

Assignment 2 (Dart)

Question 1

1:What are the various types of operators in dart? Explain with Examples.

Ans :Following are the operators in dart

1.Arithmetic Operators

2.Relational Operators

3.Logical Operators

1.Arithmetic Operators:

Addition	+	Adds one operand to the other
Subtraction	-	Subtracts the second operand from the first
Multiplication	*	Multiplies one operand by the other
Division	/	Divides the first operand by the second
Modulo remainder	%	Divides the first INTEGER operand by the second, and returns the remainder

```
void main()
```

```
{
```

```
  int a = 5;
```

```
  int b = 6;
```

```
  // Adding a and b
```

```
var c = a + b;

print("Sum of a and b is $c");


// Subtracting a and b

var d = a - b;

print("The difference between a and b is $d");


// Multiplication of a and b

var f = a * b;

print("The product of a and b is $f");


// Division of a and b

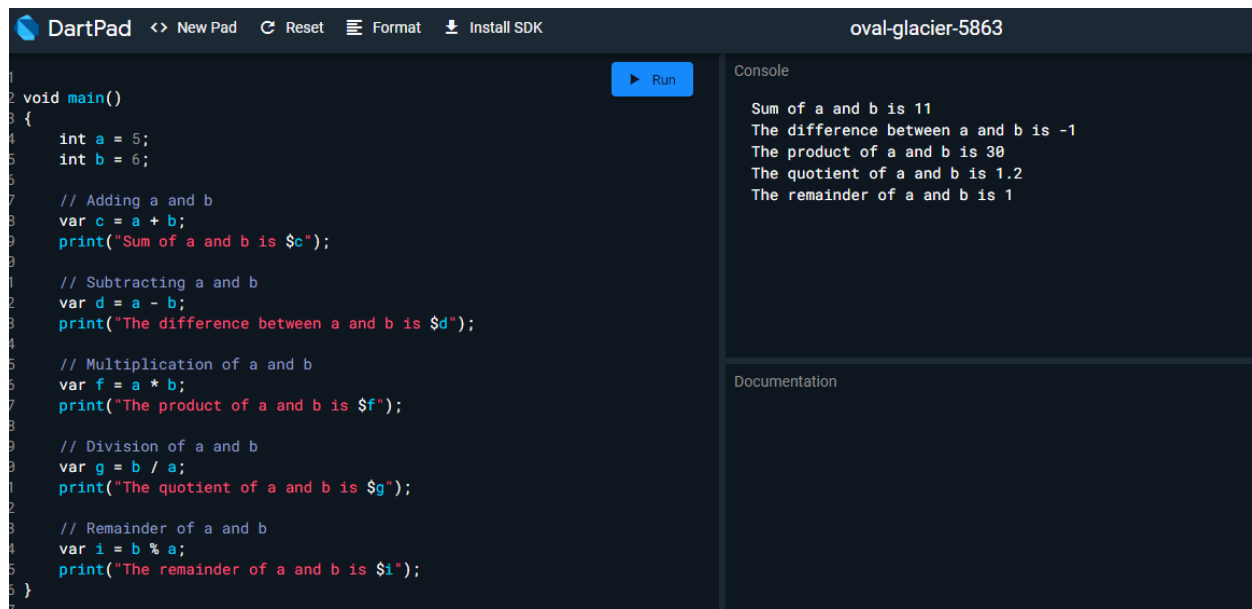
var g = b / a;

print("The quotient of a and b is $g");


// Remainder of a and b

var i = b % a;

print("The remainder of a and b is $i");
}
```



The screenshot shows the DartPad web interface. The top bar includes the DartPad logo, navigation links (New Pad, Reset, Format, Install SDK), and the user identifier 'oval-glacier-5863'. The main editor area contains the following Dart code:

```
1 void main()
2 {
3   int a = 5;
4   int b = 6;
5
6   // Adding a and b
7   var c = a + b;
8   print("Sum of a and b is $c");
9
10  // Subtracting a and b
11  var d = a - b;
12  print("The difference between a and b is $d");
13
14  // Multiplication of a and b
15  var f = a * b;
16  print("The product of a and b is $f");
17
18  // Division of a and b
19  var g = b / a;
20  print("The quotient of a and b is $g");
21
22  // Remainder of a and b
23  var i = b % a;
24  print("The remainder of a and b is $i");
25 }
```

A 'Run' button is located to the right of the code editor. The console on the right displays the output of the code:

```
Sum of a and b is 11
The difference between a and b is -1
The product of a and b is 30
The quotient of a and b is 1.2
The remainder of a and b is 1
```

Below the console is a 'Documentation' section.

