

1. Explain the difference between parse and tryParse.

Provide an example that shows what happens when the string is invalid in both cases.


```
answers.dart U x
Day2 > answers.dart > main
1  import 'dart:io';
2
3  void main() {
4      stdout.write("Enter Number to add one: ");
5      String userInput = stdin.readLineSync();
6
7      int? numberPlusOne = int.tryParse(userInput);
8      // int? numberPlusOne = int.parse(userInput);
9
10     if (numberPlusOne != null) {
11         print(numberPlusOne + 1);
12     } else {
13         print("Invalid input");
14     }
15     // ! tryParse return Null if not valid which can be checked , for parse there will be expcetion
16     // ! if we dont handle it , app will crash
17 }
18
```

2. Handle empty string input from the user in two different ways:

Using a while loop, allow the user to try again up to 3 attempts.

Using a for loop, allow the user to try again up to 5 attempts.

```
2
Run | Debug
3 void main() {
4     int counter = 0;
5     String? userInput;
6
7     while (counter < 3) {
8         stdout.write("Enter something while ");
9         userInput = stdin.readLineSync();
10        if (userInput != null && userInput.isNotEmpty) {
11            print("done");
12            break;
13        } else {
14            print("try again ");
15        }
16        counter++;
17    }
18
19    for (int i = 1; i ≤ 5; i++) {
20        stdout.write("Enter something for ");
21        userInput = stdin.readLineSync();
22
23        if (userInput != null && userInput.isNotEmpty) {
24            print("done");
25            break; // exit loop if valid
26        } else {
27            print(" try again");
28        }
29    }
30 }
```



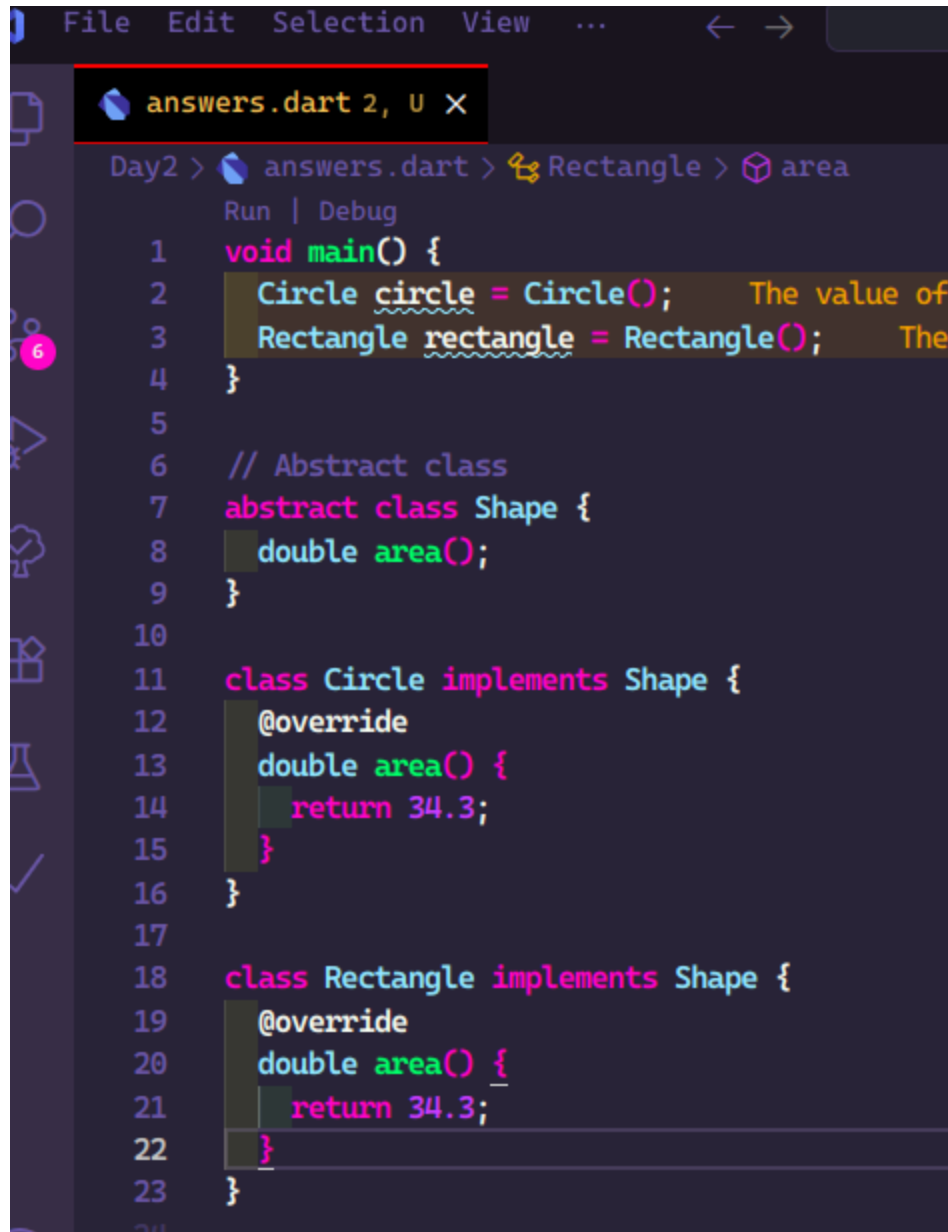
3. Explain the purpose of try-catch. Why do we use it?
4. Use try-catch to handle the error when dividing by zero

```
answers.dart U x
Day2 > answers.dart > main
Run | Debug
1 void main() {
2   try {
3     double x = 10 / 0;
4     print(x);
5   } catch (e) {
6     print("""
7       We use try catch to run code that can crash and raise exception ,
8       we try running it , if raising an error, we catch error (something like if else) to handle it
9     """);
10  }
11 }
12
```

