**Name: Muhammad Ethisham**

**Roll Number: KWOWFL3434**

**Assignment:03**

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

The following are the various types of operators in Dart:

**Arithmetic Operators:**

+ Addition Use to add two operands

– Subtraction Use to subtract two operands

\* Multiply Use to multiply two operands

/ Division Use to divide two operands

~/ Division Use two divide two operands but give output in integer

% Modulus Use to give remainder of two operands

**Code:**

void main()

{

int a = 2;

int b = 3;

// 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");

// Using unary minus

var e = -d;

print("The negation of difference between a and b is $e");

// 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");

// Using ~/ to divide a and b

var h = b ~ / a;

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

// Remainder of a and b

var i = b % a;

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

}

Assignment Operators:

= Equal to Use to assign values to the expression or variable

??= Assignment operator Assign the value only if it is null.

**Code:**

void main()

{

int a = 5;

int b = 7;

// Assigning value to variable c

var c = a \* b;

print(c);

// Assigning value to variable d

var d;

d ? ? = a + b; // Value is assign as it is null

print(d);

// Again trying to assign value to d

d ? ? = a - b; // Value is not assign as it is not null

print(d);

}

**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.

**Code:**

void main()

{

int a = 5;

int b = 7;

// 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);

}

**Relational Operators:**

> Greater than (Check which operand is bigger and give result as boolean expression.)

< Less than (Check which operand is smaller and give result as boolean expression.)

>= Greater than or equal to (Check which operand is greater or equal to each other and give result as boolean expression.)

<= less than equal to (Check which operand is less than or equal to each other and give result as boolean expression.)

== Equal to (Check whether the operand are equal to each other or not and give result as boolean expression.)

!= Not Equal to (Check whether the operand are not equal to each other or not and give result as boolean expression.)

**Code:**

void main()

{

int a = 2;

int b = 3;

// 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");

}

**Cost of one movie ticket is 600 PKR. Write a script to store ticket price in a**

**variable & calculate the cost of buying 5 tickets to a movie.**

**Code**:

void main ()

{

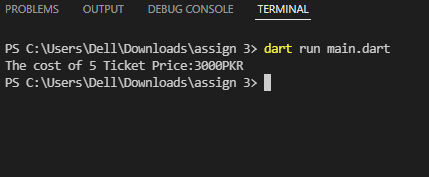
num ticketprice=600;

num total\_ticket=5;

print("The cost of 5 Ticket Price:${ticketprice \* total\_ticket}PKR");

}

**Output:**



**How to get difference of lists in Dart? Problem: Consider you have two lists [1,2,3,4,5,6,7] and [3,5,6,7,9,10]. How would you get the difference as output? E.g. [1, 2, 4].**

Since we are looking for unique elements we could use the difference method of the Set class.

**Code:**

void main ()

{

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

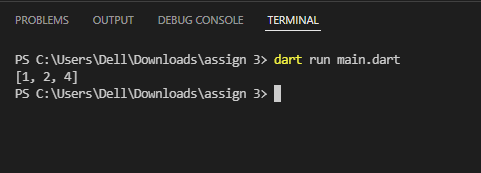
List<int> second = [3,5,6,7,9,10];

List<int> difference = first.toSet().difference(second.toSet()).toList();

print(difference.toString());

}

**Output:**



**What is a difference between these operators “?? And?”**

?? Called also null operator. This operator returns expression on its left, except if it is null, and if so, it returns right expression.

? this is NOT a null-aware operator, but a ternary one. If expression is true it goes with the option1 and if not, with the option2.