2.Relational Operators:

- > Greater than
- < Less than
- >= Greater than or equal to
- <= Less than or equal to
- == Equal to
- != Not equal to

```
void main()
```

```
{
```

```
    int a = 4;
```

```
    int b = 5;
```

```
// Greater between a and b
```

```
var c = a > b;
```

```
print("a is greater than b is $c");
```

```
// Smaller between a and b
```

```
var d = a < b;
```

```
print("a is smaller than b is $d");
```

```
// Greater than or equal to between a and b
```

```
var e = a >= b;
```

```
print("a is greater than b is $e");
```

```
// Less than or equal to between a and b
```

```
var f = a <= b;
```

```
print("a is smaller than b is $f");
```

```
// Equality between a and b
```

```
var g = b == a;
```

```
print("a and b are equal is $g");
```

```
// Unequality between a and b
```

```
var h = b != a;
```

```
print("a and b are not equal is $h");
```

```
}
```

```
void main()
{
    int a = 4;
    int b = 5;

    // Greater between a and b
    var c = a > b;
    print("a is greater than b is $c");

    // Smaller between a and b
    var d = a < b;
    print("a is smaller than b is $d");

    // Greater than or equal to between a and b
    var e = a >= b;
    print("a is greater than b is $e");

    // Less than or equal to between a and b
    var f = a <= b;
    print("a is smaller than b is $f");

    // Equality between a and b
    var g = b == a;
    print("a and b are equal is $g");

    // Unequality between a and b
    var h = b != a;
    print("a and b are not equal is $h");
}
```

Run

Console

```
a is greater than b is false
a is smaller than b is true
a is greater than b is false
a is smaller than b is true
a and b are equal is false
a and b are not equal is true
```

Documentation

3.Logical Operators

&& And Operator Use to add two conditions and if both are true than it will return true.

|| Or Operator Use to add two conditions and if even one of them is true than it will return true.

! Not Operator It is use to reverse the result.

```
void main()
```

```
{
```

```
    int a = 8;
```

```
    int b = 9;
```

```
    // Using And Operator
```

```
bool c = a > 10 && b < 10;
```

```
print(c);
```

```
// Using Or Operator
```

```
bool d = a > 10 || b < 10;
```

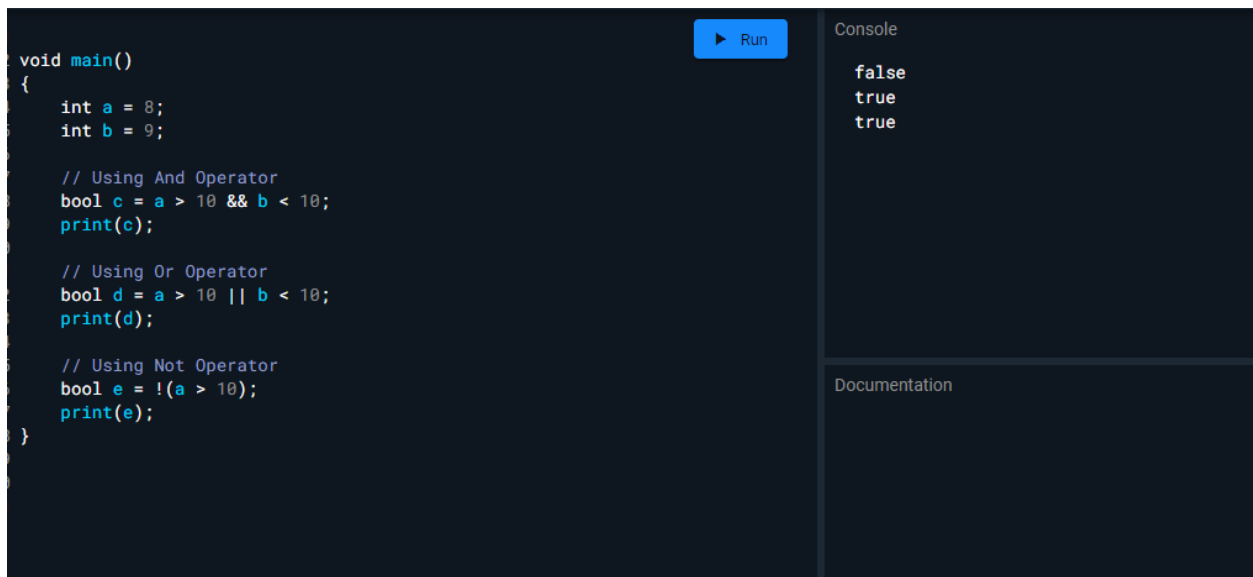
```
print(d);
```

```
// Using Not Operator
```

```
bool e = !(a > 10);
```

```
print(e);
```

```
}
```



