

# Tugas Praktikum

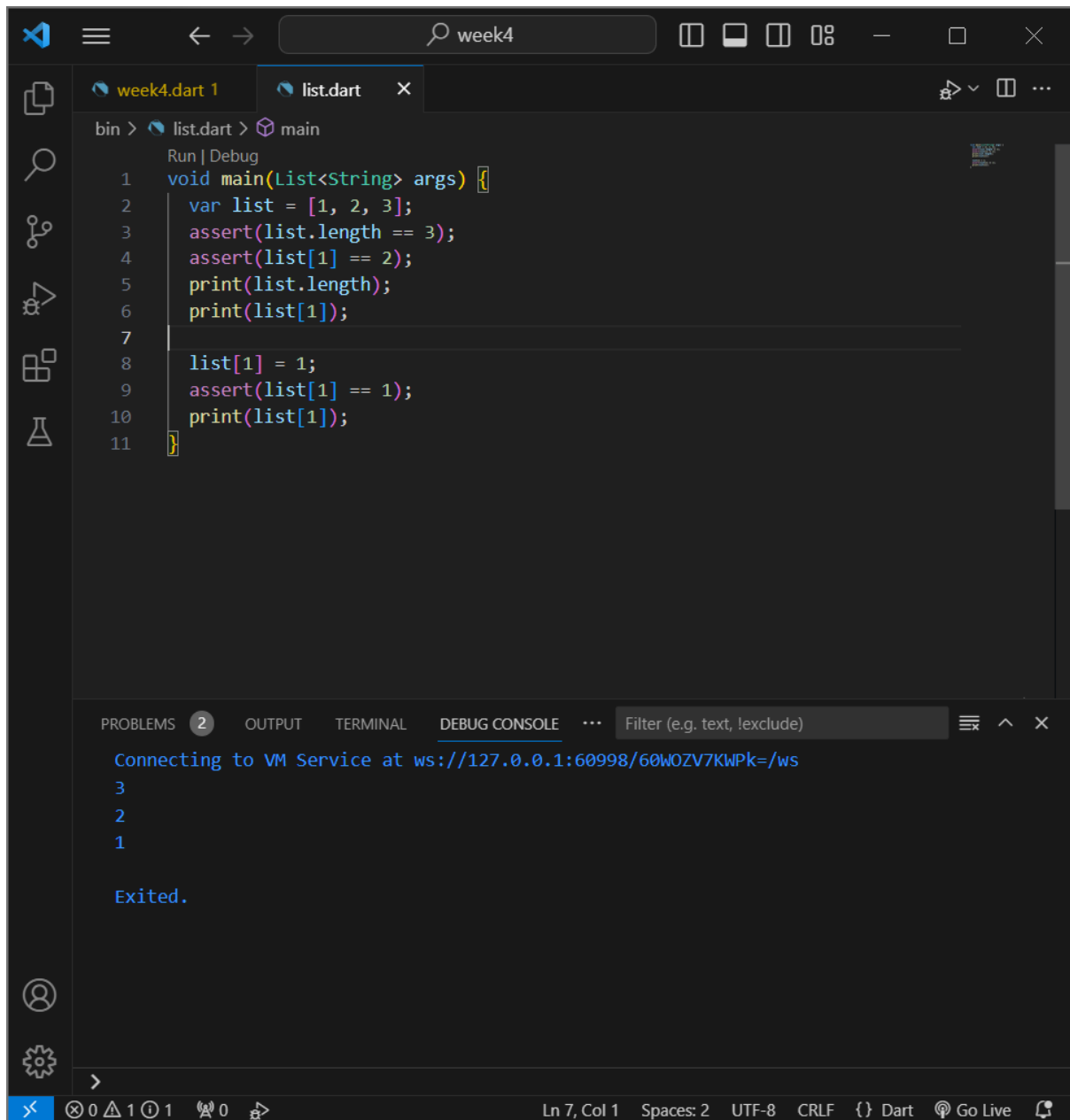
## 1. Practicum 1: Data List Type Experiment

Step 1:

Type or copy the following program code into void main().

Step 2:

Please try executing (Run) the code in step 1. What happened? Explain!



The screenshot shows an IDE window with two tabs: 'week4.dart 1' and 'list.dart'. The 'list.dart' tab is active, displaying the following Dart code:

```
bin > list.dart > main
Run | Debug
1 void main(List<String> args) {
2   var list = [1, 2, 3];
3   assert(list.length == 3);
4   assert(list[1] == 2);
5   print(list.length);
6   print(list[1]);
7
8   list[1] = 1;
9   assert(list[1] == 1);
10  print(list[1]);
11 }
```

The bottom panel shows the 'DEBUG CONSOLE' with the following output:

```
Connecting to VM Service at ws://127.0.0.1:60998/60W0ZV7KWPk=/ws
3
2
1

Exited.
```

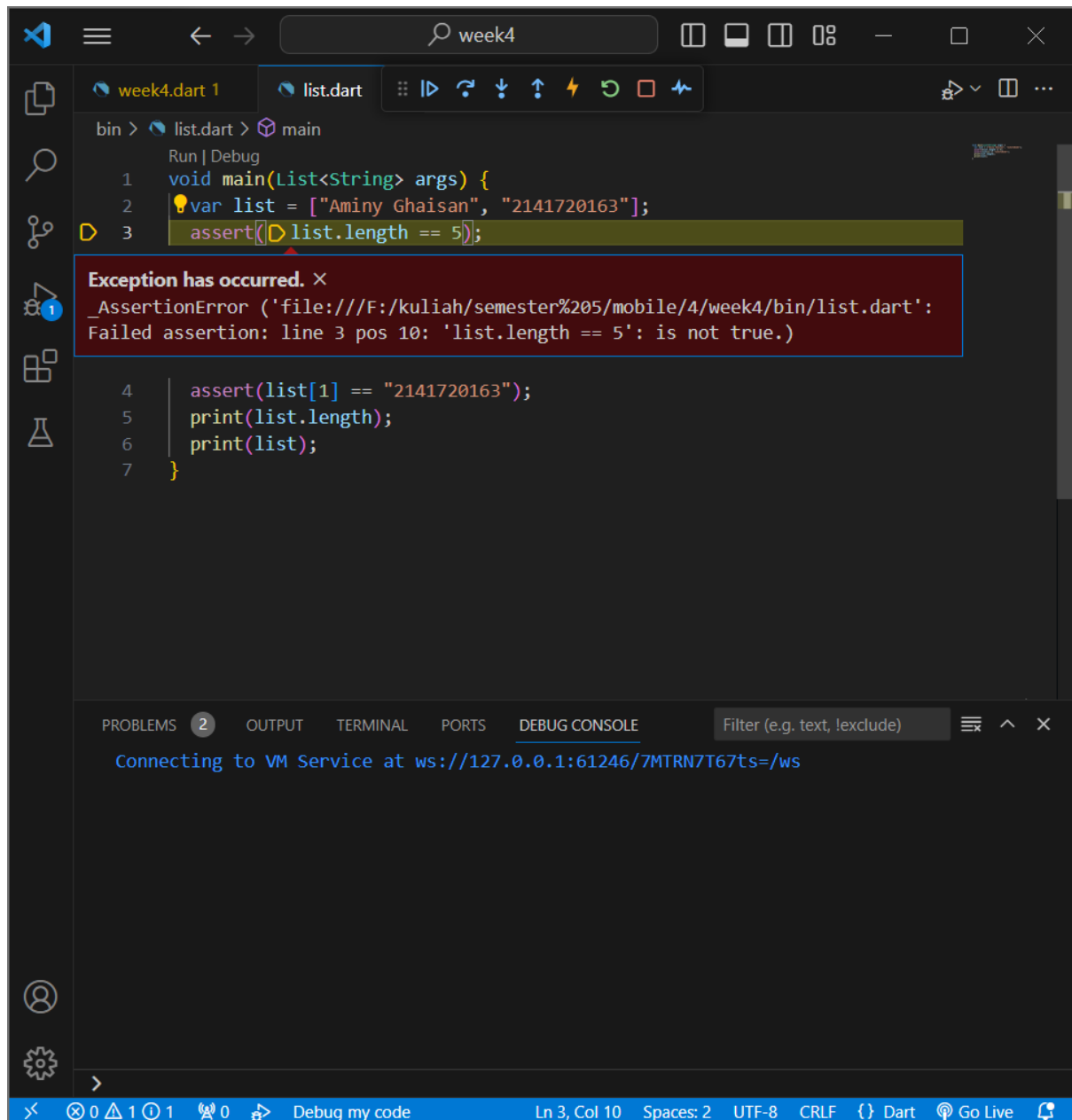
The status bar at the bottom indicates the cursor is at 'Ln 7, Col 1', with 'Spaces: 2', 'UTF-8', 'CRLF', and 'Dart' encoding.

This dart code creates an array (list), performs some assertions to ensure specific conditions are met, and prints the length and elements of the list to the console. It demonstrates basic array manipulation and debugging techniques in Dart.

Step 3:

Change the code in step 1 to become a final variable that has index = 5 with default value = null. Fill in your name and NIM in the 1st and 2nd index elements. Then print and capture the results.

What happened? If an error occurs, please correct it.



The screenshot shows an IDE window with a file named `list.dart` open. The code is as follows:

```
1 void main(List<String> args) {  
2   var list = ["Aminy Ghaisan", "2141720163"];  
3   assert(list.length == 5);  
4   assert(list[1] == "2141720163");  
5   print(list.length);  
6   print(list);  
7 }
```

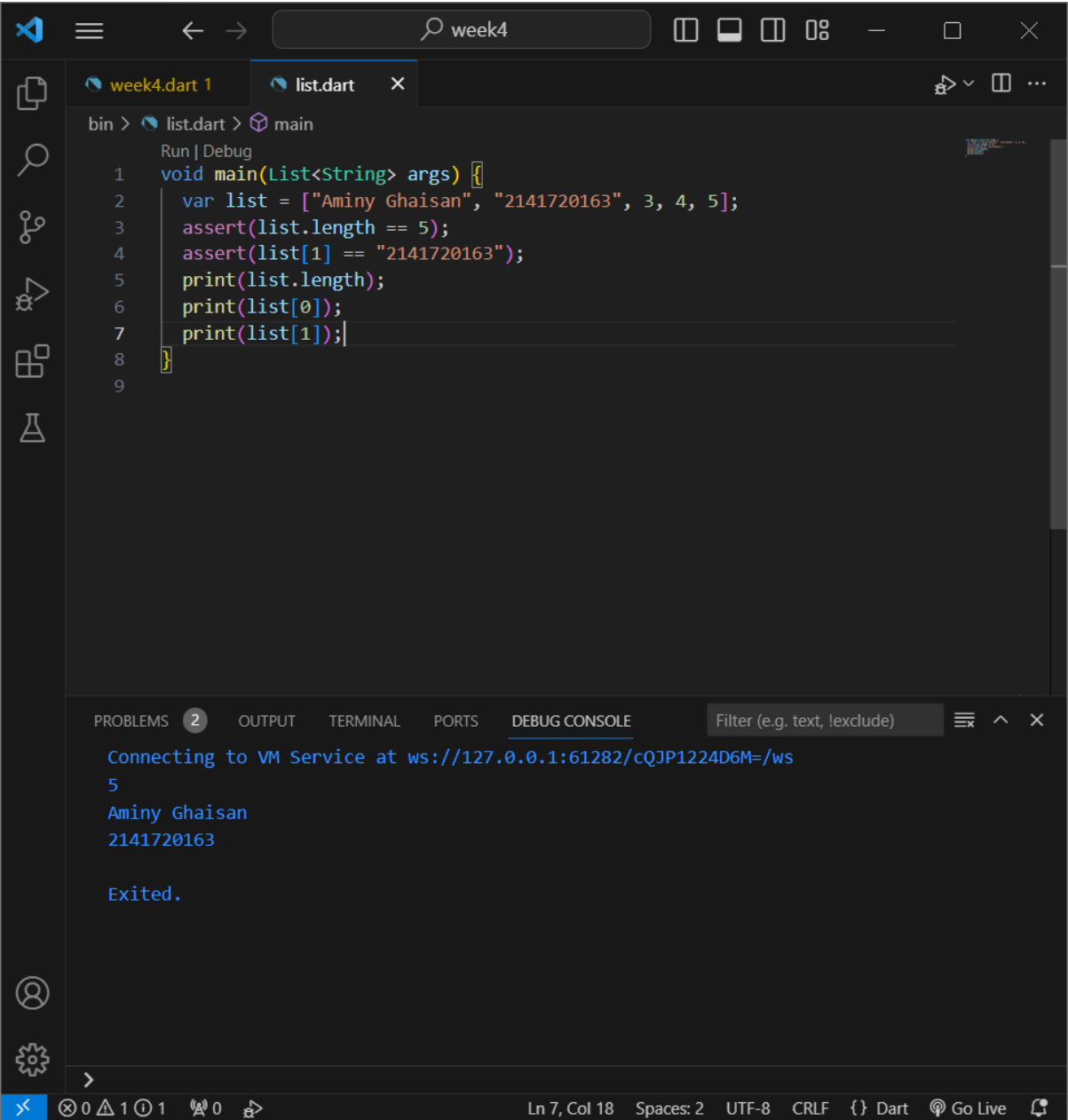
An exception has occurred, as indicated by the red error message box:

```
Exception has occurred. ×  
_AssertionError ('file:///F:/kuliah/semester%205/mobile/4/week4/bin/list.dart':  
Failed assertion: line 3 pos 10: 'list.length == 5': is not true.)
```

