

1. What is the purpose of the ? operator in Dart null safety?

The ? operator in Dart is used to **make a variable nullable**, meaning it can hold either a **value** or **null**.

Example:

```
int? age; // This means age can be an int or null.
```

This prevents null-related bugs by making nullability explicit and checked at compile time.

2. Create a `late` variable named `address`, assign a US value to it, and print it.

The `late` keyword tells Dart that the variable **will be initialized later**, but **not null** when used.

Example:

```
void main() {  
  late String address; // Declare the variable, will be initialized  
  later  
  address = "USA";      // Assign value  
  print("Address: $address"); // Print it  
}
```

3. How do you declare a nullable type in Dart null safety?

To declare a nullable type, add a ? after the type name:

```
String? name; // name can be null or a String  
int? age;     // age can be null or an int
```

4. Write a program in Dart to create an age variable and assign a null value to it using ?.

```
void main() {  
  int? age = null; // Nullable int variable with null value  
  print("Age: $age");  
}
```

5. Write a function that accepts a nullable int parameter and returns 0 if the value is null using null coalescing operator ??.

```
int handleNull(int? value) {  
    return value ?? 0; // If value is null, return 0  
}  
  
void main() {  
    print(handleNull(null));    // Output: 0  
    print(handleNull(42));     // Output: 42  
}
```

6. Write a function named `generateRandom()` in Dart that randomly returns 100 or null. Also, assign a return value of the function to a variable named `status` that can't be null. Give `status` a default value of 0 if `generateRandom()` returns null.

```
import 'dart:math';  
  
int? generateRandom() {  
    Random random = Random();  
    return random.nextBool() ? 100 : null; // 50% chance of null or 100  
}  
  
void main() {  
    int status = generateRandom() ?? 0; // Use null coalescing operator  
    print("Status: $status");  
}
```