The screenshot shows a C++ IDE with a dark theme. The code in the editor defines a `main` function where `a = 8` and `b = 9`. It demonstrates three logical operations: `&&` (AND), `||` (OR), and `!` (NOT). The console output shows the results of these operations: `false` for the AND operation, and `true` for both the OR and NOT operations. A 'Run' button is visible in the top right of the editor area.

```
void main()
{
    int a = 8;
    int b = 9;

    // Using And Operator
    bool c = a > 10 && b < 10;
    print(c);

    // Using Or Operator
    bool d = a > 10 || b < 10;
    print(d);

    // Using Not Operator
    bool e = !(a > 10);
    print(e);
}
```

Console

```
false
true
true
```

Documentation

Question 2

```

void main()

{

    var a = 2;

    var b = 1;


    print('part c = ${--a}'); //value of a is decrement by 1 times

    print('part d = ${--a - --b}'); //value of a is decrement by one times and value of B is
    decrement by one time then value of a is subtracted from value of B

    print('part e = ${--a - --b + ++b}'); //value of a is decrement by one times and value of B is
    decrement by one time then value of a is subtracted from value of B and then value of B is
    increment by one time and add into final value

    print('part result = ${--a - --b + ++b + b--}'); //value of a is decrement by one times and value
    of B is decrement by one time then value of a is subtracted from value of B and then value of B
    is increment by one time and add into final value then again value of B is added to final one
    and push the value without doing decrement


}

```

```
1
2 void main()
3 {
4     var a = 2;
5     var b = 1;
6
7
8     print('part c = ${--a}'); //value of a is decrement by 1 times
9     print('part d = ${--a - --b}'); //value of a is decrement by one times and
10    print('part e = ${--a - --b + ++b}'); //value of a is decrement by one time
11    print('part result = ${--a - --b + ++b + b--}'); //value of a is decrement
12
13
14
15 }
16
17
```

Run

Console

```
part c = 1
part d = 0
part e = 0
part result = -1
```

Documentation

List<dynamic> a

local variable

Question 3

```
void main() {

    var cost = 600;

    var buying = 5;

    var result = cost * buying;

    print('cost of buying ${buying} tickets ${result}');

}
```

```
1
2 void main() {
3
4     var cost = 600;
5     var buying = 5;
6     var result = cost * buying;
7     print('cost of buying ${buying} tickets ${result}');
8 }
9
10
```

Run

Console

```
cost of buying 5 tickets 3000
```

Documentation

Question 4

```
void main()

{

List<int> one = [1,2,3,4,5,6,7];

List<int> two = [3,5,6,7,9,10];

List<int> difference = one.toSet().difference(two.toSet()).toList();

print(difference.toString());

}
```



The screenshot shows an IDE with a code editor on the left and a console on the right. The code in the editor is as follows:

```
1
2
3 void main()
4 {
5     List<int> one = [1,2,3,4,5,6,7];
6     List<int> two = [3,5,6,7,9,10];
7     List<int> difference = one.toSet().difference(two.toSet()).toList();
8     print(difference.toString());
9 }
10
```

A blue "Run" button is visible above the code. The console on the right displays the output: `[1, 2, 4]`. Below the console is a "Documentation" section, and at the bottom right, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

Question 5

?

```

void main() {

    print(2 == 2 ? "a truth" : "a lie"); // <- a truth

    print(1 == 2 ? "a truth" : "a lie"); // <- a lie

    5 == 6 ? doThis() : doThat(); // <- done that

}

void doThis() {

    print('done this');

}

void doThat() {

    print('done that');

}

```

```

1 void main() {
2   print(2 == 2 ? "a truth" : "a lie"); // <- a truth
3   print(1 == 2 ? "a truth" : "a lie"); // <- a lie
4   5 == 6 ? doThis() : doThat(); // <- done that
5 }
6
7 void doThis() {
8   print('done this');
9 }
10 void doThat() {
11   print('done that');
12 }
13
14
15
16
17
18

```

Run

Console

```

a truth
a lie
done that

```

Documentation

??