The bottom of the IDE shows the `DEBUG CONSOLE` tab with the message: `Connecting to VM Service at ws://127.0.0.1:61246/7MTRN7T67ts=/ws`. The status bar at the bottom indicates the file is at `Ln 3, Col 10` with `Spaces: 2`, `UTF-8` encoding, `CRLF` line endings, and the language is `Dart`.

The code above experiences an error because the list declaration is `'List<int?>'` which means it only contains integer or null type values. However I filled the elements with string data type.

Code that has been corrected:



The screenshot shows an IDE window with two tabs: `week4.dart 1` and `list.dart`. The `list.dart` tab is active, displaying the following Dart code:

```
bin > list.dart > main
Run | Debug
1 void main(List<String> args) {
2   var list = ["Aminy Ghaisan", "2141720163", 3, 4, 5];
3   assert(list.length == 5);
4   assert(list[1] == "2141720163");
5   print(list.length);
6   print(list[0]);
7   print(list[1]);
8 }
9
```

Below the code editor, the `DEBUG CONSOLE` tab is selected, showing the execution output:

```
Connecting to VM Service at ws://127.0.0.1:61282/cQJP1224D6M=/ws
5
Aminy Ghaisan
2141720163

Exited.
```

The status bar at the bottom indicates the current position is `Ln 7, Col 18`, with `Spaces: 2`, `UTF-8` encoding, `CRLF` line endings, and the `Dart` language.

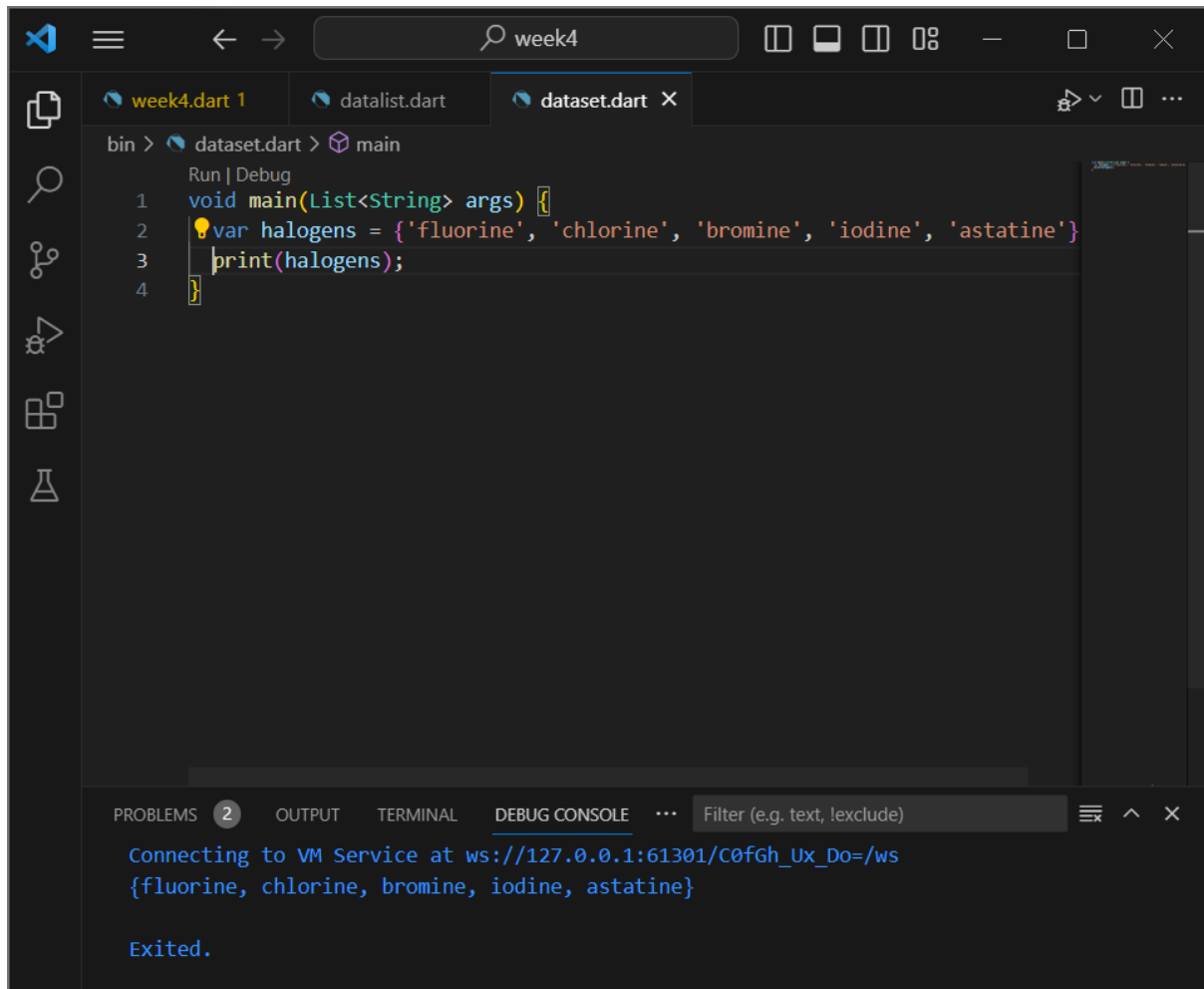
## 2. Practicum 2: Data Set Type Experiment

Step 1:

Type or copy the following program code into the main() function.

Step 2:

Please try executing (Run) the code in step 1. What happened? Explain! Then fix it if an error occurs.



The screenshot shows an IDE window with three tabs: `week4.dart 1`, `datalist.dart`, and `dataset.dart`. The `dataset.dart` tab is active, showing the following code:

```
bin > dataset.dart > main
Run | Debug
1 void main(List<String> args) {
2   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'}
3   print(halogens);
4 }
```

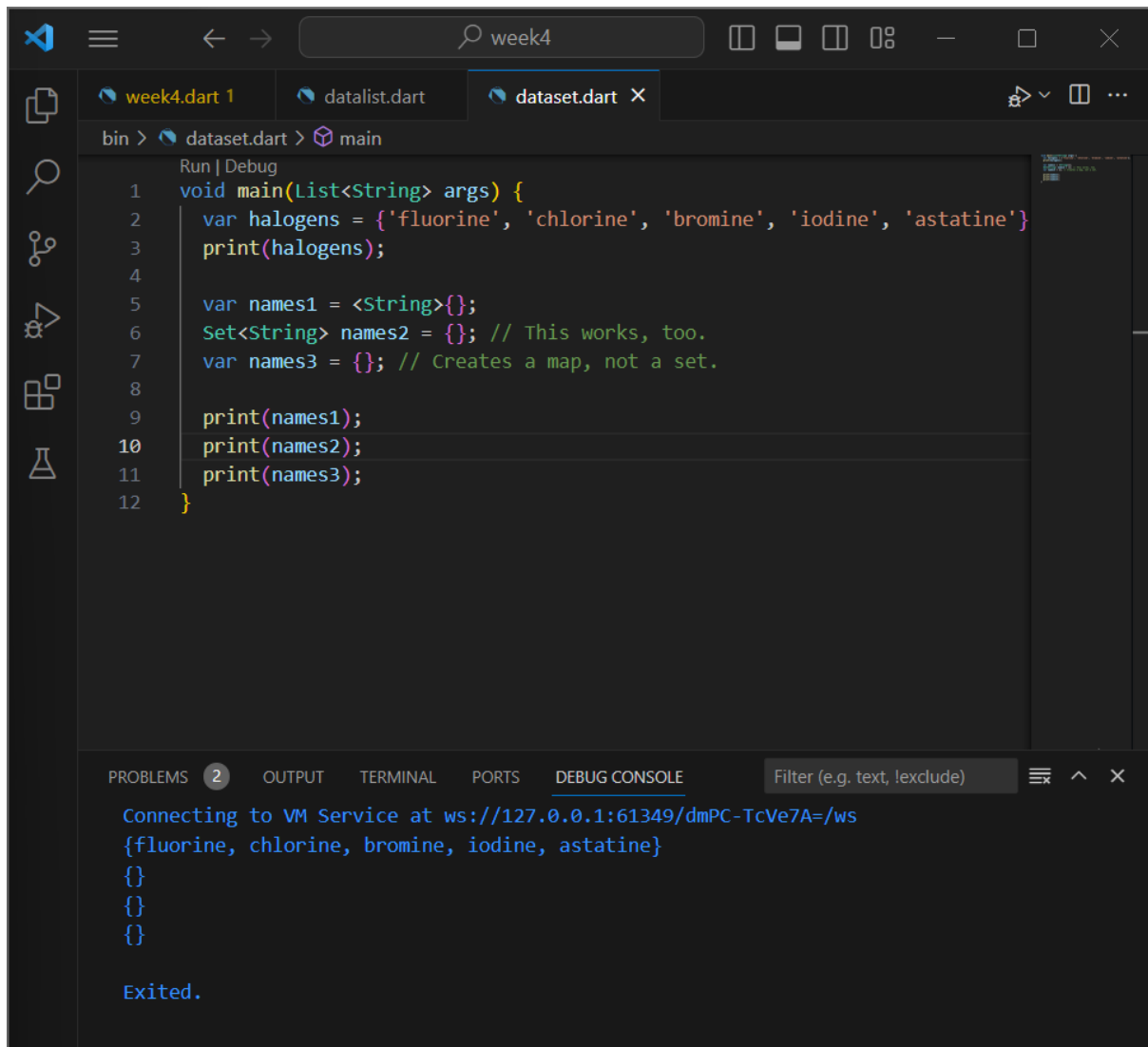
The bottom panel shows the `DEBUG CONSOLE` with the following output:

```
Connecting to VM Service at ws://127.0.0.1:61301/C0fGh_Ux_Do=/ws
{fluorine, chlorine, bromine, iodine, astatine}
Exited.
```

The code above does not produce an error. The code above is an example of using a "Set" in Dart, which is a collection of unique elements. The code contains the names of halogen elements, and then the contents of the set are printed to the screen with the names of the halogen elements.

Step 3:

Add the following program code, then try executing (Run) your code.