**What are the data types supported in Dart? Explain with Examples.**

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

Maps Map -It represents a set of values as key-value pairs

A)**First declare an array and assign the numbers of the table of 7. B). Second declare another array and assign the numbers 1-10. C). Now write down the table of 7 using map.fromiterables method.**

**Code:**

void main ()

{

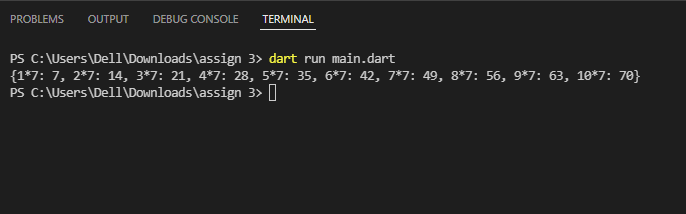
List<String> num7 = ["7","14","21","28","35","42","49","56","63","70"];

List<String> num10 = ["1\*7","2\*7","3\*7","4\*7","5\*7","6\*7","7\*7","8\*7","9\*7","10\*7"];

print(Map.fromIterables(num10,num7));

}

**Output:**



**Write a program that**

**a. Store correct password in a JS variable.**

**b. Asks user to enter his/her password**

**c. Validate the two passwords:**

**d. Check if user has entered password. If not, then give message “Please**

**enter your password”**

**e. Check if both passwords are same. If they are same, show message**

**“Correct! The password you**

**f. entered matches the original password”. Show “Incorrect password”**

**otherwise.**

**Code:**

import 'dart:io';

void main ()

{

var js="12345";

print("Enter your Password:");

var userpass=stdin.readLineSync();

if(userpass==js)

{

print("Correct! The password you entered matches the original password");

}

else if(userpass!.isEmpty)

{

print("Please Enter your Password");

}

else

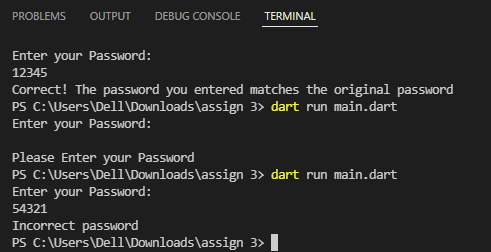
{

print("Incorrect password");

}

}

**Output:**



**Write a program to store 3 student names in an array. Take another array to**

**store score of these three students. Assume that total marks are 500 for each**

**student, display the scores & percentages of students.**

**Code:**

void main()

{

print("--------------Marksheet of Students--------------");

List<dynamic> sn=["1.Ali","2.Ehtisham","3.Zayn"];

List<dynamic> marks=[400,460,350];

var total=500;

var per1=(marks[0]\*100)/total ;

print("Student Name:${sn.elementAt(0)} \n Scored Marks:${marks.elementAt(0)} \n Percentage:$per1%\n");

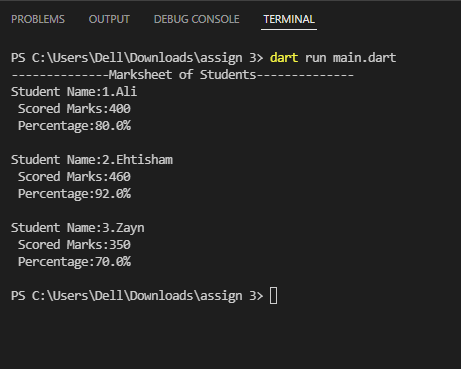
per1=(marks[1]\*100)/total ;

print("Student Name:${sn.elementAt(1)} \n Scored Marks:${marks.elementAt(1)} \n Percentage:$per1%\n");

per1=(marks[2]\*100)/total ;

print("Student Name:${sn.elementAt(2)} \n Scored Marks:${marks.elementAt(2)} \n Percentage:$per1%\n");

}



**Declare 5 legal & 5 illegal variable names.**

//legal

var abc;

var bc\_d;

var acd1;

var ABC;

var aa1c;

//illegal

var 1bcd;

var bc-d;

var if\_else\_vn;

var bc d;

var bc#d;

**Write a program to replace the “Hyder” to “Islam” in the word**

**“Hyderabad” and display the result.**

**Code:**

void main()

{

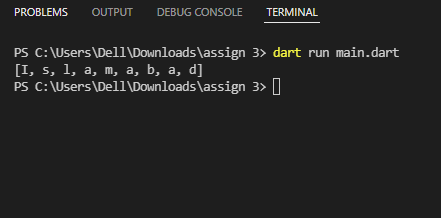
List city=["H","y","d","e","r","a","b","a","d"];

city.replaceRange(0, 5, ["I","s","l","a","m"]);

print(city);