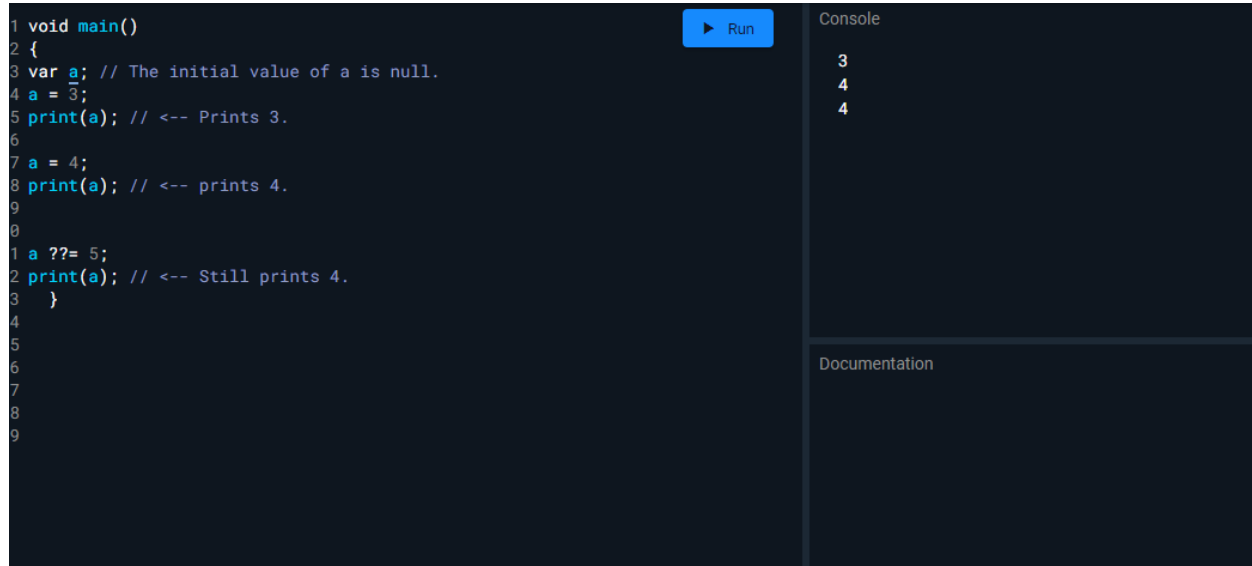
```

void main()
{
var a; // The initial value of a is null.
a = 3;
print(a); // <-- Prints 3.

```

```
a = 4;
print(a); // <-- prints 4.
```

```
a ??= 5;
print(a); // <-- Still prints 4.
}
```



The screenshot shows a code editor with the following Dart code:

```
1 void main()
2 {
3   var a; // The initial value of a is null.
4   a = 3;
5   print(a); // <-- Prints 3.
6
7   a = 4;
8   print(a); // <-- prints 4.
9
10
11   a ??= 5;
12   print(a); // <-- Still prints 4.
13 }
14
15
16
17
18
19
```

A blue "Run" button is located to the right of the code. The console on the right shows the output:

```
3
4
4
```

The documentation panel on the right is empty.

Question 6

Data type	Keyword	Description
Number:	int, double, num	Numbers in Dart are used to represent numeric literals
Strings	String	Strings represent a sequence of characters
Booleans	bool	It represents Boolean values true and false
Lists	List	It is an ordered group of objects

Examples:

1. Int :

```
void main() {
```

```
    // declare an integer
```

```
    int num1 = 3;
```

```
    // declare a double value
```

```
    double num2 = 2.5;
```

```
    // print the values
```

```
    print(num1);
```

```
    print(num2);
```

```
    var a1 = num.parse("1");
```

```
    var b1 = num.parse("3.34");
```

```
    var sum = a1+b1;
```

```
    print("Product = ${sum}");
```

```
}
```

output

3

2.5

product = 4.34

2. String

```
void main() {
```

```
    String string = 'hello dart ';
```

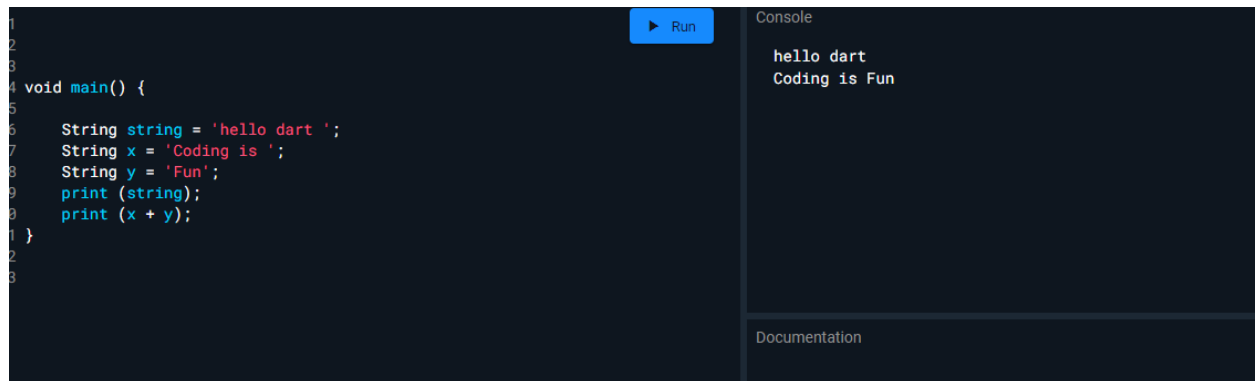
```
    String x = 'Coding is ';
```

```
    String y = 'Fun';
```

```
    print (string);
```

```
    print (x + y);
```

```
}
```



The screenshot shows a code editor with the following Dart code:

```
1  
2  
3  
4 void main() {  
5  
6     String string = 'hello dart ';  
7     String x = 'Coding is ';  
8     String y = 'Fun';  
9     print (string);  
10    print (x + y);  
11 }  
12  
13
```

A blue 'Run' button is visible above the code. To the right, the 'Console' panel displays the output:

```
hello dart  
Coding is Fun
```

Below the console, there is a 'Documentation' section which is currently empty.

3. Boolean:

```
void main() {
```

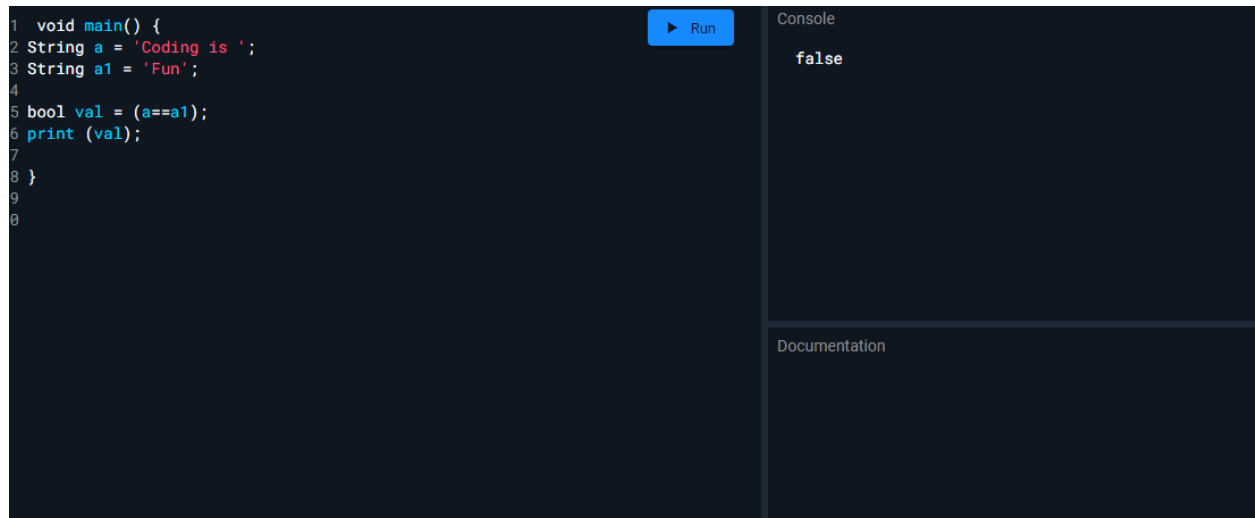
```
    String a = 'Coding is ';
```

```
    String a1 = 'Fun';
```

```
bool val = (a==a1);

print (val);

}
```