The screenshot shows an IDE window with three tabs: `week4.dart 1`, `datalist.dart`, and `dataset.dart X`. The active file is `dataset.dart`, which contains the following Dart code:

```
1 void main(List<String> args) {  
2   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'}  
3   print(halogens);  
4  
5   var names1 = <String>{};  
6   Set<String> names2 = {}; // This works, too.  
7   var names3 = {}; // Creates a map, not a set.  
8  
9   print(names1);  
10  print(names2);  
11  print(names3);  
12 }
```

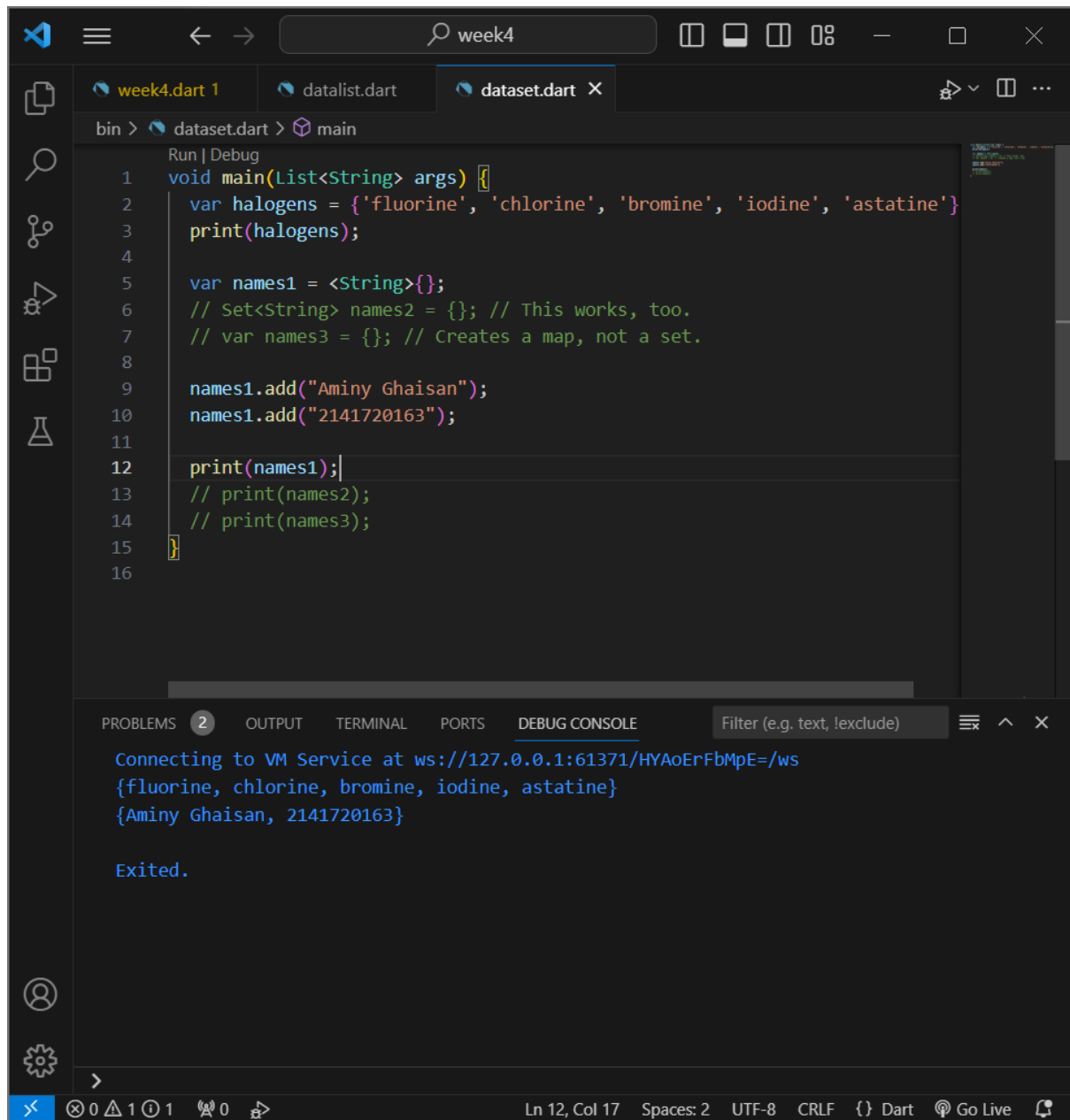
The IDE's bottom panel shows the `DEBUG CONSOLE` with the following output:

```
Connecting to VM Service at ws://127.0.0.1:61349/dmPC-TcVe7A=/ws  
{fluorine, chlorine, bromine, iodine, astatine}  
{}  
{}  
{}  
  
Exited.
```

What happens? If there is an error, please fix it but still use all three variables.

- Actually, there is no error, but 'names3' is actually an empty map, not a set. If you want to create a set, make sure to specify the data type as seen in 'names1' and 'names2'.

Add your name and NIM elements to the two Set variables with two different functions, namely `.add()` and `.addAll()`. The Map variable is deleted, we will try it later in the next practicum.



The screenshot shows an IDE window with three tabs: `week4.dart 1`, `datalist.dart`, and `dataset.dart`. The `dataset.dart` tab is active, showing the following Dart code:

```
bin > dataset.dart > main
Run | Debug
1 void main(List<String> args) {}
2   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'}
3   print(halogens);
4
5   var names1 = <String>{};
6   // Set<String> names2 = {}; // This works, too.
7   // var names3 = {}; // Creates a map, not a set.
8
9   names1.add("Aminy Ghaisan");
10  names1.add("2141720163");
11
12  print(names1);
13  // print(names2);
14  // print(names3);
15
16
```

The bottom panel shows the `DEBUG CONSOLE` with the following output:

```
Connecting to VM Service at ws://127.0.0.1:61371/HYAOErFbMpE=/ws
{fluorine, chlorine, bromine, iodine, astatine}
{Aminy Ghaisan, 2141720163}

Exited.
```

The status bar at the bottom indicates the cursor is at `Ln 12, Col 17`, with `Spaces: 2`, `UTF-8`, `CRLF`, `{ }`, `Dart`, and `Go Live` options.

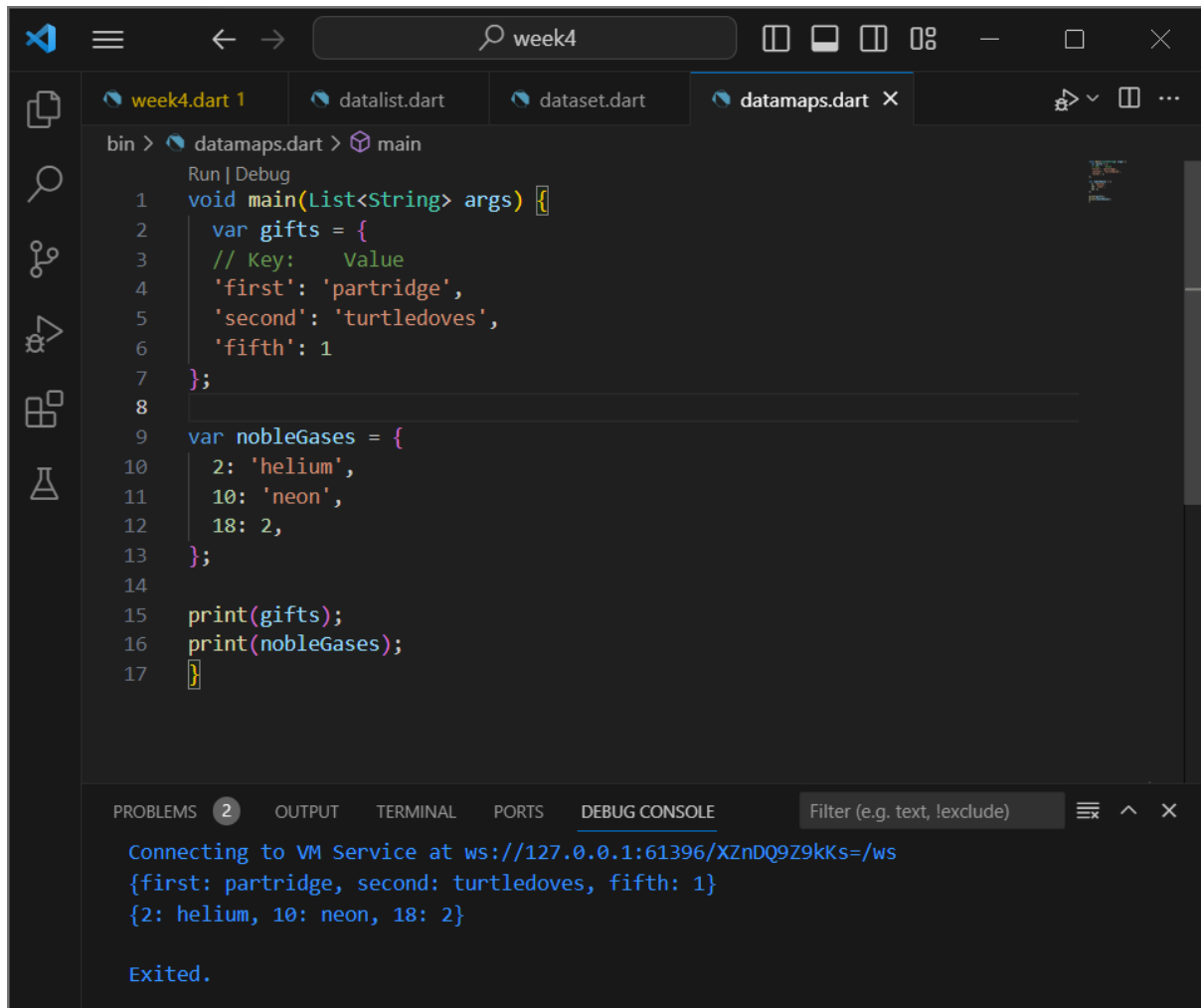
### 3. Practicum 3: Data Maps Type Experiment

Step 1:

Type or copy the following program code into the main() function.

Step 2:

Please try executing (Run) the code in step 1. What happened? Explain! Then fix it if an error occurs.



The screenshot shows an IDE window with a file explorer on the left and a code editor. The file explorer shows a project named 'week4' with files 'week4.dart 1', 'datalist.dart', 'dataset.dart', and 'datamaps.dart'. The code editor is open to 'datamaps.dart' and shows the following Dart code:

```
bin > datamaps.dart > main
Run | Debug
1 void main(List<String> args) {
2   var gifts = {
3     // Key: Value
4     'first': 'partridge',
5     'second': 'turtledoves',
6     'fifth': 1
7   };
8
9   var nobleGases = {
10    2: 'helium',
11    10: 'neon',
12    18: 2,
13  };
14
15  print(gifts);
16  print(nobleGases);
17 }
```

The bottom panel shows the 'DEBUG CONSOLE' with the following output:

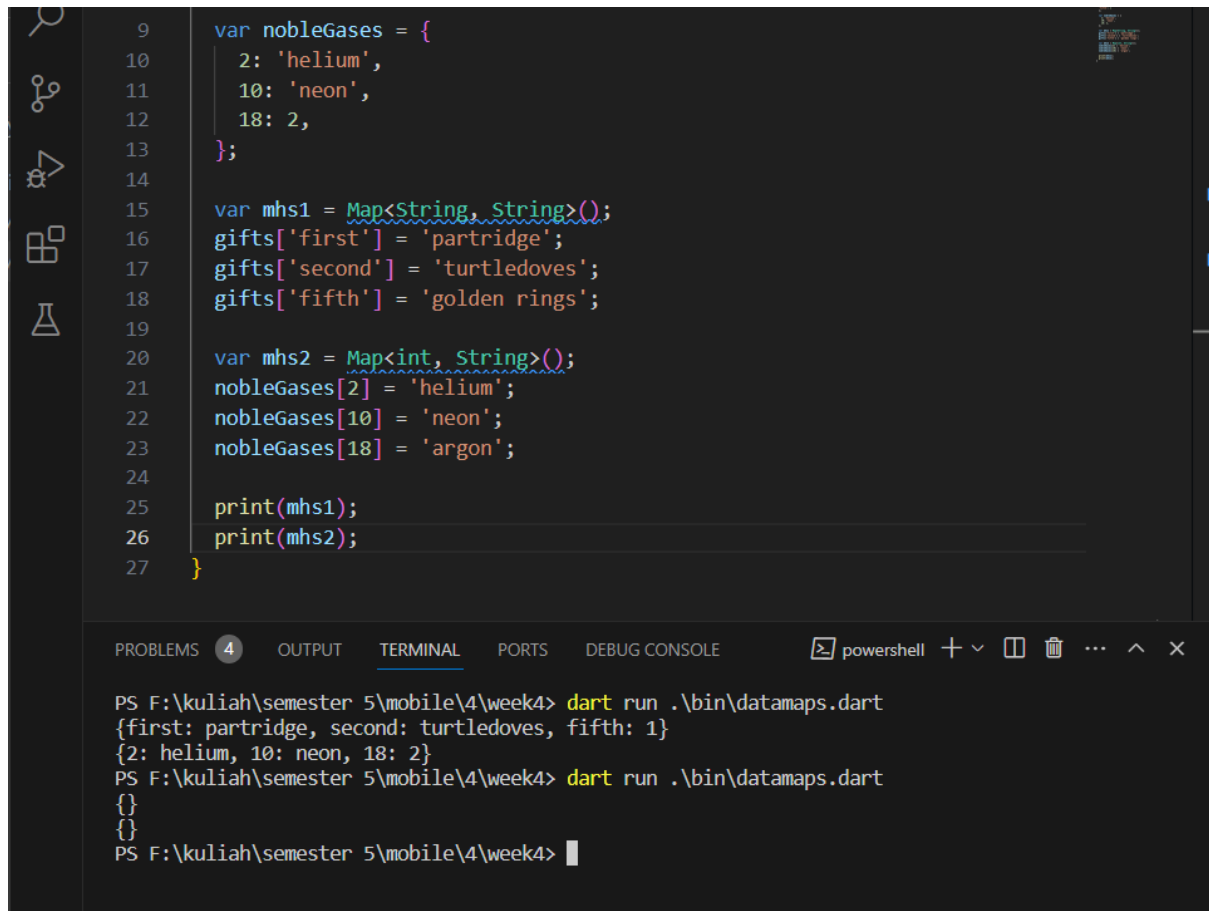
```
Connecting to VM Service at ws://127.0.0.1:61396/XZnDQ9Z9kKs=/ws
{first: partridge, second: turtledoves, fifth: 1}
{2: helium, 10: neon, 18: 2}

Exited.
```

No errors occurred, and produced output as instructed by print gifts and nobleGases.

Step 3:

Add the following program code, then try executing (Run) your code. What happened? If an error occurs, please correct it.



```
9  var nobleGases = {
10    2: 'helium',
11    10: 'neon',
12    18: 2,
13  };
14
15  var mhs1 = Map<String, String>();
16  gifts['first'] = 'partridge';
17  gifts['second'] = 'turtledoves';
18  gifts['fifth'] = 'golden rings';
19
20  var mhs2 = Map<int, String>();
21  nobleGases[2] = 'helium';
22  nobleGases[10] = 'neon';
23  nobleGases[18] = 'argon';
24
25  print(mhs1);
26  print(mhs2);
27 }
```

PROBLEMS 4 OUTPUT TERMINAL PORTS DEBUG CONSOLE powershell + - [ ] [X] ... ^ X

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datamaps.dart
{first: partridge, second: turtledoves, fifth: 1}
{2: helium, 10: neon, 18: 2}
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datamaps.dart
{}
{}
PS F:\kuliah\semester 5\mobile\4\week4>
```

Actually it didn't produce an error, but because I created two new folders, namely 'mhs1' and 'mhs2', the contents of which were empty. This is different from the two initial folders 'gifts' and 'nobleGases' which already contain contents and have key-value pairs that match the specified data type.



```
bin > datamaps.dart > main
1 void main(List<String> args) {}
2 //Key: Value
3 var gifts = {
4   'first': 'partridge',
5   'second': 'turtledoves',
6   'fifth': 1,
7   'name': 'Aminy Ghaisan',
8   'nim': 2141720163
9 };
10 var nobleGases = {
11   2: 'helium',
12   10: 'neon',
13   18: 2,
14   0: 'Aminy Ghaisan',
15   1: 2141720163
16 };
17
18 var mhs1 = Map<String, String>();
19 mhs1['first'] = 'partridge';
20 mhs1['second'] = 'turtledoves';
21 mhs1['fifth'] = 'golden rings';
22 mhs1['name'] = 'Aminy Ghaisan';
23 mhs1['nim'] = '2141720163';
24
25 var mhs2 = Map<int, String>();
26 mhs2[2] = 'helium';
27 mhs2[10] = 'neon';
28 mhs2[18] = 'argon';
29 mhs2[0] = 'Aminy Ghaisan';
30 mhs2[1] = '2141720163';
31
32 print(mhs1);
33 print(mhs2);
34 print(gifts);
35 print(nobleGases);
36
```

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datamaps.dart
{first: partridge, second: turtledoves, fifth: golden rings, name: Aminy Ghaisan, nim: 2141720163}
{2: helium, 10: neon, 18: argon, 0: Aminy Ghaisan, 1: 2141720163}
{first: partridge, second: turtledoves, fifth: 1, name: Aminy Ghaisan, nim: 2141720163}
{2: helium, 10: neon, 18: 2, 0: Aminy Ghaisan, 1: 2141720163}
PS F:\kuliah\semester 5\mobile\4\week4>
```

#### 4. Practicum 4: List Data Type Experiments: Spread and Control-flow Operators

Step 1:

Type or copy the following program code into the main() function.

Step 2:

Please try executing (Run) the code in step 1. What happened? Explain! Then fix it if an error occurs.

The image shows a VS Code editor window with a Dart file named `datalist_spread_control_flow.dart`. The code defines a `main` function that takes a list of arguments and prints the contents of two lists, `list1` and `list2`, and the length of `list2`.

```
bin > datalist_spread_control_flow.dart > main
Run | Debug
1 void main(List<String> args) {
2   var list = [1, 2, 3];
3   var list2 = [0, ...list];
4   print(list1);
5   print(list2);
6   print(list2.length);
7 }
```

The terminal output shows the execution of the program, displaying the contents of the lists and an error message: `Error: Undefined name 'list1'.` The error message is highlighted in red.

```
{2: helium, 10: neon, 18: argon, 0: Aminy Ghaisan, 1: 2141720163}
{first: partridge, second: turtledoves, fifth: 1, name: Aminy Ghaisan, nim: 2141720163}
{2: helium, 10: neon, 18: 2, 0: Aminy Ghaisan, 1: 2141720163}
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist_spread_control_flow.dart
bin/datalist_spread_control_flow.dart:4:9: Error: Undefined name 'list1'.
  print(list1);
      ^^^^^
```

The error message is highlighted in red.

Experiencing an error because previously there was no 'list1' variable declaration. Therefore, there is a code improvement by replacing 'list1' with 'list' and producing the output as below

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

```
bin > datalist_spread_control_flow.dart > main
Run | Debug
1 void main(List<String> args) {
2   var list1 = [1, 2, 3];
3   var list2 = [0, ...list1];
4   print(list1);
5   print(list2);
6   print(list2.length);
7 }
```

The terminal at the bottom shows the command `dart run .\bin\datalist_spread_control_flow.dart` being executed. The output displays a map, a list, and the length of the list, but it is preceded by an error message:

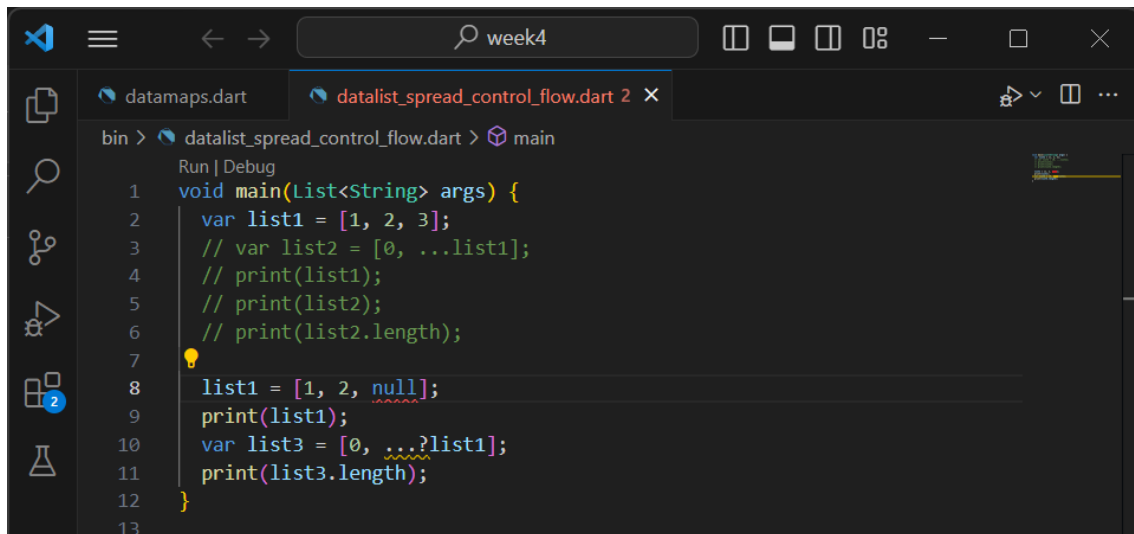
```
{first: partridge, second: turtledoves, fifth: 1, name: Aminy Ghaisan, nim: 2141720163}
{2: helium, 10: neon, 18: 2, 0: Aminy Ghaisan, 1: 2141720163}
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist_spread_control_flow.dart
bin\datalist_spread_control_flow.dart:4:9: Error: Undefined name 'list1'.
  print(list1);
        ^^^^^
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist_spread_control_flow.dart
[1, 2, 3]
[0, 1, 2, 3]
4
PS F:\kuliah\semester 5\mobile\4\week4>
```

The status bar at the bottom indicates the cursor is at line 4, column 16, with 2 spaces, UTF-8 encoding, CRLF line endings, and the Dart language.

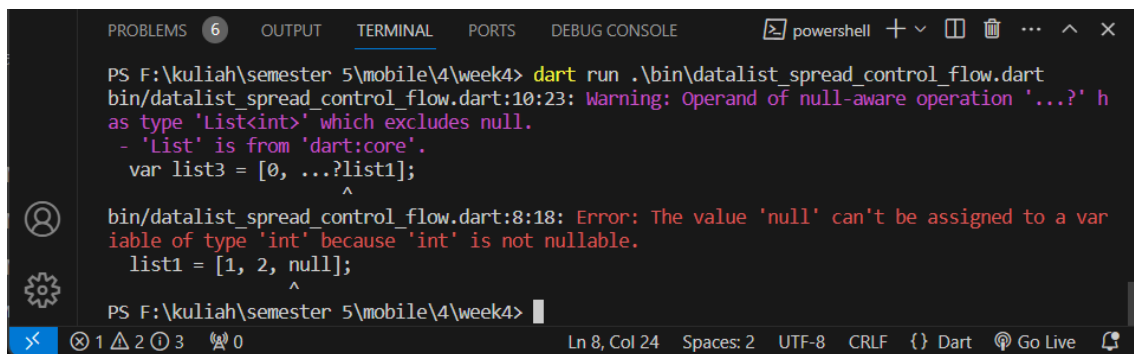
Step 3:

Add the following program code, then try executing (Run) your code.

What happened? If an error occurs, please correct it. Add a list variable containing your NIM using Spread Operators. Document the results and make a report!

