|  |  |  |  |
| --- | --- | --- | --- |
| **Mobile Applications and Web Development -IS4904(Practical)** | | | |
| **Student Name: رغد فيصل القجيز** | **Student ID:443051831** | | **Section:1989** |
| **Assignment (Exploring the Health Informatics Stack)** | | | |
| **Date: 5th March 2024** | | **Max Points:** | |

**Submission Guidelines:**

Submit the essay and Dart program files via the designated submission platform.

Ensure code is well-commented and follows Dart style conventions.

**Question** 1: Introduction to Null Safety in Dart

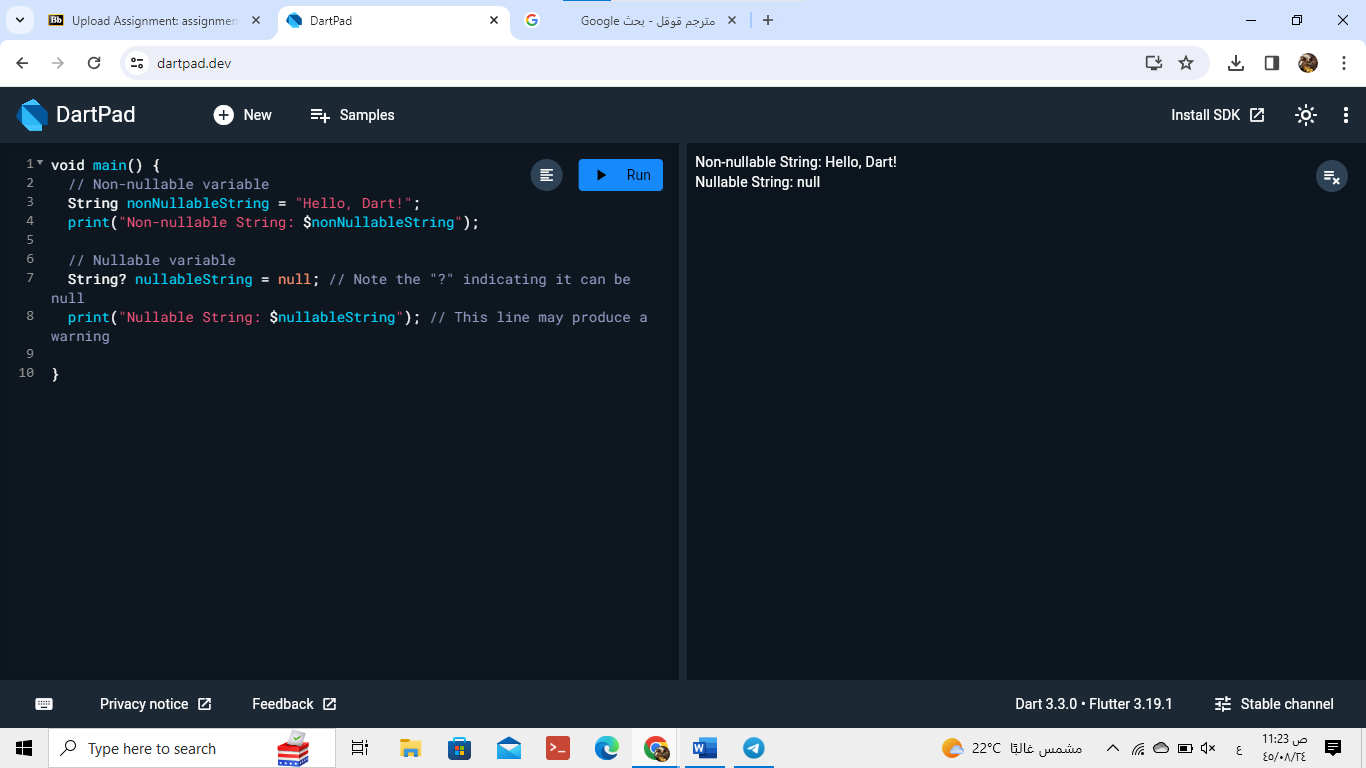
**Objective**:

To reinforce understanding of the concept of null safety in Dart and its significance in writing reliable code.