5. What is an abstract class in Dart? Explain in your own words.

```
answers.dart U x
Day2 > answers.dart > main
Run | Debug
1 void main() {
2   print("""
3     Abstract class is like a model or rules ( rules like certain methods ) so that other classes
4     extend it and implmeent it , it is not to create objects or for direct use
5   """);
6 }
7
8 abstract class SomethingAbstract {}
9
```

6. Create an abstract class Shape with an abstract method area().  
Create two classes Circle and Rectangle that implement the Shape and area() method.

A screenshot of an IDE window titled 'answers.dart 2, U X'. The breadcrumb navigation shows 'Day2 > answers.dart > Rectangle > area'. The code is as follows:

```
1 void main() {  
2   Circle circle = Circle();  
3   Rectangle rectangle = Rectangle();  
4 }  
5  
6 // Abstract class  
7 abstract class Shape {  
8   double area();  
9 }  
10  
11 class Circle implements Shape {  
12   @override  
13   double area() {  
14     return 34.3;  
15   }  
16 }  
17  
18 class Rectangle implements Shape {  
19   @override  
20   double area() {  
21     return 34.3;  
22   }  
23 }
```

7. Create a base class Person with properties name and age.  
Create a subclass Student that extends Person and adds grade. Then print the student's details.

The screenshot shows an IDE window with a file named `answers.dart`. The code defines a `Person` class and a `Student` class that extends `Person`. The `main` function creates a `Student` object and prints its details.

```
Day2 > answers.dart > Student
Run | Debug
1 void main() {
2   Student student = Student("aymn", 30, "A+");
3   student.printDetails();
4 }
5 class Person {
6   String name;
7   int age;
8   Person(this.name, this.age);
9 }
10 class Student extends Person {
11   String grade;
12   Student(super.name, super.age, this.grade);
13   void printDetails() {
14     print("Name: $name , Age: $age ,Grade: $grade");
15   }
16 }
17
```

The terminal output shows the command `$ dart answers.dart` being executed, resulting in the printed details: `Name: aymn, Age: 30, Grade: A+`.

8. Difference Between extends and implements. Explain the difference in Dart.

**Answer :** **implements** is used as the name says for implementing abstract class ( where code is not implemented , just definitions ) while **extend** is to make the inheritance principle in OOP

9. ATM Example: Create a simple ATM program

**Answer :** it is in [ `ATM.dart` ] file in the folder

10. Create a class 'book'

**Answer :** it is in [ `Book.dart` ] file in the folder

**11. Explain the purpose of static?**

**Static simply mean that , define value or logic to be accessible globally from the class , as the name says “static” since those defined as static usually contain something static that is not changing per object**