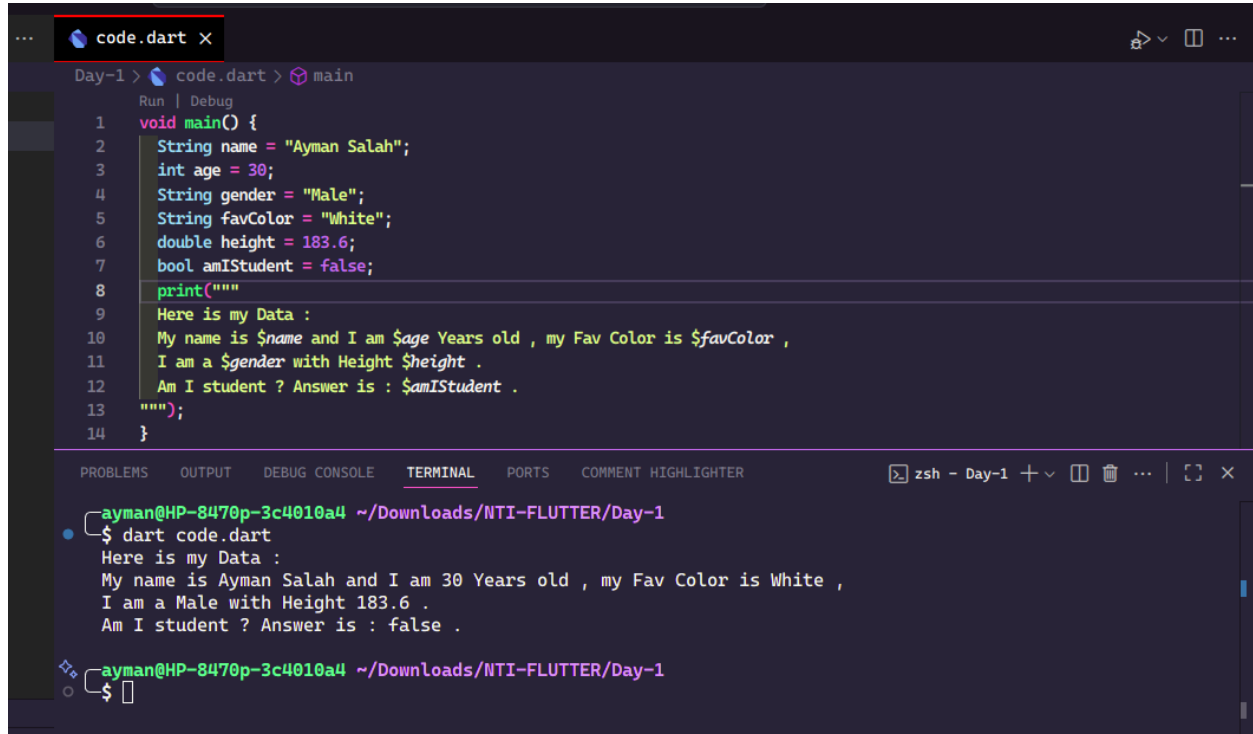


1. Write a program that defines variables representing your personal data using the appropriate data types:

Name (String) - Age (int) - Gender (String) - Favorite color (String) - Height (double) - Are you a student? (bool)

Then print all the data.



