables.
2. Write a program that removes duplicates from a list using a Set.
3. Write a program that stores student marks using a Map.
4. Write a program that uses string interpolation to print a bill.
5. Write a program that uses nullable variables and null-aware operators safely.

Section E: Logical & Hard Questions

1. Why is null safety important for mobile apps?
2. What logical bug can occur if you use a List instead of a Set for voting?
3. Why is using two separate Lists worse than using a Map?
4. How does bad naming affect debugging?
5. Design a data structure for a movie booking app using Set and Map.

Section F: Flutter-Based Thinking

1. Where would you use Set in UI state management?
2. Why do Flutter forms use Maps?
3. How do null-aware operators help prevent red screens?
4. Why is `dynamic` used when decoding JSON?
5. How would you store liked posts using a Set?

Tip for Students

Always think: - Safety - Readability - Performance - Real-life behavior

Do not just write code—**understand why it works.**

If you want, I can: - Add full answer key - Add explanations - Add difficulty levels - Convert this into tests or worksheets - Add Flutter mini-projects