**Assignment 2**

**Name: Abdullah Khan**

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

**Answer:**

There are various types of operators in Dart and are written as follows.

1. **Arithmetic Operators.**

These are the operators which are used to perform basic arithmetic operations. These operators need two operands to perform operation on. There are six types of arithmetic operators their function and symbol are written as follows.

|  |  |  |
| --- | --- | --- |
| Operator Symbol | Operator Name | Operator Function |
| + | Addition | Adds two operands and can be used for string concatenation. |
| - | Subtraction | Use to subtract two operands. |
| / | Division | Use to divide two operands. |
| ~/ | Base Division | Use to divide two operands but gives answer in integer. |
| % | Modulus | Gives remainder after performing division on operands |
| \* | Multiplication | Use to multiply two operands |

Example:

void main() {

print(5 + 5); //10

print(5 - 6); //-1

print(5 \* 5); //25

print(5 / 5); //1.0

print(5 ~/ 5); //1

print(5 % 12); //5

}

1. **Relational Operators.**

These are the operators which are used to perform relational operations on operands. They are binary operators i.e., they require two operands. They return output in Boolean values i.e., true or false. There are six types of relational operators their function and symbol are written as follows.

|  |  |  |
| --- | --- | --- |
| Operator Symbol | Operator Name | Operator Function |
| > | Greater than | Check which operand is greater. |
| < | Less than | Check which operand is lesser. |
| >= | Greater than or equal to | Checks if operands are equal or greater. |
| <= | Less than or equal to | Checks if operands are equal or lesser. |
| == | Equal to | Checks if the two operands are equal |
| != | Not equal to | Checks if the two operands are not equal |

Example:

void main() {

print(5 == 5); //true

print(5 > 6); //false

print(5 < 4); //false

print(5 != 67); //true

print(5 >= 5); //true

print(5 <= 12); //true

}

1. **Bitwise Operators.**

These are the operators which are used to perform operation on bits of operand. They take input in integer then converts it into binary number and performs operation.

|  |  |  |
| --- | --- | --- |
| Operator Symbol | Operator Name | Operator Function |
| & | Bitwise AND | Perform bitwise AND on operands. |
| | | Bitwise OR | Perform bitwise OR on operands. |
| ^ | Bitwise XOR | Perform bitwise XOR on operands. |
| ~ | Bitwise NOT | Perform bitwise NOT on operand. |
| << | Left Shift | Shift bits from left and add 0 to right. |
| >> | Right Shift | Shift bits from right and add 0 to left. |

Example:

void main() {

print(~5); //-6

print(5 & 6); //4

print(5 | 4); //5

print(5 ^ 67); //70

print(5 >> 5); //0

print(5 << 12); //20400

}

1. **Assignment Operator.**

This type of operator is used to assign values to variables.

|  |  |  |
| --- | --- | --- |
| Operator Symbol | Operator Name | Operator Function |
| = | Equal to | Assigns value to the variable |

Example:

void main() {

int a = 34;

print(a); //34

}

1. **Logical Operator**.

These are the operators which are used to combine two logical expressions.

|  |  |  |
| --- | --- | --- |
| Operator Symbol | Operator Name | Operator Function |
| && | And Operator. | Use to combine two expressions if both are true returns true. |
| || | OR Operator. | Use to combine two expressions if anyone is true returns true. |
| ! | NOT Operator. | Reverses the output i.e., if true returns false and vice versa. |

Example:

void main() {

int a = 34;

int b = 23;

print(a > 12 && b < 100); //true

print(a > 12 || b < 100); //true

print(!(a > 12 && b < 100)); //false

}

**Question 2**: What will be the output in variables a, b & result after execution of the

following script:

a. var a = 2, b = 1;

b. var result = --a - --b + ++b + b--;

Explain the output at each stage:

c. --a;

d. --a - --b;

e. --a - --b + ++b;

f. --a - --b + ++b + b--;

**Answer:**

void main() {

var a = 2, b = 1;

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

print(--a); //0

At this stage pre decrement takes place means a-1-1 which is 0.

print(--b); //-1

At this stage pre decrement takes place means b-1-1 which is -1.

print(b--); //0

At this stage post decrement takes place means b-1 which is 0.

print(--a - --b); //1

As we know the values of -–a and -–b so we’ll just put values; 0-(-1) i.e., 1.

print(--a - --b + ++b); //1

As we know the values of (-–a - -–b) so we’ll just put values; 1+(0) i.e., 1.

print(--a - --b + ++b + b--); //1

As we know the values of (-–a - -–b + ++b) so we’ll just put values; 1+(0) i.e., 1.

}

Question 3: 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.