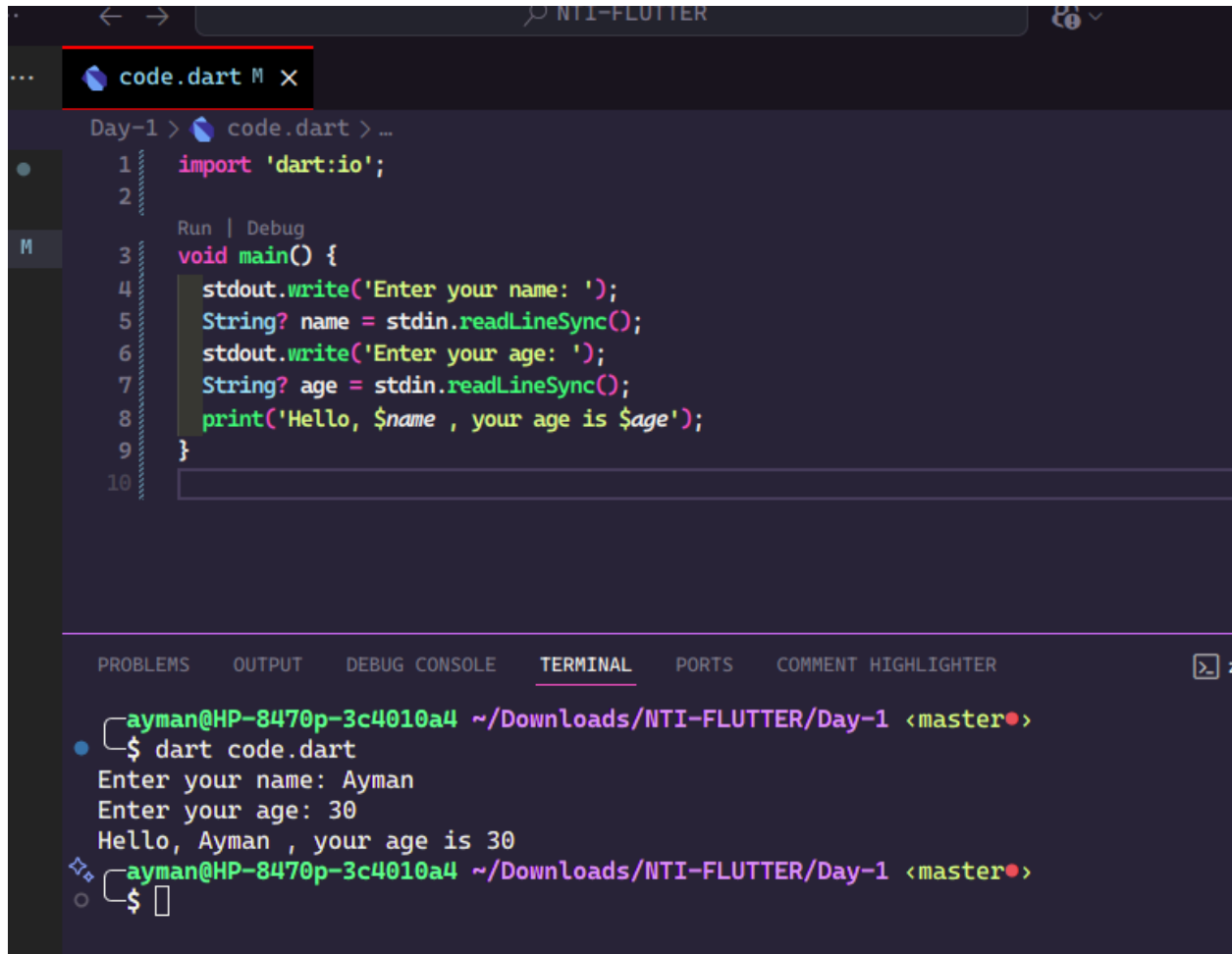
The screenshot shows an IDE window titled 'code.dart'. The editor contains the following Dart code:

```
1 void main() {  
2   String name = "Ayman Salah";  
3   int age = 30;  
4   String gender = "Male";  
5   String favColor = "White";  
6   double height = 183.6;  
7   bool amIStudent = false;  
8   print("""  
9     Here is my Data :  
10    My name is $name and I am $age Years old , my Fav Color is $favColor ,  
11    I am a $gender with Height $height .  
12    Am I student ? Answer is : $amIStudent .  
13    """);  
14 }
```

Below the editor is a terminal window. It shows the command `$ dart code.dart` being executed, followed by the output of the program:

```
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1  
$ dart code.dart  
Here is my Data :  
My name is Ayman Salah and I am 30 Years old , my Fav Color is White ,  
I am a Male with Height 183.6 .  
Am I student ? Answer is : false .  
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1  
$
```

2. Write a program that takes the name and age from the user (input) and prints them.



The image shows a screenshot of an IDE with a dark theme. At the top, a tab labeled 'code.dart' is active. The editor displays the following Dart code:

```
1 import 'dart:io';
2
3 void main() {
4     stdout.write('Enter your name: ');
5     String? name = stdin.readLineSync();
6     stdout.write('Enter your age: ');
7     String? age = stdin.readLineSync();
8     print('Hello, $name , your age is $age');
9 }
10
```

Below the code editor, a terminal window is open, showing the execution of the program. The prompt is 'ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1 <master>'. The user enters '\$ dart code.dart', and the program outputs 'Enter your name: Ayman', 'Enter your age: 30', and 'Hello, Ayman , your age is 30'.