}

**Output:**



**Write a program to generate your K-Electric bill 7. All the amounts should**

**be rounded off to 2 decimal places. Display the following fields:**

**a. Customer Name**

**b. Current Month**

**c. Number of units**

**d. Charges per unit**

**e. Net Amount Payable (within Due Date)**

**f. Late Payment Surcharge**

**g. Gross Amount Payable (after Due Date)**

**Where, Net Amount Payable (within Due Date) = Number of units \* Charges per unit**

**& Gross Amount Payable (after Due Date) = Net Amount + Late Payment Surcharge**

**OR**

**(12) Write a program that shows the message “First fifteen days of the month”**

**if the date is less than 16th of the month else shows “Last days of the**

**month”.**

**Code:**

import 'dart:io';

void main()

{

print("Enter your Name:");

var name=stdin.readLineSync();

print("Enter Current Month of DueDate in Numbers:");

num month=num.parse(stdin.readLineSync()!);

print("Enter Total Units:");

num units=num.parse(stdin.readLineSync()!);

units=num.parse(units.toStringAsFixed(2));

print("Enter Charges per Unit:");

num perunit=num.parse(stdin.readLineSync()!);

perunit=num.parse(perunit.toStringAsFixed(2));

print("Enter Day of Due Date in Numbers:");

num day=num.parse(stdin.readLineSync()!);

num latepay=100;

num netamount=units\*perunit;

netamount=num.parse(netamount.toStringAsFixed(2));

num grosspay=netamount+latepay;

grosspay=num.parse(grosspay.toStringAsFixed(2));

print("----------K-Electric bill-------------");

print("Customer Name:$name");

print("Current Month:$month");

print("Number of Units Used:$units");

print("Charges per Unit:Rs.$perunit");

print("Late Payment Surcharge after DueDate:Rs.$latepay");

if(day<16 && month<13)

{

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

print("Your Net Amount Payable (within Due Date) is:Rs.$netamount");

}

else if(day>31 && month>12)

{

print("Invalid Due Date");

}

else

{

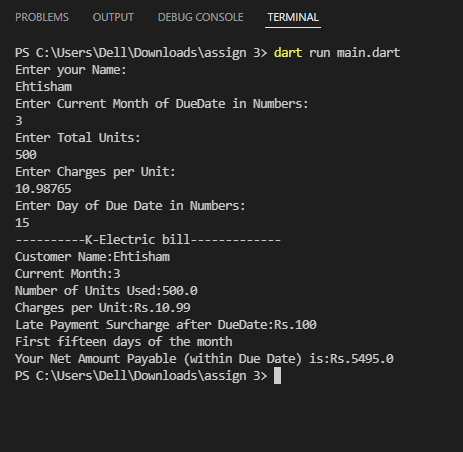
print("Last days of the month");

print("Gross Amount Payable (after Due Date) is:Rs.$grosspay");

}

}

**Output:**



**Find 5 new methods of List and String.**

**List:**

shuffle(): This method re-arranges order of the elements in the given list randomly.

reversed: reversed is a getter which reverses iteration of the list depending upon given list order.

getRange(): This method returns elements from specified range [start] to [end] in same order as in the given list. Note that, start element is inclusive but end element is exclusive.

replaceRange(): This method helps to replace / update some elements of the given list with the new ones. The start and end range need to be provided alongwith the value to be updated in that range.

firstWhere(): This method returns the first element from the list when the given condition is satisfied.

**String:**

startsWith(): This method returns a bool when the string matches the pattern. Optionally, we can provide an index from where the pattern matching should start.

split(): This method is used to split given string into substring which further can be used to retrieve or perform an action on substrings.

replaceFirst(): if we just want to replace first few characters from a given string, we use this method.

isEmpty(): returns true if string is empty.

isNonEmpty(): returns true if string is not empty.