**Answer:**

void main() {

int ticket\_quantity = 5;

int ticket\_price = 600;

//for 5 tickets

int total = ticket\_quantity \* ticket\_price;

print("The price of 5 tickets = ${total}");

}

**Question 4**: 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].

**Answer:**

In dart the difference of two list can be obtained by using two method which are predefined in dart.   
The methods are “remove where” and “contains”. Remove where method takes a parameter and checks the list and where he finds the same value as parameter it removes it. Contain method is used to se if a list contains these values so, we designed a logic in a way that remove where a list contain as element as other list

**Code:**

void main() {

List<int> l1 = [1, 2, 3, 4, 5, 6, 7];

List<int> l2 = [3, 5, 6, 7, 9, 10];

l1.removeWhere((element) => l2.contains(element));

print(l1);

}

**Question 5**: What is a difference between these operators “?? And?”

**Answer:**

These are called conditional operators. These are operators we use when we are performing comparison on operands.

|  |  |
| --- | --- |
| Operator Symbol | Operator Function |
| ?? | It is like if – else. If comparison is true returns expression 1 else returns expression 2 |
| ? | Checks if condition is non-null then returns value else returns expression 2. |

Example:

void main() {

var a;

var b = 56;

var g = 10;

var h = 12;

var c = a ?? b;

print(c);

Returns 56 because is null

//If the given condition is TRUE then it returns exp1 otherwise exp2.

var d = g > h ? "value greater than 10" : "value lesser than 12";

print(d);

}

**Question 6:** What are the data types supported in Dart? Explain with Examples.

**Answer:**

There are many data types in Dart some of them are listed below.

1. **Number.**

Numbers in dart can be declared by 3 different keywords.

|  |  |
| --- | --- |
| Keyword | Function |
| int | Used to declare non-floating-point numbers. |
| double | Used to declare floating point numbers. |
| num | Used to declare both int and double numbers. |

Example:  
void main() {

int b = 56;

double g = 10.2;

num h = b \* g;

print(h); //571.199

}

1. **Strings.**

Strings in dart can be declared by only 1 keyword that is “String”.

Example

void main() {

String b = "Abdullah";

print(b); //Abdullah

}

1. **Booleans.**

Booleans in dart can be declared by only 1 keyword that is “bool”. They are used to declare true or false value or when we want output in true or false.

Example:

void main() {

int b = 56;

double g = 10.2;

bool h = b > g;

print(h); //true

}

1. **List.**

Lists are just like an array but you can specify if you want only string, int or combination of both. For list that contains only integers we use List<int>, for floating point values List<double>, for strings List<String> and for combination of both List<dynamic>.

Example:  
void main() {

List<int> h = [1, 3, 4, 5];

List<double> g = [12.3, 121, 11.11, 11.3];

List<String> f = ["Abdullah", "Khan"];

List<dynamic> i = ["Abdullah", 12, 12.1];

}

1. **Map.**

Maps are a bit same as list but here for every value there is a key by which is can be called. They are declared by keyword “Map”.

Example:  
void main() {

Map h = {1: 12, 3: 23, 4: 232, 5: 232};

}

Question 7: Solve:

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.

**Answer:**

void main() {

List<int> table = [7, 14, 21, 28, 35, 42, 49, 56, 63, 70];

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

Map times\_Table = Map.fromIterables(times, table);

print(times\_Table);

}

**Question 8:** 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.

**Answer:**

import 'dart:io';

void main() {

print("Enter your Password?");

String password = stdin.readLineSync()!;

if (password == "Saad" || password == "Asad") {

print("You Entered Correct Password");

} else {

print("You Entered Wrong password");

}

}

**Question 9**: 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.

**Answer:**

void main() {

List<String> name = ["Humza", "Abdullah", "Anas"];

List<double> score = [498, 489, 490];

int total = 500;

List<double> n = [

(score[0] / total) \* 100,

(score[1] / total) \* 100,

(score[2] / total) \* 100

];

print("${name[0]} has got ${score[0]} marks and ${n[0]}%\n"

"${name[1]} has got ${score[1]} marks and ${n[1]}%\n"

"${name[2]} has got ${score[2]} marks and ${n[2]}%\n");

}

**Question 10**: Declare 5 legal & 5 illegal variable names.

**Answer:**

1. **Legal Variable:**

There are certain conditions when naming variables in dart. If your variable name meets them then your variable is called legal variable. Following are some examples.

1. int name = 123;
2. String n1 = “Abdullah”;
3. double number = 33.1;
4. List<int> score = [1,2,3,4];
5. List<dynamic> m\_n = [“wee”,420];
6. **Illegal Variable**:

There are certain conditions when naming variables in dart. If your variable name doesn’t meets them then your variable is called legal variable. Following are some examples.

1. int 1name = 12;
2. String \_h = “Abdullah”;
3. String 12 = “Khan”;
4. String print = “Hellow”;
5. List<dynamic> m-n = [“wee”,420];

**Question 11**: Write a program to replace the “Hyder” to “Islam” in the word

“Hyderabad” and display the result.

**Answer:**

void main() {

String n = "Hyderabad";

String m = "Islam" + n.substring(5, 9);

print(m);

}

**Question 12**: 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

**Answer:**

void main() {

String customerName = "Abdullah Khan";

String currentMonth = "October";

double totalUnits = 344.3;

double unitRate = 9.15;

double netAmount = unitRate \* totalUnits;

String n = netAmount.toStringAsFixed(2);

double lateSurcharge = 500;

double grossAmount = netAmount + lateSurcharge;

String g = grossAmount.toStringAsFixed(2);

print(

"Consumer name:$customerName\nBilling Month:$currentMonth\nUnits consumed:$totalUnits\n"

"Charge per unit :$unitRate\nAmount payable within due date :$n\n"

"Late Payment Surcharge:$lateSurcharge\nAmount payable after due date:$g");

}