**Tasks**:

1. Write a short essay explaining the importance of null safety in Dart programming(two lines max).

**Dart's null safety acts as a safety net, catching potential null errors at compile time, not runtime, resulting in more robust and manageable code.**

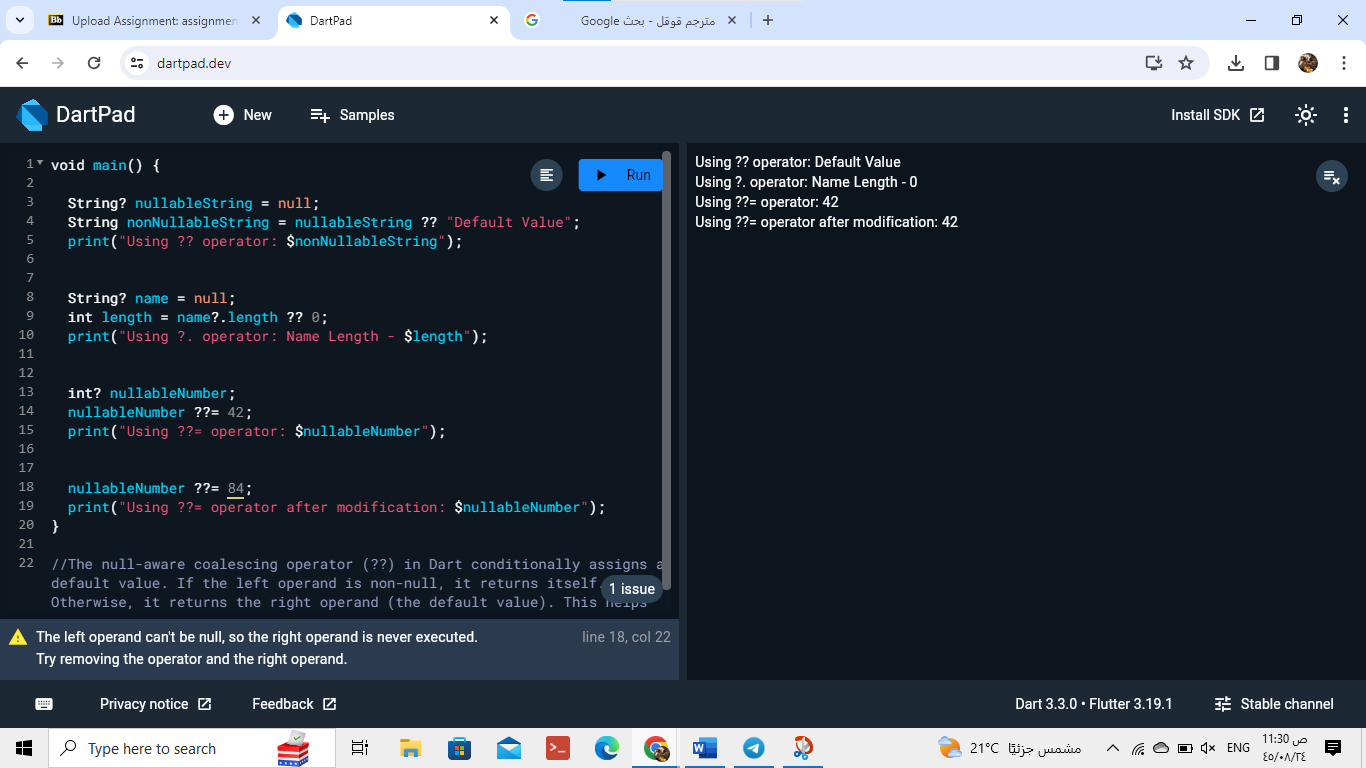
1. Create a Dart program that demonstrates the difference between nullable and non-nullable variables. Use both nullable and non-nullable variables in your program and explain how null safety affects their usage. 