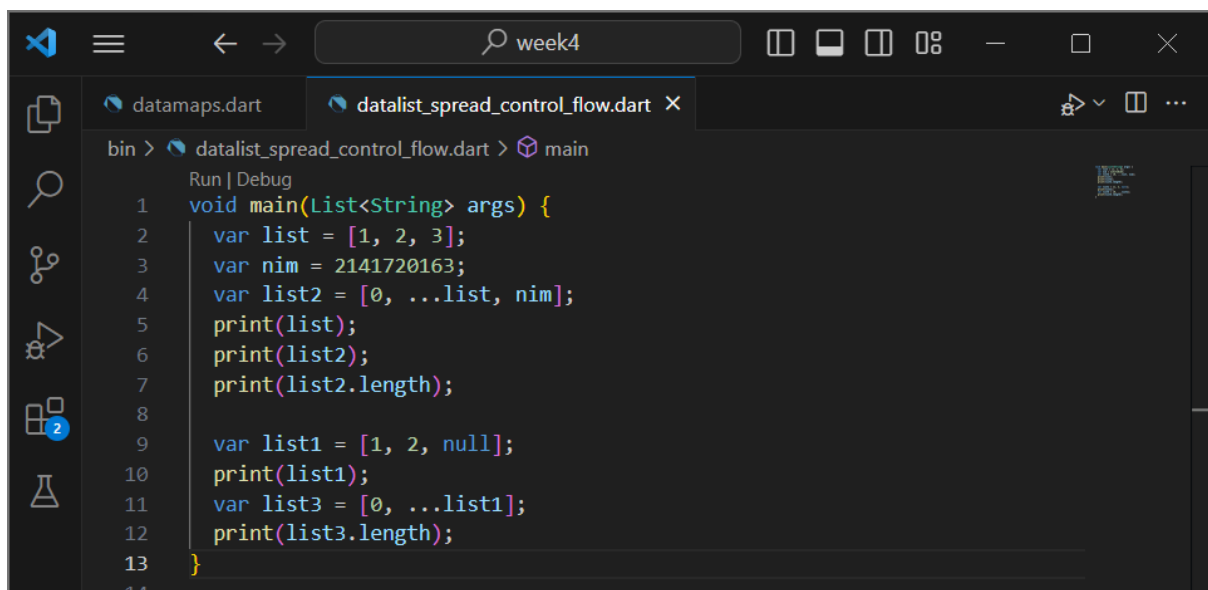


```
bin > datalist_spread_control_flow.dart > main
Run | Debug
1 void main(List<String> args) {
2   var list1 = [1, 2, 3];
3   // var list2 = [0, ...list1];
4   // print(list1);
5   // print(list2);
6   // print(list2.length);
7
8   list1 = [1, 2, null];
9   print(list1);
10  var list3 = [0, ...?list1];
11  print(list3.length);
12 }
13
```

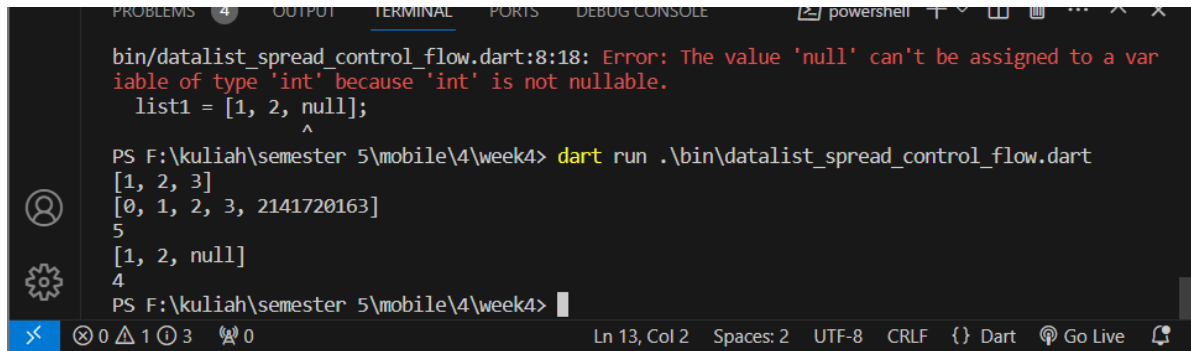


```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist_spread_control_flow.dart
bin/datalist_spread_control_flow.dart:10:23: Warning: Operand of null-aware operation '...?' has
as type 'List<int>' which excludes null.
- 'List' is from 'dart:core'.
  var list3 = [0, ...?list1];
                    ^
bin/datalist_spread_control_flow.dart:8:18: Error: The value 'null' can't be assigned to a var
iable of type 'int' because 'int' is not nullable.
  list1 = [1, 2, null];
                ^
PS F:\kuliah\semester 5\mobile\4\week4>
```

The code above occurs an error when running, because there is a writing error in the variable where the variable declaration is 'list1' but tries to access it as 'list1' in the 'list3' statement. This causes an error because 'list1' was not declared previously. To improve the code below, add the variables var and nim and produce the output below



```
bin > datalist_spread_control_flow.dart > main
Run | Debug
1 void main(List<String> args) {
2   var list = [1, 2, 3];
3   var nim = 2141720163;
4   var list2 = [0, ...list, nim];
5   print(list);
6   print(list2);
7   print(list2.length);
8
9   var list1 = [1, 2, null];
10  print(list1);
11  var list3 = [0, ...list1];
12  print(list3.length);
13 }
14
```

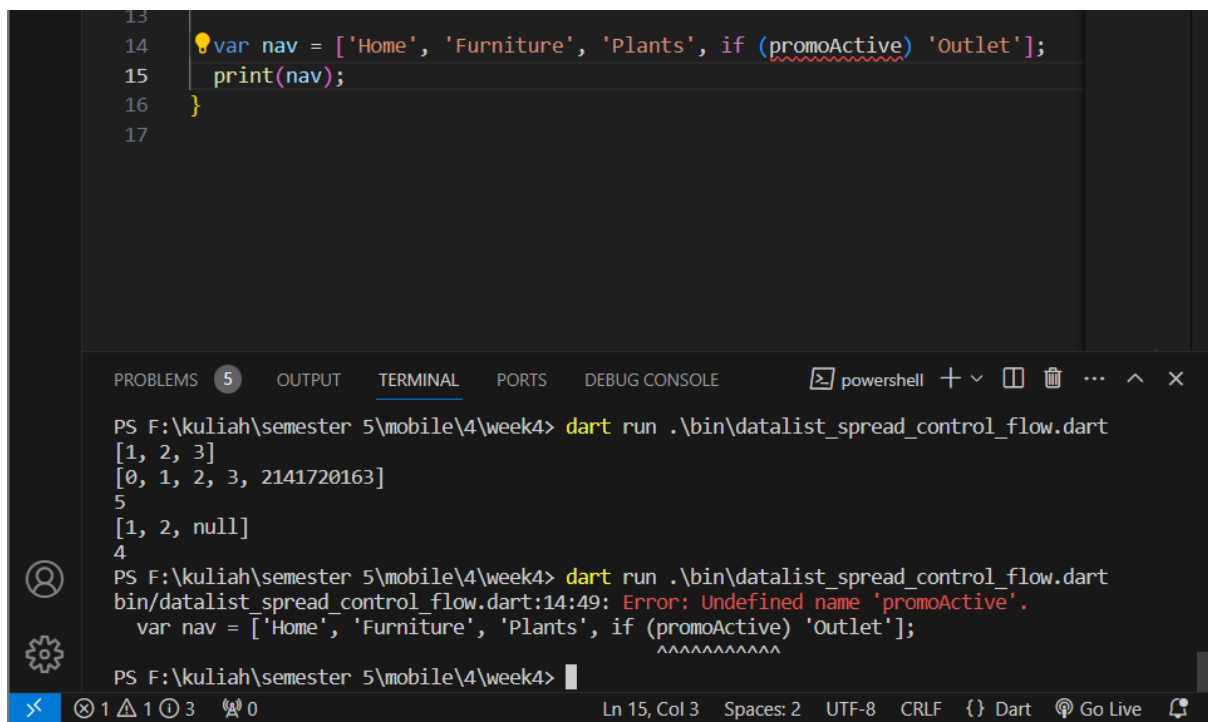


```
bin/datalist_spread_control_flow.dart:8:18: Error: The value 'null' can't be assigned to a variable of type 'int' because 'int' is not nullable.
  list1 = [1, 2, null];
               ^
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist_spread_control_flow.dart
[1, 2, 3]
[0, 1, 2, 3, 2141720163]
5
[1, 2, null]
4
PS F:\kuliah\semester 5\mobile\4\week4>
```

Step 4:

Add the following program code, then try executing (Run) your code.

What happened? If an error occurs, please correct it. Show the result if the promoActive variable is true and false.



```
13
14 var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
15 print(nav);
16 }
17
```

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist_spread_control_flow.dart
[1, 2, 3]
[0, 1, 2, 3, 2141720163]
5
[1, 2, null]
4
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist_spread_control_flow.dart
bin/datalist_spread_control_flow.dart:14:49: Error: Undefined name 'promoActive'.
  var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
                                              ^^^^^^^^^^^^^
PS F:\kuliah\semester 5\mobile\4\week4>
```

An error occurred when running the code above because the proActive variable was not declared or did not have a clear value.

- If the variable is true

```

13
14   var promoActive = true;
15   var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
16   print(nav);
17 }
18

```

bin/datalist\_spread\_control\_flow.dart:14:49: Error: Undefined name 'promoActive'.  
var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];  
~~~~~

PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist\_spread\_control\_flow.dart  
[1, 2, 3]  
[0, 1, 2, 3, 2141720163]  
5  
[1, 2, null]  
4  
[Home, Furniture, Plants, Outlet]  
PS F:\kuliah\semester 5\mobile\4\week4>

- If the variable is false

```

14   var promoActive = false;
15   var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
16   print(nav);
17 }
18

```

[1, 2, null]  
4  
[Home, Furniture, Plants, Outlet]  
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist\_spread\_control\_flow.dart  
[1, 2, 3]  
[0, 1, 2, 3, 2141720163]  
5  
[1, 2, null]  
4  
[Home, Furniture, Plants]  
PS F:\kuliah\semester 5\mobile\4\week4>

Step 5:

Add the following program code, then try executing (Run) your code.

