**AQSA SHAHZAD**

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

ANSWER:

**ARITHMETIC OPERATOR:** Arithmetic Operators are the most common operators that are used to perform addition, subtraction, multiplication, divide, etc. Let's take variable a holds 30 and variable b hold 10, then

Addition (+): It adds the left operand to the right operand.

Example: a+b will return 40

Subtraction (-): It subtracts the right operand from the left operand.

Example: a-b will return 20

Multiplication (\*): It multiplies the one operand to another operand.

Example: a\*b will return 300

Modulus (%): It returns a reminder after dividing one operand to another.

Example: a%b will return 0

Division (~/): It divides the first operand by the second operand and returns integer quotient.

Example: a/b will return 3

Increment (++): It increment the value of operand.

Example: ++x

Decrement: It decrement the value of the operand.

Example: --x

**RELATIONAL OPERATOR:** Relational operators or Comparison operators are used to making a comparison between two expressions and operands. The comparison of two expressions returns the Boolean true and false. Suppose a holds 20 and b hold 10 then,

> (greater than): a>b will return TRUE.

< (less than): a<b will return FALSE.

>=(greater than or equal to): a>=b will return TRUE.

<=(less than or equal to): a<=b will return FALSE.

==(is equal to): a==b will return FALSE.

!=(not equal to): a!=b will return TRUE.

**LOGICAL OPERATOR:** The Logical Operators are used to evaluate the expressions and make the decision. Suppose a holds 20 and b holds 15 then,

&& (Logical AND): It returns if all expressions are true.

Example: (A > 15 && B >15) is false

|| (Logical OR): It returns TRUE if any expression is true.

Example: (A > 15 || B >15) is true

! (Logical NOT): !(A>15) is false

**QUESTION: What is a difference between these operators ?? and ?**

ANSWER: The Conditional Operator is same as if-else statement and provides similar functionality as conditional statement. It is the second form of **if-else statement**. It is also identified as **"Ternary Operator"**.

**Syntax 1:** condition ? exp1 : exp2 (If the given condition is TRUE then it returns exp1 otherwise exp2)

**Syntax 2:** exp1 ?? expr2 (If the exp1 is not-null, returns its value, otherwise returns the exp2's value)

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

ANSWER:

**LEGAL VARIABLE:** 1) abc12 2) AGE 3) create\_table 4) xyz99 5) PRACTICE\_1

**ILLEGAL VARIABLE:** 1) &abc 2) \_AGE 3) 1xyz 4) $practice 5) ^var

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

ANSWER: The Dart language supports the following data types: Number, String, Booleans, List, Map.

**NUMBER:** Numbers in Dart are used to represent numeric literals. The Number Dart come in two flavours: Integer, Double.

Integer: Integer values represent non-fractional values, i.e., numeric values without a decimal point. For example, the value "10" is an integer. Integer literals are represented using the **int** keyword.

Double: Dart also supports fractional numeric values i.e. values with decimal points. For example, the value "10.10". The keyword **double** is used to represent floating point literals.

**STRING:** A string is the sequence of the character. If we store the data like - name, address, special character, etc. It is signified by using either single quotes or double quotes.

Example: String abc = “ HELLO WORLD”;

**BOOLEAN:** The Boolean type represents the two values - true and false. If condition satisfied it’s give true otherwise false.

Example: bool isValid = **true**;

**LISTS:** The list is a collection of the ordered objects (value). The elements in the list are separated by the comma enclosed in the square bracket[].

Example: var list = [1,2,3]

**MAP:** The maps type is used to store values in key-value pairs. Each key is associated with its value. The key and value can be any type. In Map, the key must be unique, but a value can occur multiple times. The Map is defined by using curly braces ({}), and comma separates each pair.

Example: var student = {'name': 'abc',  'age':20,  'Branch': 'Computer Science'}

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

ANSWER:

**METHODS OF LISTS:**

**sublist():** This method returns a new list containing elements from index between start and end. Note that end element is *exclusive* while start is *inclusive*.

var Mylist = [1,2,3,4,5];

print(myList.sublist(1,3)); // [2,3]

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

myList.shuffle();

print(‘$myList’); //[5,4,3,1,2]

**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.

print(myList.getRange(1,4)); //(2,3,4)

**take():**This method returns iterable starting from index 0 till the count provided from given list.

var sportsList=[‘cricket’,’tennis’,’football’];

print(sportList.take(2)); //(cricket,tennis)

**skip():**This method ignores the elements starting from index 0 till count  and returns remaining iterable from given list.

print(sportsList.skip(2)); //(football)

**METHODS OF STRING:**

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

String strList=”Hello World”;

print(strList.split(‘ ‘)); // [Welcome, To, Dart, World]

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

**compareTo():**This method is used to compare one input with other and returns an integer.