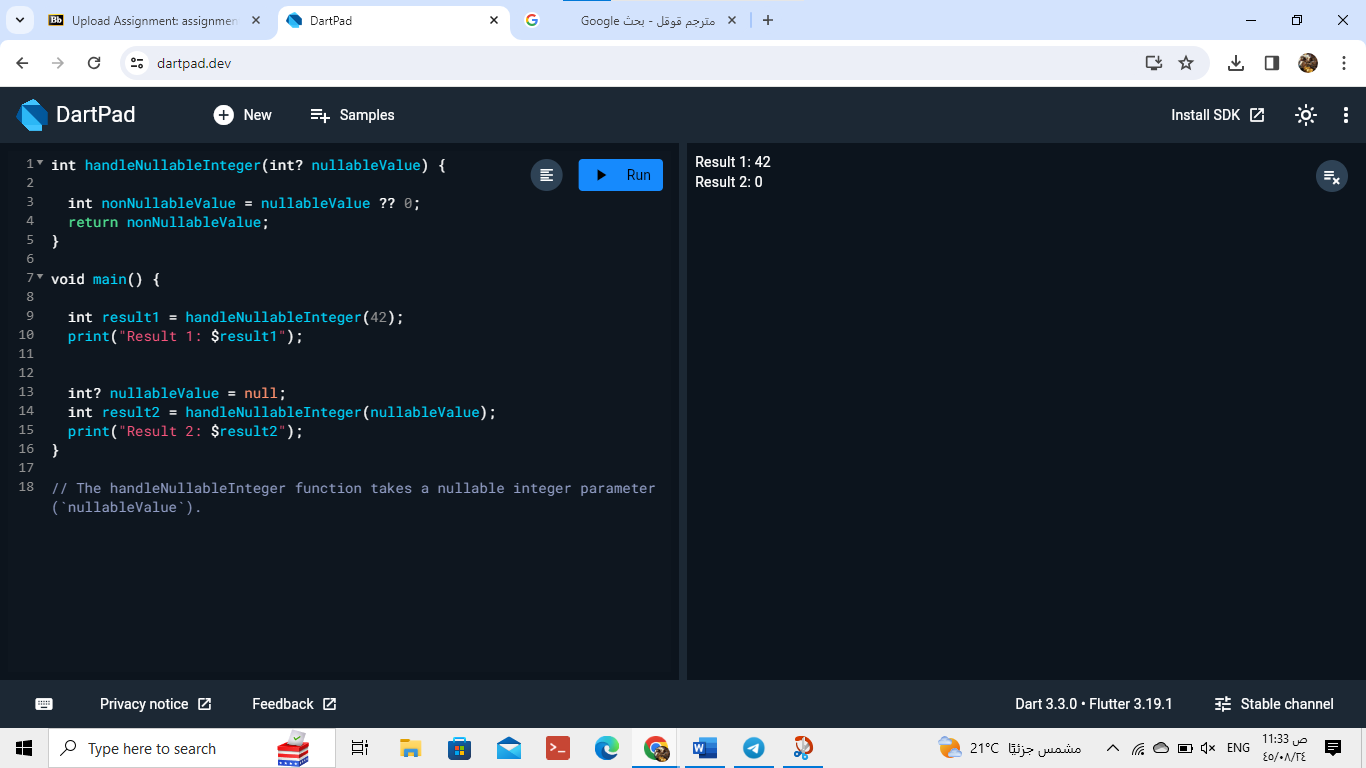
**Question** 2: Handling Null Values in Dart