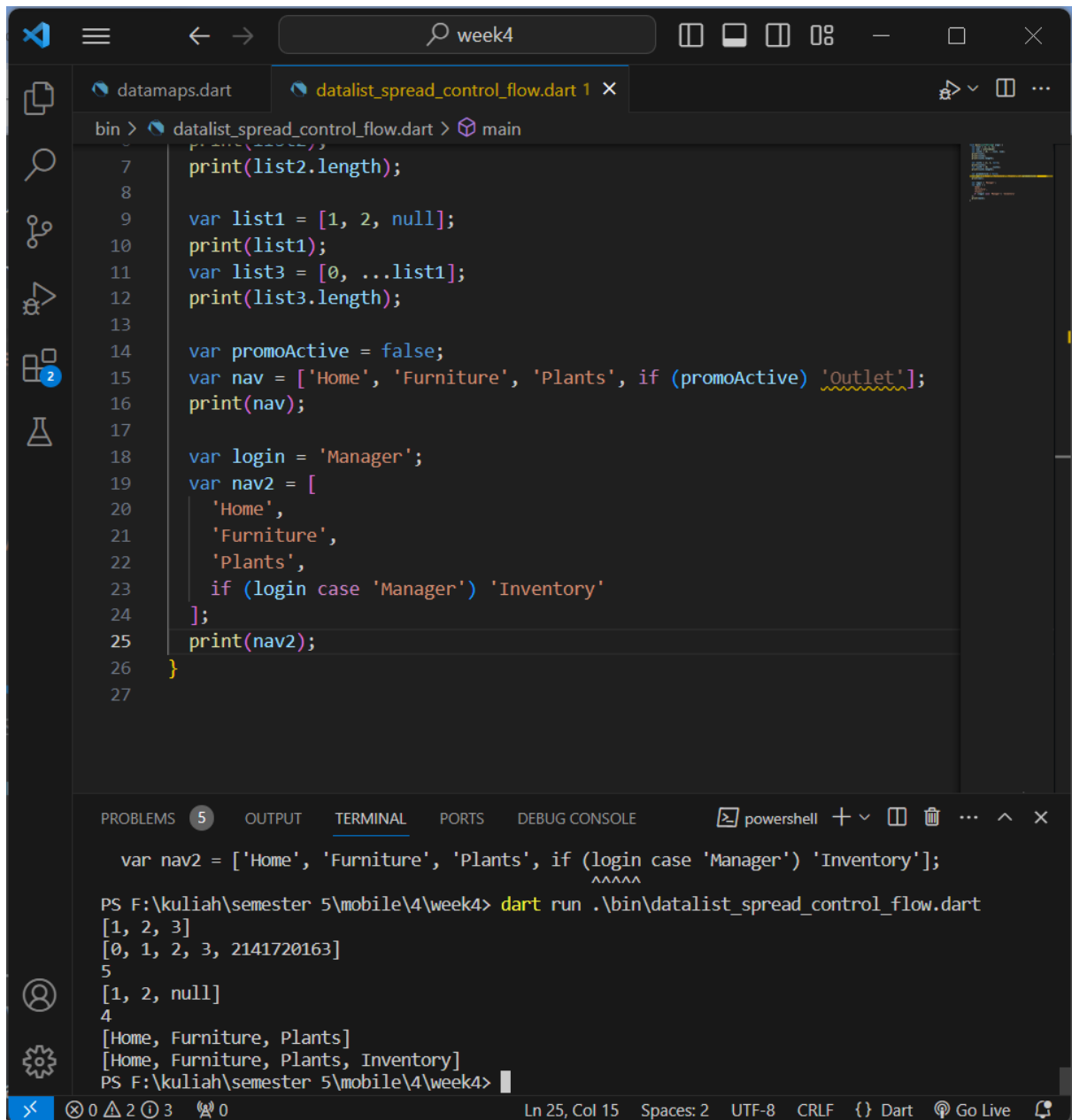
```
bin > datalist_spread_control_flow.dart > main
Run | Debug
1 void main(List<String> args) {
2   var list = [1, 2, 3];
3   var nim = 2141720163;
4   var list2 = [0, ...list, nim];
5   print(list);
6   print(list2);
7   print(list2.length);
8
9   var list1 = [1, 2, null];
10  print(list1);
11  var list3 = [0, ...list1];
12  print(list3.length);
13
14  var promoActive = false;
15  var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
16  print(nav);
17
18  var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager') 'In
19  print(nav2);
20 }
21
```

PROBLEMS 6 OUTPUT TERMINAL PORTS DEBUG CONSOLE powershell + -

```
[1, 2, 3]
[0, 1, 2, 3, 2141720163]
5
[1, 2, null]
4
[Home, Furniture, Plants]
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist_spread_control_flow.dart
bin/datalist_spread_control_flow.dart:18:50: Error: Undefined name 'login'.
    var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager') 'Inventory'];
                                              ^^^^^
PS F:\kuliah\semester 5\mobile\4\week4>
```

Ln 19, Col 3 Spaces: 2 UTF-8 CRLF {} Dart Go Live

An error occurred during the run process because there was no definition of the variable name 'login' previously.



The screenshot shows an IDE with two tabs: `datamaps.dart` and `datalist_spread_control_flow.dart 1`. The active file is `datalist_spread_control_flow.dart`, which contains the following Dart code:

```
bin > datalist_spread_control_flow.dart > main
7   print(list2.length);
8
9   var list1 = [1, 2, null];
10  print(list1);
11  var list3 = [0, ...list1];
12  print(list3.length);
13
14  var promoActive = false;
15  var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
16  print(nav);
17
18  var login = 'Manager';
19  var nav2 = [
20    'Home',
21    'Furniture',
22    'Plants',
23    if (login case 'Manager') 'Inventory'
24  ];
25  print(nav2);
26 }
27
```

The terminal output shows the execution of the code:

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datalist_spread_control_flow.dart
[1, 2, 3]
[0, 1, 2, 3, 2141720163]
5
[1, 2, null]
4
[Home, Furniture, Plants]
[Home, Furniture, Plants, Inventory]
PS F:\kuliah\semester 5\mobile\4\week4>
```

The status bar at the bottom indicates the cursor is at line 25, column 15, with 2 spaces, UTF-8 encoding, CRLF line endings, and the Dart language.

In the displayed output it can be explained that the 'Inventory' element is included in 'nav2' only if the value of the 'login' variable is 'Manager'. if 'login' is not the same as 'Manager', then 'Inventory' will not be included in nav2



Add the following program code, then try executing (Run) your code.

What happened ? If an error occurs, please correct it. Explain the benefits of Collection For and document the results.

- No error occurs because the code above creates two lists: `listOfInts` which contains the numbers `[1, 2, 3]`, and `listOfStrings` which is created with a `for-in` expression that adds a `#` sign in front of each element in `listOfInts`. The result is a `listOfStrings` containing `['#0', '#1', '#2', '#3']`. The `assert` statement checks whether the 1st element in `listOfStrings` is `#1`, which is true. Finally, the results of `listOfStrings` are printed using `print`.
- `Collection` for allows creating declarative data collections based on source collections, producing code that is more concise and easy to read, and makes it easier to transform and manipulate data in the Dart programming language.

## 5. Practicum 5: Data Records Type Experiment

Step 1:

Type or copy the following program code into the main() function.

Step 2:

Please try executing (Run) the code in step 1. What happened? Explain! Then fix it if an error occurs.

The screenshot shows an IDE with three tabs: `datamaps.dart`, `datalist_spread_control_flow.dart 1`, and `datarecords.dart 1`. The `datarecords.dart` tab is active, showing the following Dart code:

```
bin > datarecords.dart > print
1 var record = ('first', a: 2, b: true, 'last');
2 print(record)
```

The IDE's interface includes a sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Testing. The bottom panel contains a **TERMINAL** window with the following output:

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
bin\datarecords.dart:2:14: Error: Expected '{' before this.
print(record)
^...
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
bin\datarecords.dart:2:14: Error: Expected '{' before this.
print(record)
^...
PS F:\kuliah\semester 5\mobile\4\week4>
PS F:\kuliah\semester 5\mobile\4\week4>
```

The status bar at the bottom indicates the cursor is at **Ln 2, Col 14**, with **Spaces: 2**, **UTF-8** encoding, **CRLF** line endings, and the **Dart** language. It also shows icons for **Go Live** and a notification bell.

The code above experienced an error when run because there was no sign; at the end of the code.

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
PS F:\kuliah\semester 5\mobile\4\week4>
PS F:\kuliah\semester 5\mobile\4\week4>
PS F:\kuliah\semester 5\mobile\4\week4>
PS F:\kuliah\semester 5\mobile\4\week4>
```

Step 3:

Add the following program code outside the scope of void main(), then try executing (Run) your code. What happened ? If an error occurs, please correct it.

```
bin > datarecords.dart > tukar
Run | Debug
1 void main(List<String> args) {
2   var record = ('first', a: 2, b: true, 'last');
3   print(record);
4 }
5
6 (int, int) tukar((int, int) record) {
7   var (a, b) = record;
8   return (b, a);
9 }
```

```
PS F:\kuliah\semester 5\mobile\4\week4>
PS F:\kuliah\semester 5\mobile\4\week4>
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
PS F:\kuliah\semester 5\mobile\4\week4>
```

What happened? If an error occurs, please correct it. Use the exchange() function in main() so that the process of exchanging field values in Records is clearly visible.

No error occurred, but code outside the scope of void main() was not executed

The screenshot shows an IDE with three tabs: `datamaps.dart`, `datalist_spread_control_flow.dart 1`, and `datarecords.dart`. The `datarecords.dart` tab is active, showing the following Dart code:

```
bin > datarecords.dart > main
Run | Debug
1 void main(List<String> args) {
2   var record = ('first', a: 2, b: true, 'last');
3   print(record);
4
5   (int, int) tukar((int, int) record) {
6     var (a, b) = record;
7     return (b, a);
8   }
9
10  print(tukar((4, 3)));
11 }
12
```

The terminal window at the bottom shows the command `dart run .\bin\datarecords.dart` and the output:

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
(3, 4)
PS F:\kuliah\semester 5\mobile\4\week4>
```

An error message is displayed in the terminal:

```
Try removing the extra positional arguments.
print(4, 3);
   ^
org-dartlang-sdk:///sdk/lib/core/print.dart:8:6: Context: Found this candidate, but the arguments don't match.
void print(Object? object) {
    ^^^^^
```

Step 4:

Add the following program code in the void main() scope, then try executing (Run) your code.

What happened ? If an error occurs, please correct it. Initialize your name and NIM fields in the student record variable above. Document the results and make a report!

The screenshot shows an IDE with three tabs: `datamaps.dart`, `datalist_spread_control_flow.dart 1`, and `datarecords.dart 1`. The active file is `datarecords.dart`, which contains the following Dart code:

```
bin > datarecords.dart > main
Run | Debug
1 void main(List<String> args) {
2   var record = ('first', a: 2, b: true, 'last');
3   print(record);
4
5   (int, int) tukar((int, int) record) {
6     var (a, b) = record;
7     return (b, a);
8   }
9
10  print(tukar((4, 3)));
11
12  // Record type annotation in a variable declaration:
13  (String, int) mahasiswa;
14  print(mahasiswa);
15 }
16
```

The terminal at the bottom shows the output of running the program:

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
(3, 4)
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
bin/datarecords.dart:14:9: Error: Non-nullable variable 'mahasiswa' must be assigned before it
can be used.
  print(mahasiswa);
  ^^^^^^^^^
PS F:\kuliah\semester 5\mobile\4\week4>
```

When running the program above, an error occurs due to calling var student which has not been given any value. Then improve it by assigning values to student variables according to the record annotation type, namely String and Integer

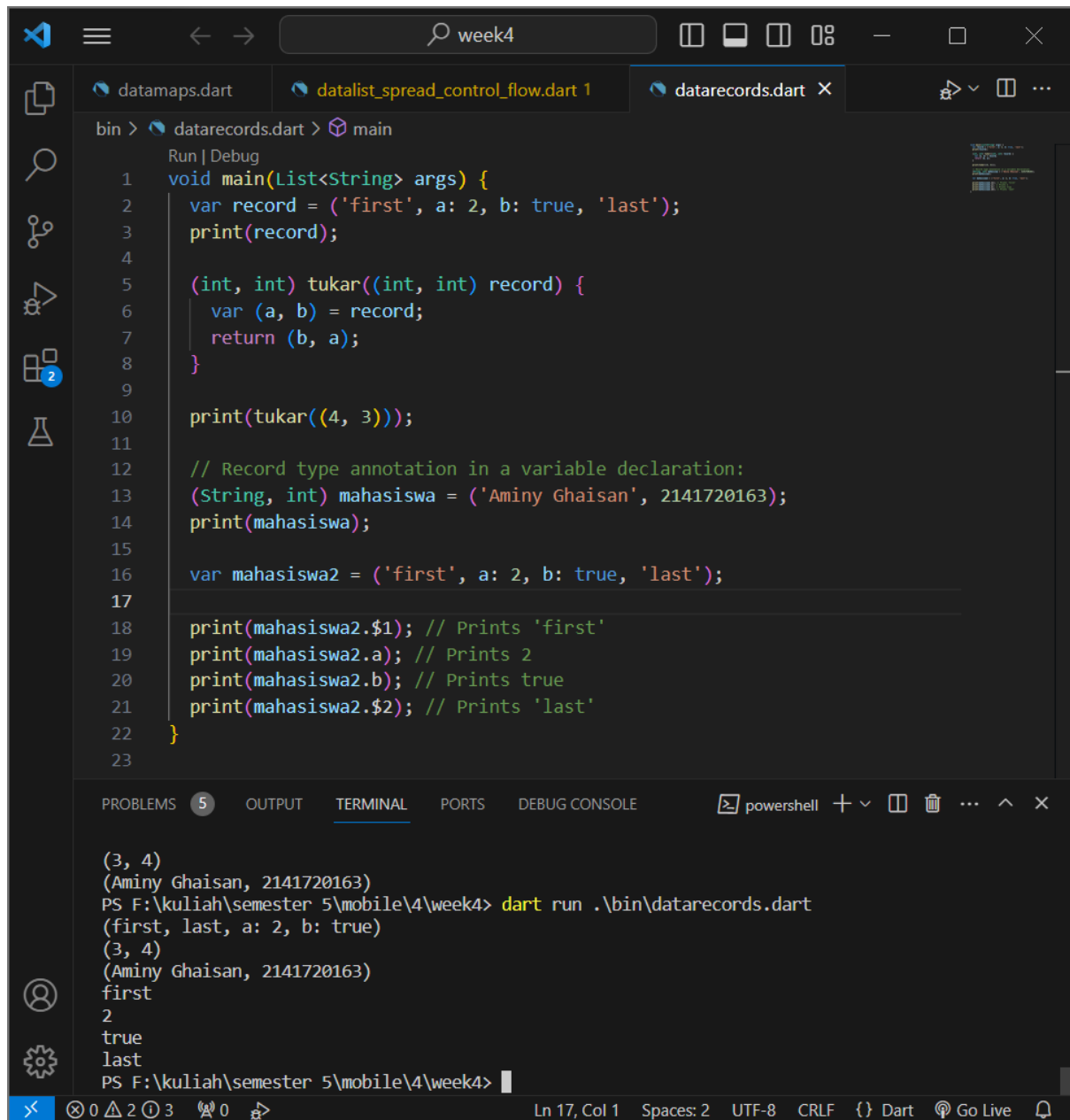
The screenshot shows a terminal window with the following output:

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
(3, 4)
(Aminy Ghaisan, 2141720163)
PS F:\kuliah\semester 5\mobile\4\week4>
```

Step 5:

Add the following program code in the void main() scope, then try executing (Run) your code.