A screenshot of a code editor with a dark theme. The editor shows a C# program with the following code:

```
1 void main() {
2   String a = 'Coding is ';
3   String a1 = 'Fun';
4
5   bool val = (a==a1);
6   print (val);
7
8 }
9
10
```

A blue 'Run' button is located to the right of the code. To the right of the editor is a 'Console' panel showing the output 'false'. Below the console is a 'Documentation' panel, which is currently empty.

4.List

```
void main()

{

    List a = new List(3);

    a[0] = 'hi';

    a[1] = 'hello';

    a[2] = 'bye';

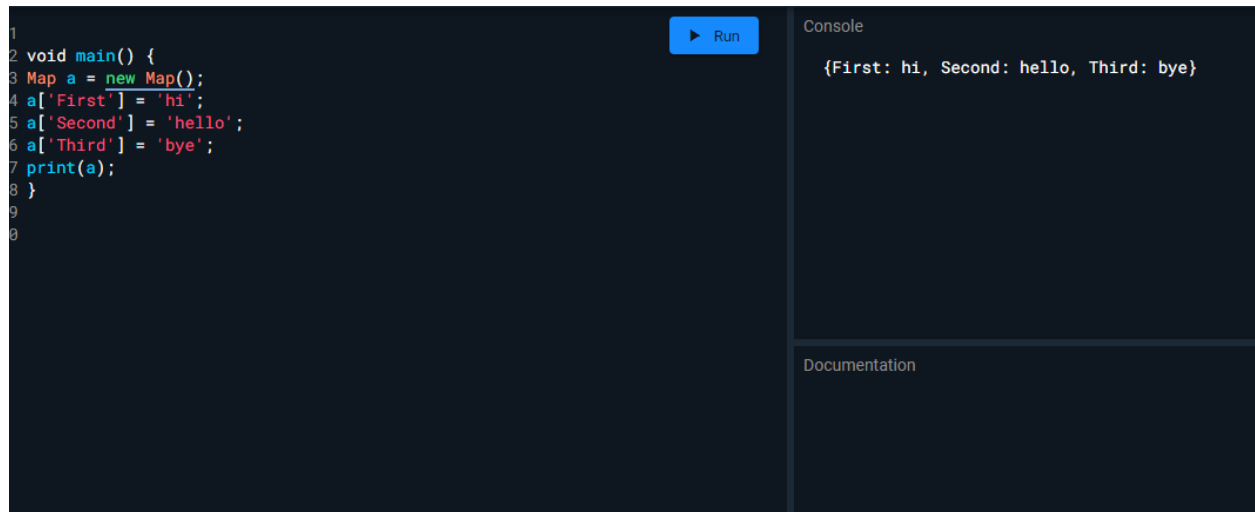

    print(a);

    print(a[0]);

}
```

5.Map

```
void main() {  
  
    Map a = new Map();  
  
    a['First'] = 'hi';  
  
    a['Second'] = 'hello';  
  
    a['Third'] = 'bye';  
  
    print(a);  
  
}
```



The screenshot shows a code editor with a dark theme. On the left, there is a Dart code snippet for creating and printing a Map. On the right, there is a 'Console' panel showing the output of the code, which is a string representation of the Map: '{First: hi, Second: hello, Third: bye}'. Below the console, there is a 'Documentation' panel which is currently empty.

```
1  
2 void main() {  
3   Map a = new Map();  
4   a['First'] = 'hi';  
5   a['Second'] = 'hello';  
6   a['Third'] = 'bye';  
7   print(a);  
8 }  
9  
0
```