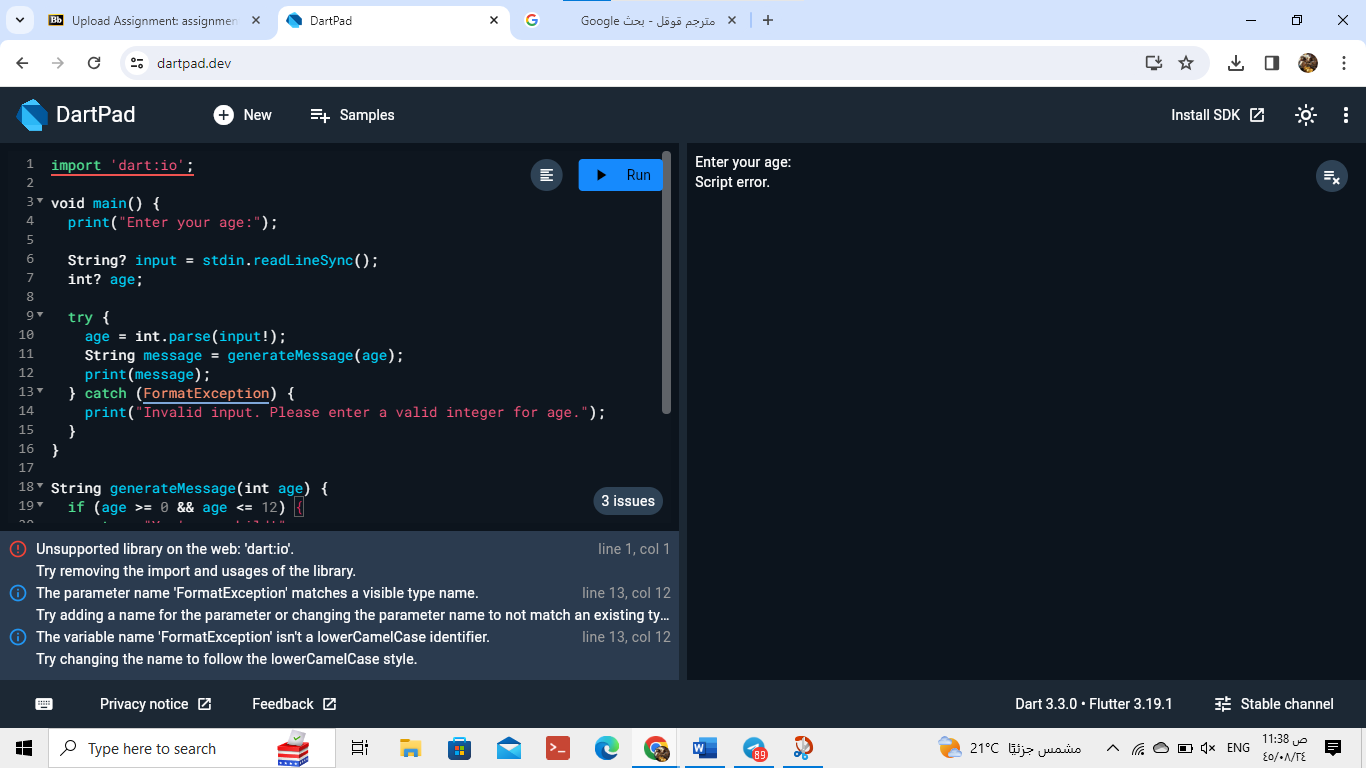
**Objective**:

To practice working with nullable types and null-aware operators in Dart to effectively handle null values.

**Tasks**:

1. Write a Dart program that demonstrates the use of null-aware operators (??, ?., ??=) to handle null values. Include examples of each operator in your program and explain their purpose. 
2. Write a Dart program and develop a function that takes a nullable integer parameter and returns a non-nullable integer. Handle null values appropriately within the function and explain your approach.



1. Create a Dart program that reads user input for a person's age and prints a customized message based on the input. Ensure the program handles null values and invalid input gracefully.
2. 

This code :

import 'dart:io';

void main() {

print("Enter your age:");

String? input = stdin.readLineSync();

int? age;

try {

age = int.parse(input!);

String message = generateMessage(age);

print(message);

} catch (FormatException) {

print("Invalid input. Please enter a valid integer for age.");

}

}

String generateMessage(int age) {

if (age >= 0 && age <= 12) {

return "You're a child!";

} else if (age >= 13 && age <= 19) {

return "You're a teenager!";

} else if (age >= 20) {

return "You're an adult!";

} else {

return "Invalid age.";

}

}

**Question** 3: Handling Null Values in Dart

**Objective**:

To delve deeper into advanced techniques for handling null values in Dart and practice implementing null-safe code patterns.

**Tasks**:

1. Write a Dart program that utilizes the **late** keyword and null assertion operator (!) to handle delayed initialization of non-nullable variables.
2. 