What happened? If an error occurs, please correct it. Replace one of the contents of the record with your name and NIM, then document the results and make a report!



The screenshot shows an IDE with three tabs: `datamaps.dart`, `datalist_spread_control_flow.dart 1`, and `datarecords.dart`. The `datarecords.dart` tab is active, showing the following Dart code:

```
bin > datarecords.dart > main
Run | Debug
1 void main(List<String> args) {
2   var record = ('first', a: 2, b: true, 'last');
3   print(record);
4
5   (int, int) tukar((int, int) record) {
6     var (a, b) = record;
7     return (b, a);
8   }
9
10  print(tukar((4, 3)));
11
12  // Record type annotation in a variable declaration:
13  (String, int) mahasiswa = ('Aminy Ghaisan', 2141720163);
14  print(mahasiswa);
15
16  var mahasiswa2 = ('first', a: 2, b: true, 'last');
17
18  print(mahasiswa2.$1); // Prints 'first'
19  print(mahasiswa2.a); // Prints 2
20  print(mahasiswa2.b); // Prints true
21  print(mahasiswa2.$2); // Prints 'last'
22 }
23
```

The bottom panel shows the `TERMINAL` output:

```
(3, 4)
(Aminy Ghaisan, 2141720163)
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\datarecords.dart
(first, last, a: 2, b: true)
(3, 4)
(Aminy Ghaisan, 2141720163)
first
2
true
last
PS F:\kuliah\semester 5\mobile\4\week4>
```

The status bar at the bottom indicates the cursor is at `Ln 17, Col 1`, with `Spaces: 2`, `UTF-8`, `CRLF`, and `{ } Dart` settings.

## 6. Practicum Assignment

### 2. Jelaskan yang dimaksud Functions dalam bahasa Dart!

Functions in Dart are blocks of code used to execute specific tasks. Functions can receive arguments, perform logic, and return values (if needed). They are used to organize code in a modular way, allowing for reuse and structuring code more effectively.

### 3. Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!

#### a. Positional Optional Parameters (Optional Parameters)

Positional optional parameters are parameters that do not need to be provided when calling a function and can be omitted if not needed.

The screenshot shows an IDE with three tabs: `datalist_spread_control_flow.dart 1`, `datarecords.dart`, and `tugas_praktikum.dart`. The active file is `tugas_praktikum.dart`, which contains the following Dart code:

```
bin > tugas_praktikum.dart > printInfo
1 // Function with positional optional parameters
2 void printInfo(String name, [int? age, String? country]) {
3     print('Name: $name');
4     if (age != null) {
5         print('Age: $age');
6     }
7     if (country != null) {
8         print('Country: $country');
9     }
10 }
11
12 Run | Debug
13 void main() {
14     printInfo('Alice'); // Only name is provided
15     printInfo('Bob', 30); // Name and age are provided
16     printInfo('Charlie', 25, 'USA'); // Name, age, and country are provided
17 }
```

The terminal output shows the execution of the program:

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Name: Alice
Name: Bob
Age: 30
Name: Charlie
Age: 25
Country: USA
PS F:\kuliah\semester 5\mobile\4\week4>
```



The screenshot shows an IDE with three tabs: `datalist_spread_control_flow.dart 1`, `datarecords.dart`, and `tugas_praktikum.dart`. The active file `tugas_praktikum.dart` contains the following Dart code:

```
bin > tugas_praktikum.dart > ...
1 // Function with default parameters
2 void printInfo(String name, {int age = 25, String country = 'Unknown'}) {
3     print('Name: $name');
4     print('Age: $age');
5     print('Country: $country');
6 }
7
8 Run | Debug
9 void main() {
10     printInfo('Alice'); // Only name is provided, age and country use default
11     printInfo('Bob', age: 30); // Name and age are provided, country uses default
12     printInfo('Charlie', age: 25, country: 'USA'); // Name, age, and country are provided
13 }
```

The terminal at the bottom shows the command `dart run .\bin\tugas_praktikum.dart` being executed, resulting in the following output:

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Name: Alice
Age: 25
Country: Unknown
Name: Bob
Age: 30
Country: Unknown
Name: Charlie
Age: 25
Country: USA
PS F:\kuliah\semester 5\mobile\4\week4>
```

In this code, we have a function `printInfo` that takes a required name parameter and two optional parameters `age` and `country`. Both optional parameters have default values assigned. When calling the function, if you don't provide values for the optional parameters, their default values will be used.

c. Named Parameter (Dynamic Optional Parameter)

These parameters do not need to be provided when calling a function, and you can specify them by name when calling the function.

The screenshot shows the Visual Studio Code editor with a Dart file named `tugas_praktikum.dart`. The code defines a function `printInfo` with named parameters and a `main` function that calls it with different parameter sets. The terminal at the bottom shows the output of running the program, demonstrating how named parameters allow for flexible function calls.

```
bin > tugas_praktikum.dart > ...
1 // Function with named parameters
2 void printInfo({String name = 'Unknown', int age = 25, String country = 'U
3     print('Name: $name');
4     print('Age: $age');
5     print('Country: $country');
6 }
7
Run | Debug
8 void main() {
9     // Calling the function with named parameters
10    printInfo(name: 'Alice'); // Only name is provided, age and country use
11    printInfo(name: 'Bob', age: 30); // Name and age are provided, country u
12    printInfo(name: 'Charlie', age: 25, country: 'USA'); // Name, age, and c
13 }
14
```

TERMINAL

```
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Name: Alice
Age: 25
Country: Unknown
Name: Bob
Age: 30
Country: Unknown
Name: Charlie
Age: 25
Country: USA
PS F:\kuliah\semester 5\mobile\4\week4>
```

In this code, we have a function `printInfo` that uses named parameters. You can specify the parameter values by name when calling the function, which allows you to provide them in any order or omit them if you don't need to change the default values.

d. Required Parameter (Mandatory Parameter)

Required parameters are parameters that must be provided when calling a function and cannot be omitted.

The screenshot shows an IDE window with three tabs: `datalist_spread_control_flow.dart 1`, `datarecords.dart`, and `tugas_praktikum.dart`. The active file is `tugas_praktikum.dart`, which contains the following Dart code:

```
bin > tugas_praktikum.dart > ...
1 // Function with required parameters
2 void printInfo(String name, int age) {
3     print('Name: $name');
4     print('Age: $age');
5 }
6
7 Run | Debug
8 void main() {
9     // Calling the function with required parameters
10    printInfo('Alice', 25); // Both name and age are provided
11    // Uncommenting the line below would result in a compilation error
12    // printInfo('Bob'); // Error: Missing required argument: 'age'
13 }
```

The bottom panel shows the `TERMINAL` output, which displays the results of running the program three times:

```
Age: 25
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Name: Alice
Age: 25
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Name: Alice
Age: 25
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Name: Alice
Age: 25
PS F:\kuliah\semester 5\mobile\4\week4>
```

The status bar at the bottom indicates the current position is `Ln 13, Col 1`, with `Spaces: 2`, `UTF-8` encoding, `CRLF` line endings, and the file is a `Dart` script.

In this code, we have a function `printInfo` that takes two required parameters: `name` and `age`. When calling the function, both of these parameters must be provided, and omitting any of them will result in a compilation error.

#### 4. Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!

Functions as first-class objects refer to the ability to treat functions just like any other objects, such as strings, numbers, or other objects. This means that you can manipulate functions like variables, store them in variables, pass them as arguments to other functions, return them from other functions, and so on. An example syntax is as shown in the image below.

The screenshot shows an IDE with three tabs: `datalist_spread_control_flow.dart 1`, `datarecords.dart`, and `tugas_praktikum.dart`. The active file `tugas_praktikum.dart` contains the following Dart code:

```
bin > tugas_praktikum.dart > ...
7   return a - b;
8   }
9
10  Run | Debug
11  void main() {
12      // Storing functions in variables
13      int Function(int, int) operation;
14
15      operation = add;
16      print(operation(5, 3)); // Output: 8
17
18      operation = subtract;
19      print(operation(5, 3)); // Output: 2
20
21      // Passing a function as an argument to another function
22      int result = performOperation(10, 5, add);
23      print(result); // Output: 15
24  }
25
26  int performOperation(int a, int b, int Function(int, int) operation) {
27      return operation(a, b);
28  }
```

The bottom panel shows the **TERMINAL** tab with the following output:

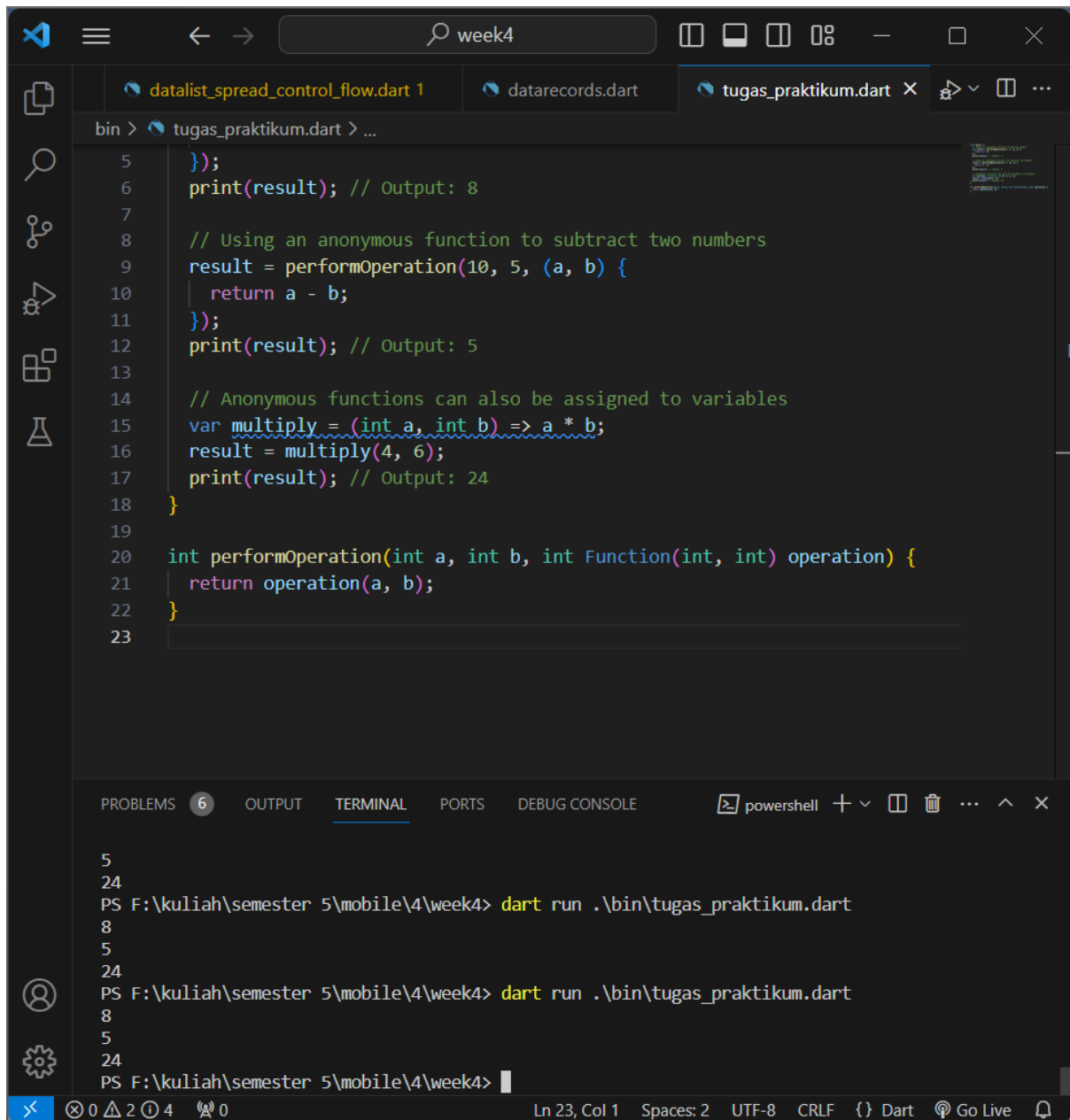
```
* History restored
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
8
2
15
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
8
2
15
PS F:\kuliah\semester 5\mobile\4\week4> |
```

The status bar at the bottom indicates: Ln 28, Col 1 | Spaces: 2 | UTF-8 | CRLF | {} Dart | Go Live.

In this code, we define two functions `add` and `subtract`. We also declare a variable `operation` that can store functions with a specific signature. We then demonstrate storing functions in variables, passing functions as arguments to another function, and using functions as first-class objects.

5. Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!

An Anonymous Function in Dart is a function that doesn't have a name and is often referred to as a lambda or closure. These functions are commonly used to write more concise code and can be assigned to variables or used in various contexts, such as adding them to or removing them from collections. Anonymous functions have parameters similar to named functions and are declared without a function name. An example syntax can be seen in this image.



The screenshot shows an IDE with three tabs: `datalist_spread_control_flow.dart 1`, `datarecords.dart`, and `tugas_praktikum.dart`. The active file is `tugas_praktikum.dart`, which contains the following Dart code:

```
5   });
6   print(result); // Output: 8
7
8   // Using an anonymous function to subtract two numbers
9   result = performOperation(10, 5, (a, b) {
10      return a - b;
11   });
12   print(result); // Output: 5
13
14   // Anonymous functions can also be assigned to variables
15   var multiply = (int a, int b) => a * b;
16   result = multiply(4, 6);
17   print(result); // Output: 24
18 }
19
20 int performOperation(int a, int b, int Function(int, int) operation) {
21     return operation(a, b);
22 }
23
```

The bottom panel shows the `TERMINAL` output, which displays the execution of the code twice:

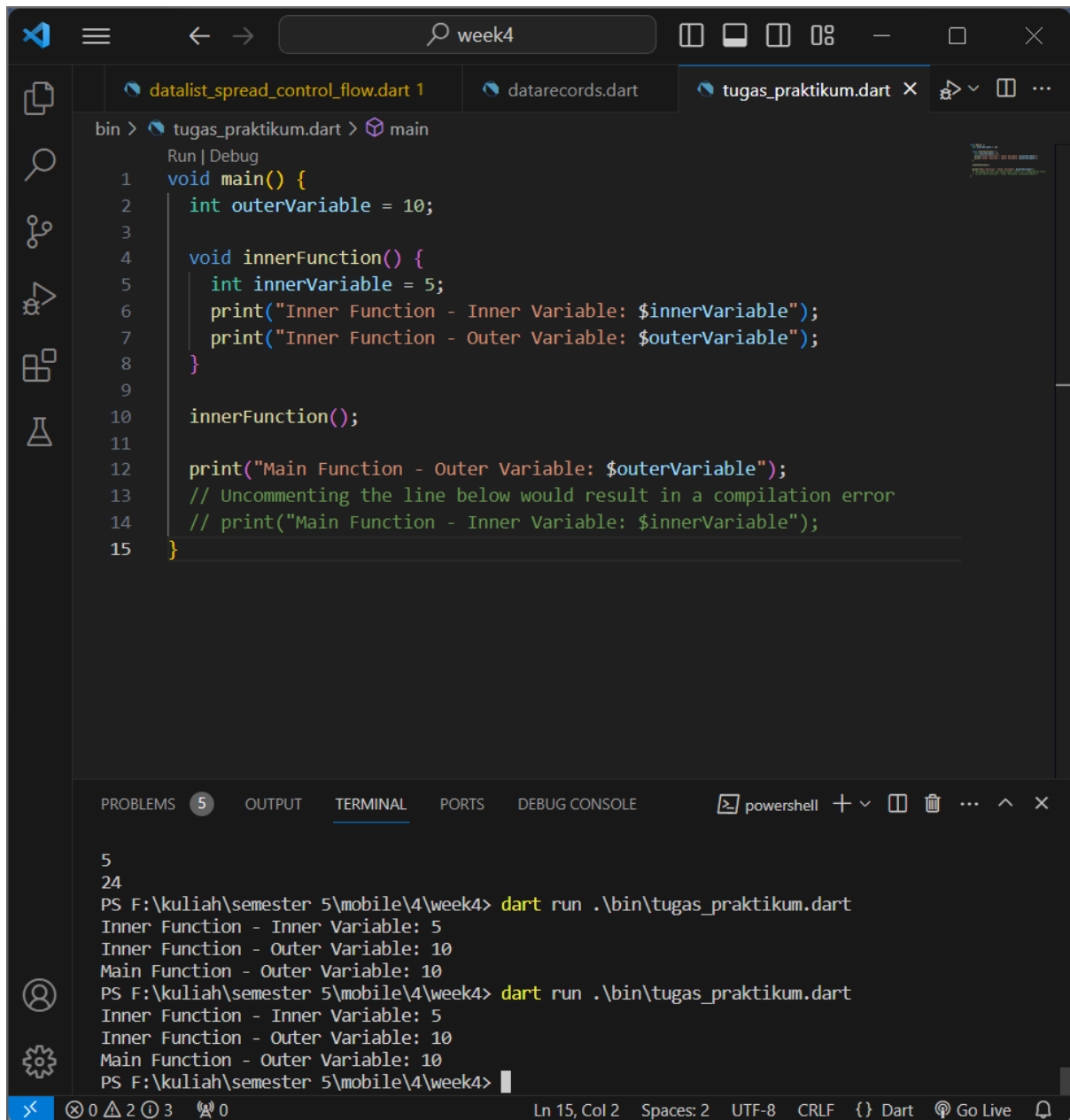
```
5
24
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
8
5
24
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
8
5
24
PS F:\kuliah\semester 5\mobile\4\week4>
```

In this code, we use anonymous functions to define the logic for adding and subtracting numbers. These anonymous functions are then passed as arguments to the `performOperation` function. Additionally, we demonstrate how to assign an anonymous function to a variable and use it to perform a multiplication operation.

6. Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!

The difference between Lexical Scope and Lexical Closure

a. Lexical Scope: Variables have a scope determined by the static structure of the code. You can see where a variable is by looking at the code layout.



The screenshot shows an IDE window with three tabs: `datalist_spread_control_flow.dart 1`, `datarecords.dart`, and `tugas_praktikum.dart`. The active file is `tugas_praktikum.dart`, which contains the following Dart code:

```
bin > tugas_praktikum.dart > main
Run | Debug
1 void main() {
2   int outerVariable = 10;
3
4   void innerFunction() {
5     int innerVariable = 5;
6     print("Inner Function - Inner Variable: $innerVariable");
7     print("Inner Function - Outer Variable: $outerVariable");
8   }
9
10  innerFunction();
11
12  print("Main Function - Outer Variable: $outerVariable");
13  // Uncommenting the line below would result in a compilation error
14  // print("Main Function - Inner Variable: $innerVariable");
15 }
```

The terminal at the bottom shows the output of running the code twice:

```
5
24
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Inner Function - Inner Variable: 5
Inner Function - Outer Variable: 10
Main Function - Outer Variable: 10
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Inner Function - Inner Variable: 5
Inner Function - Outer Variable: 10
Main Function - Outer Variable: 10
PS F:\kuliah\semester 5\mobile\4\week4>
```

In this code, we have an `outerVariable` declared in the main function, and an `innerFunction` defined within the main function. The `innerFunction` has its own `innerVariable`. Notice that we can access the `outerVariable` from within the `innerFunction` because it is in the lexical scope of `innerFunction`. However, trying to access the `innerVariable` from outside the `innerFunction` (in the main function) would result in a compilation error because it's not in the lexical scope of the main function. Lexical scope determines where variables are accessible based on their location in the code.

b. **Lexical Closures:** Functions can access variables in their surrounding scope, even when the function is used outside its original scope. The function "closes" these variables, allowing them to remain accessible.

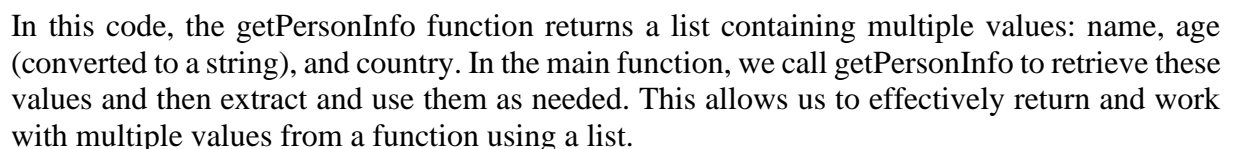
The screenshot shows an IDE with a file explorer at the top displaying three files: `datalist_spread_control_flow.dart 1`, `datarecords.dart`, and `tugas_praktikum.dart`. The main editor window shows the content of `tugas_praktikum.dart` with the following Dart code:

```
bin > tugas_praktikum.dart > ...
Run | Debug
1 void main() {
2     int outerVariable = 10;
3
4     Function closureFunction() {
5         int innerVariable = 5;
6
7         void innerFunction() {
8             print("Inner Function - Inner Variable: $innerVariable");
9             print("Inner Function - Outer Variable: $outerVariable");
10        }
11
12        return innerFunction;
13    }
14
15    final closure = closureFunction();
16
17    closure();
18
19    print("Main Function - Outer Variable: $outerVariable");
20 }
21
```

Below the editor is a terminal window with the following output:

```
PROBLEMS 5 OUTPUT TERMINAL PORTS DEBUG CONSOLE powershell + v [ ] [ ] ... ^ X
Inner Function - Outer Variable: 10
Main Function - Outer Variable: 10
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Inner Function - Inner Variable: 5
Inner Function - Outer Variable: 10
Main Function - Outer Variable: 10
PS F:\kuliah\semester 5\mobile\4\week4> dart run .\bin\tugas_praktikum.dart
Inner Function - Inner Variable: 5
Inner Function - Outer Variable: 10
Main Function - Outer Variable: 10
PS F:\kuliah\semester 5\mobile\4\week4>
```

In Dart, we can return multiple values from a function by packaging the values into an appropriate object or data type. One common way is to return a list. Here's the code to create a function that returns multiple values using a list:





Github Link:

[https://github.com/aminyG/Mobile\\_Programming/tree/main/week4](https://github.com/aminyG/Mobile_Programming/tree/main/week4)