3. Explain the difference between var and dynamic with a practical example, then print a message to show the difference.

```

Day-1 > code.dart > ...
1 // 3. Explain the difference between var and dynamic with a practical example, then print a
2 // message to show the difference.
3
Run | Debug
4 void main() {
5     var someValue = "Ayman" ; // String    The value of the local variable 'someValue' isn't use
6     dynamic anotherValue = "Ayman" ; // String too    The value of the local variable 'another
7
8
9     someValue = 4 ; // ERROR , Var is dynamics for the FIRST assign ONLY    A value of type '3
10    anotherValue = 4 ; // CORRECT , dynamic remains Dynamics to assign to different types , not
11
12    print("""
13    Var is dynamics for the FIRST assign ONLY ,
14    dynamic remains Dynamics to assign to different types , not only once
15    """);
16 }
17

```

4. Explain the similarities and differences between final and const with a practical example, then print a message to show the result.

```

File Edit Selection View ...
code.dart M x  Answers.dart 2, M
Day-1 > code.dart > main
1 //4. Explain the similarities and differences between final and const with a practical example, then
2 // print a message to show the result.
Run | Debug
3 void main() {
4     // const DateTime constTime = new DateTime.now(); // ERROR
5     // final DateTime finalTime = new DateTime.now(); // CORRECT
6
7     // print(constTime);
8     // print(finalTime);
9     print("""
10    const → value remain constant , assignmet happen at CompileTime (while you are coding),this is why it raise error with new DateTime.now();
11    It keeps assigning new Value ( not constant ) all time while you are coding
12
13    final → value remain constant , assignmet happen at Runtime (when you click RUN), this is why it accept new DateTime.now(); without issue
14    """);
15 }
16

```

5. Define variables of the following types: List, Map, Set.  
 Then perform operations on them such as:  
 Finding the length, printing the first element and other operations.  
 And print the results.

```

File Edit Selection View ... NTI-FLUTTER
code.dart M x Answers.dart 2, M
Day-1 > code.dart > main
Run | Debug
1 void main() {
2   List listOfThings = ["ayman", 14.5, 34, false];
3
4   Map mapOfThings = {"name": "ayman", true: "yes", "height": 143.3};
5
6   Set setOfThings = {5, 3, 2}; //
7
8   print("""
9     Here are some list facts : list length = ${listOfThings.length} , First Item : ${listOfThings.first} ,
10    Some set facts : First item : ${setOfThings.first} , last Item : ${setOfThings.last} ,
11    For the map : Keys : ${mapOfThings.keys} ,Values : ${mapOfThings.values}
12    """);
13 }
14
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENT HIGHLIGHTER
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1 <master>
$ dart code.dart
Here are some list facts : list length = 4 , First Item : ayman ,
Some set facts : First item : 5 , last Item : 2 ,
For the map : Keys : (name, true, height) ,Values : (ayman, yes, 143.3)
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1 <master>
$

```

6. Explain the difference between: List, Map, Set And also the difference between Map and JSON , Provide an example to clarify the idea.

```

code.dart 3, M x Answers.dart 2, M
Day-1 > code.dart > main
Run | Debug
1 void main() {
2   List listOfThings = ["ayman", 14.5, 34, false]; The value of the local variable 'listOfThings' isn't used. Try removing the
3   Map mapOfThings = {"name": "ayman", true: "yes", "height": 143.3}; The value of the local variable 'mapOfThings' isn't used.
4   Set setOfThings = {5, 3, 2}; The value of the local variable 'setOfThings' isn't used. Try removing the variable or using it
5
6   print("""
7     1- All of them are Collection of Items/Data
8     2- List , collection of things starting with index 0
9     3- set is same as List ,with no duplicate items
10    4- Map adds more feature by allowing keys to be ( in addition to numbers ) any other values
11
12    5- Map and Json look similar , But JSON ALWAYS need string keys , while Map can have any key even boolean as above
13    """);
14 }
15