Run

Console

```
{First: hi, Second: hello, Third: bye}
```

Documentation

QUESTION 7:

```
void main() {
```

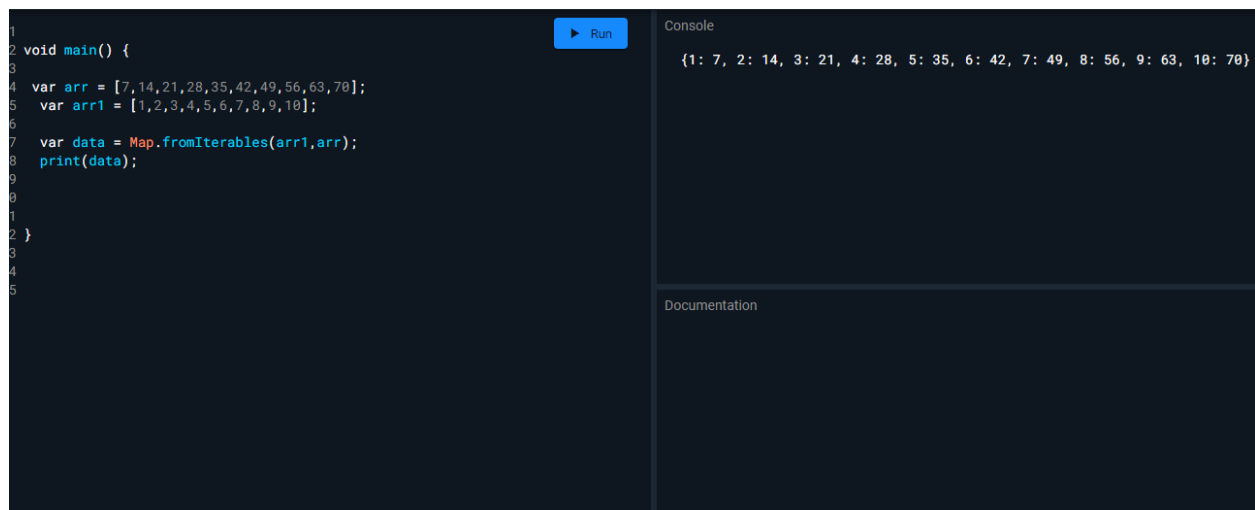
```
    var arr = [7,14,21,28,35,42,49,56,63,70];
```

```
var arr1 = [1,2,3,4,5,6,7,8,9,10];
```

```
var data = Map.fromIterables(arr1,arr);
```

```
print(data);
```

```
}
```



The screenshot shows a code editor with a dark theme. On the left, a Dart program is written in a monospaced font. The code defines a `main` function, creates an array `arr` with 10 integers, another array `arr1` with 10 integers, and then uses `Map.fromIterables` to create a `Map` from these two arrays. The result is printed. A blue 'Run' button is visible above the code. On the right, the 'Console' panel shows the output of the program: a map with 10 entries, each consisting of an index and a value. Below the console, there is a 'Documentation' panel which is currently empty.

```
1 void main() {  
2  
3  
4   var arr = [7, 14, 21, 28, 35, 42, 49, 56, 63, 70];  
5   var arr1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];  
6  
7   var data = Map.fromIterables(arr1, arr);  
8   print(data);  
9  
10  
11 }  
12  
13  
14  
15
```

Run

Console

```
{1: 7, 2: 14, 3: 21, 4: 28, 5: 35, 6: 42, 7: 49, 8: 56, 9: 63, 10: 70}
```

Documentation

QUESTION 9:

```
void main()
```

```
{
```

```
var arr = ['Hasan' , 'ALI' , 'HAMZA'];
```

```
var arr1 = [419,380,390];
```

```
var data = Map.fromIterables(arr,arr1);
```

```
print(data);
```



```

var percent_1 = arr1[0] / 500 * 100;

var percent_2 = arr1[1] / 500 * 100;

var percent_3 = arr1[2] / 500 * 100;


print("percentage of ${arr[0]} : ${percent_1}");

print("percentage of ${arr[1]} : ${percent_2}");

print("percentage of ${arr[2]} : ${percent_3}");

}

```

The screenshot shows a code editor with the following Scala code:

```

1 void main()
2 {
3   var arr = ['Hasan', 'ALI', 'HAMZA'];
4   var arr1 = [419, 380, 390];
5   var data = Map.fromIterables(arr, arr1);
6   print(data);
7
8   var percent_1 = arr1[0] / 500 * 100;
9   var percent_2 = arr1[1] / 500 * 100;
10  var percent_3 = arr1[2] / 500 * 100;
11
12  print("percentage of ${arr[0]} : ${percent_1}");
13  print("percentage of ${arr[1]} : ${percent_2}");
14  print("percentage of ${arr[2]} : ${percent_3}");
15 }
16
17
18
19
20

```

The console output is:

```

{Hasan: 419, ALI: 380, HAMZA: 390}
percentage of Hasan : 83.8
percentage of ALI : 76
percentage of HAMZA : 78

```

Question 10:

Declare 5 legal & 5 illegal variable names.

