**Assignment 2**

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

The operators are special symbols that are used to carry out certain operations on the operands. The Dart has numerous built-in operators which can be used to carry out different functions, for example, ‘+’ is used to add two operands. Operators are meant to carry operations on one or two operands.

Types:

1. Arithmetic Operators
2. Equality & Rational Operators
3. Logical Operators

**Arithmetic Operators:** **Let variable a holds 20 and variable b holds 10, then −**

|  |  |
| --- | --- |
| Operators & Meanings | Examples |
| +Add  -Subtract  \*Multiply  /Divide  -Divide, returning an integer result  %Get the reminder of an integer division (Modulo)  ++Increment  --Decrement | A + b Returns 30  A – B returns 10  A \* B returns 200  A / B returns 2  A / B returns 2  A % B returns 0  ++ adds 1 to operand  --subtract 1 to operand |

**Equality & Rational Operators:**

**Operator Description Example**

|  |  |  |
| --- | --- | --- |
| **>**  **<**  **>=**  **<=**  **==**  **!=** | **Greater than**  **Lesser than**  **Greater than or equal to**  **Lesser than or equal to**  **Equality**  **Not Equal** | **(A>B) is False**  **(A<B) is True**  **(A>=B) is False**  **(A<=B) is True**  **(A==B) is False**  **(A!=B) is True** |

**Logical Operators:**

**Operator Description Examples**

|  |  |  |
| --- | --- | --- |
| &&  ||  ! | And – The operator returns true only  If all the expressions specified return true  OR- The operator returns true if atleast one of the expressions specified return true  NOT- The operator returns the inverse  of the expressions result.  For E.g.:!(7>5) returns False | (A>10 && B > 10) is False  (A > | | B > 10) is true |

1. What will be the output in variables a, b & result after execution of the following script:
   1. var a = 2, b = 1;

void main() {

var a = 2, b = 1;

int result = (a+b);

print (result);

}

Shape

Description automatically generated with low confidence

* 1. var result = --a - --b + ++b + b--; Explain the output at each stage:
  2. --a;
  3. --a - --b;
  4. --a - --b + ++b;

1. --a - --b + ++b + b--;

Solve:

void main() {

var a =7;

var b = 9;

var result = (--a - --b) + (++b + b--);

print (result);

}

e**. --a;** //answer: 06

* 1. **--a - --b;** //answer: -2
  2. **--a - --b + ++b;** //answer: 07
  3. **--a - --b + ++b + b--;** //answer: 16

1. 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.



void main(){

var Ticketprice = 600;

var total = (Ticketprice\*5);

print("Calculating the amount of 5 Tickets...\n");

print("And your 5 ticket price would be: ${total}");

}

Text

Description automatically generated

1. 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].

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:

Text

Description automatically generated

1. What is a difference between these operators “?? And?”

condition ? expr1 : expr2

If condition is true, then the expression evaluates **expr1** (and returns its value); otherwise, it evaluates and returns the value of **expr2**.

Example:

void main() {

var a = 10;

var res = a > 12 ? "value greater than 10":"value lesser than or equal to 10";

print(res);

}

Output:

value lesser than or equal to 10

expr1 ?? expr2

If **expr1** is non-null, returns its value; otherwise, evaluates and returns the value of **expr2**

Code:

void main() {

var a = null;

var b = 12;

var res = a ?? b;

print(res);

}

Output:

12

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

**Data Types of Darts.**

A data type is a classification of data which tells the compiler or interpreter how the programmer intends to use the data such as

1)**Number:** The number in Dart Programming is the data type that is used to hold the numeric value. Dart numbers can be classified as:

• The int data type is used to represent whole numbers.

• The double data type is used to represent 64-bit floating-point numbers.

• The num type is an inherited data type of the int and double types.

Key Words :- int, double, num

Examples:

Int age = 18;

Double Temprature = 15.5;

Num age = 19;

Num Temprature 18.5;

2)**String:** It used to represent a sequence of characters. It is a sequence of UTF-16 code units. The keyword string is used to represent string literals. String values are embedded in either single or double-quotes.

Key Words :-String

Examples:

String name = “Asad”;

String date = “18 -september -2021”;

3**)Boolean:** It represents Boolean values true and false. The keyword bool is used to represent a Boolean literal in DART.

Key Words :-Bool

Examples:

Bool status = True;

Or

Bool Status = False;

4)**List:** List data type is similar to arrays in other programming languages. A list is used to represent a collection of objects. It is an ordered group of objects.

Examples:

Var list1 = [1, 2,3,4,5]

5)**Map:** The Map object is a key and value pair. Keys and values on a map may be of any type. It is a dynamic collection.

Examples:

void main() {

Map wtd = new Map();

wtd['First'] = 'Welcome';

wtd['Second'] = 'To';

wtd['Third'] = 'Darts';

print(wtd);

}

1. Solve:
   1. First declare an array and assign the numbers of the table of 7.
   2. Second declare another array and assign the numbers 1-10
   3. Now write down the table of 7 using map.fromiterables method.

void main(){

List<int> list = [1, 2, 3,4,5,6,7,8,9,10];

var map = Map<String, int>.fromIterable(list,key: (item) => item.toString(),value: (item) => item \* 7);

print(map);

}

A picture containing text, monitor, screen, television

Description automatically generated

1. Write a program that
   1. Store correct password in a JS variable.
   2. Asks user to enter his/her password
   3. Validate the two passwords:
   4. Check if user has entered password. If not, then give message “Please enter your password”
   5. Check if both passwords are same. If they are same, show message

“Correct! The password you

* 1. entered matches the original password”. Show “Incorrect password” otherwise.

Solve:

void main(){

var js="rizwan10179";

var user\_entered\_password="rizvu007";

if(js == user\_entered\_password){

print("Your password is correct!");

}

else if(js != user\_entered\_password){

print("'${user\_entered\_password}' This is your password,\n and it is incorrect. please provide correct password."); }

else if (!user\_entered\_password){

print("Please Provide Password");

}

}

1. 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.

void main(){

var students=["Rizwan","Rizvu","Arhum"];

var marks=[(400/500)\*100,(300/500)\*100,(350/500)\*100];

print("Student ${students[0]} got ${marks[0]}%");

print("Student ${students[1]} got ${marks[1]}%");

print("Student ${students[2]} got ${marks[2]}%");

}

Output:

Text

Description automatically generated

1. Declare 5 legal & 5 illegal variable names.

void main(){

//legal variabels

var \_first;

var second;

var third\_3;

var fourth\_fourth;

var five5;

//Illegal variabels

var 123;

var 1abc;

var FILLEe;

var @21;

var \*av;

}

1. Write a program to replace the “Hyder” to “Islam” in the word “Hyderabad” and display the result.

void main(){

final myString = 'hyderabad';

final replaced = myString.replaceFirst(RegExp('hyder'), 'islam');

print("Your word is ${myString}");

print("Converted word is ${replaced}");

}

Output:

Text

Description automatically generated

1. 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:



* + 1. Customer Name
    2. Current Month
    3. Number of units
    4. Charges per unit
    5. Net Amount Payable (within Due Date)
    6. Late Payment Surcharge
    7. 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

1. 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:

void main(){

DateTime dateX = DateTime(2020, 9, 14);

DateTime dateY = DateTime(2022, 10, 15);

DateTime dateC = DateTime.now();

if (dateX.isBefore(dateC) && dateY.isAfter(dateC)){

print("dateC is between dateX and dateY");

} else {

print("dateC isn't between dateX and dateY");

}

}

Output:

Text

Description automatically generated