```

7. Print the even numbers from 1 to 60 in two different ways using: for loop - while loop

The screenshot shows an IDE with two tabs: `code.dart` and `Answers.dart`. The `code.dart` tab is active and contains the following Dart code:

```
Day-1 > code.dart > main
Run | Debug
1 void main() {
2   for (int i = 0; i < 61; i++) {
3     if (i.isEven) {
4       print(i);
5     }
6   }
7   //
8   int ii = 0;
9   while (ii < 61) {
10    if (ii.isEven) {
11      print(ii);
12    }
13    ii++;
14  }
```

The `TERMINAL` panel at the bottom shows the output of the program, which is a list of even numbers from 38 to 60:

```
38
40
42
44
46
48
50
52
54
56
58
60
```

8. Write a program to calculate BMI (Body Mass Index) using switch case.

The image shows a screenshot of an IDE with two tabs: `code.dart` and `Answers.dart`. The `code.dart` tab is active, showing a Dart program for calculating BMI. The code is as follows:

```
1 void main() {  
2   double weight = 70.0;  
3   double height = 1.75;  
4   double bmi = weight / (height * height)  
5   switch (bmi) {  
6     case < 18.5:  
7       print('Underweight');  
8       break;  
9     case ≥ 18.5 && < 25:  
10      print('Normal weight');  
11      break;  
12     case ≥ 25 && < 30:  
13      print('Overweight');  
14      break;  
15     default:  
16      print('Unknown');  
17   }
```

The IDE interface includes a sidebar on the left with a file explorer showing `5, M` and `M`. At the bottom, there are tabs for `PROBLEMS` (5), `OUTPUT`, `DEBUG CONSOLE`, and `TERMINAL`. The `TERMINAL` tab is active, showing the command `$ dart code.dart` and the output `Normal weight`.

9. Write a function to calculate BMI using if conditions, then print the result when calling the function.

The screenshot shows an IDE with two tabs: `code.dart` and `Answers.dart`. The `code.dart` tab is active, displaying the following Dart code:

```
Day-1 > code.dart > calculateWithIf
Run | Debug
1 void main() {
2   calculateWithIf();
3 }
4
5 void calculateWithIf() {
6   double weight = 70.0;
7   double height = 1.75;
8   double bmi = weight / (height * height);
9   if (bmi < 18.5) {
10    print('underweight');
11  } else if (bmi ≥ 18.5 && bmi < 25) {
12    print('normal');
13  } else if (bmi ≥ 25 && bmi < 30) {
14    print('overweight');
15  } else {
16    print('over');
17  }
```

The bottom panel shows the **TERMINAL** output:

```
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1 <master>
$ dart code.dart
normal
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1 <master>
$
```

10. Explain the difference between while and do-while with an example that shows how each one works.

The screenshot shows an IDE window titled "NTI-FLUTTER". The editor has two tabs: "code.dart 1, M" and "Answers.dart 5, M". The "code.dart" tab is active, showing the following Dart code:

```
1 void main() {  
2   bool impossible = false;  
3  
4   while (impossible) {  
5     print("I will never show");  
6   }  
7  
8   do {  
9     print("Do While , makes the impossible,, possible");  
10  } while (impossible);  
11 }  
12
```

The IDE interface includes a "Run | Debug" button, a "PROBLEMS" panel with 6 items, and a "TERMINAL" panel. The terminal shows the execution of the code:

```
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1 <master>  
$ dart code.dart  
Do While , makes the impossible,, possible  
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1 <master>  
$
```

11. Write a program to generate and print the multiplication table of the number 5.



```
on View ... < NTI-FLUTTER 86
... code.dart M x Answers.dart 6
Day-1 > code.dart > main
1 // 11. Write a program to generate and print the multiplication table of the number 5.
  Run | Debug
2 void main() {
3   int fixedFive = 5;
4   for (var i = 0; i < 11; i++) {
5     print("5 * ${i} = ${i * fixedFive}");
6   }
7 }
8

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENT HIGHLIGHTER zsh - Day-1
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1 <main>
$ dart code.dart
5 * 0 = 0
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
ayman@HP-8470p-3c4010a4 ~/Downloads/NTI-FLUTTER/Day-1 <main>
```