LEGAL VARIABLES:

1.hello_hi

2.hamza

3.num_123

4.hassan_var

5.ali456

ILLEGAL VARIABLES

1. 69var_a

2. \$pen

3. \$snake#1

4. &mobile

5. ./net

QUESTION 11:

```
void main()
```

```
{
```

```
    //var mylist = ['hyder','abad'];
```

```
    var myList = ['hyder', 'abad'];
```

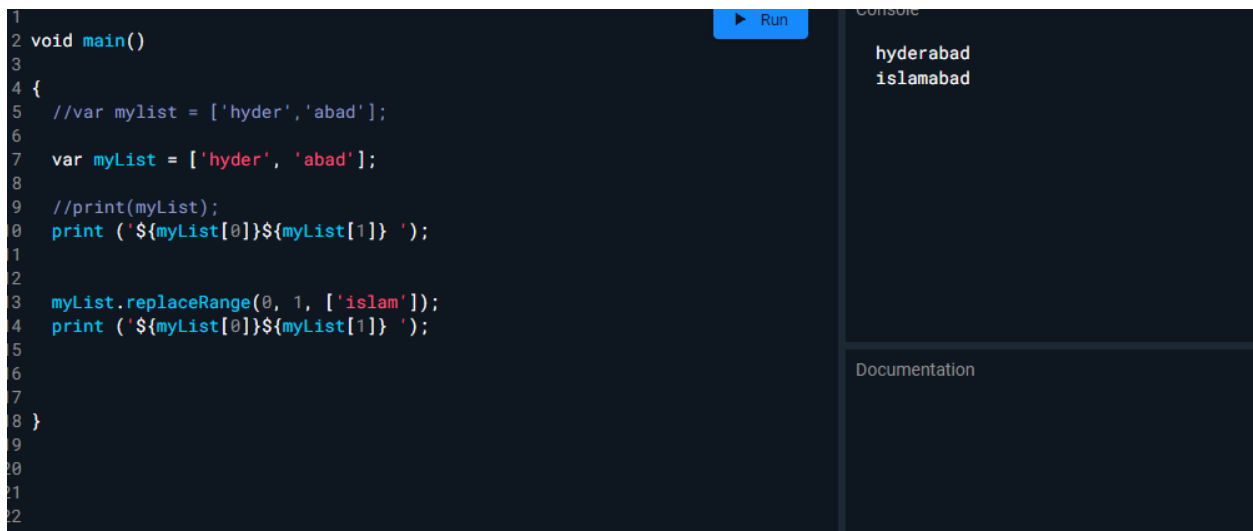
```
    //print(myList);
```

```
    print ('${myList[0]}${myList[1]} ');
```

```
myList.replaceRange(0, 1, ['islam']);
```

```
print ('${myList[0]}${myList[1]} ');
```

```
}
```



```
1
2 void main()
3
4 {
5     //var mylist = ['hyder','abad'];
6
7     var myList = ['hyder', 'abad'];
8
9     //print(myList);
10    print ('${myList[0]}${myList[1]} ');
11
12
13    myList.replaceRange(0, 1, ['islam']);
14    print ('${myList[0]}${myList[1]} ');
15
16
17 }
18
19
20
21
22
```

Run

Console

hyderabad
islamabad

Documentation

QUESTION 13:

```
var date = 1;
```

```
if (date < 16 )
```

```
{
```

```
    print("First fifteen days of the month");
```

```
}
```

else

{

print ("Last days of the months");

}

```
1 void main()
2 {
3   var date = 1;
4   if (date < 16 )
5   {
6     print("First fifteen days of the month");
7   }
8   else
9   {
10    print ("Last days of the months");
11  }
12 }
13
14
15
16
17
```

Run

Console

First fifteen days of the month

Documentation

```
1 void main()
2 {
3   var date = 17;
4   if (date < 16 )
5   {
6     print("First fifteen days of the month");
7   }
8   else
9   {
10    print ("Last days of the months");
11  }
12 }
13
14
15
16
17
```

Run

Console

Last days of the months

Documentation

