///Dart 1st Assignment///

***Name:-ABHIJEET KUMAR***

***ROLL NO. :-00114504420***

void main() {

print("Hello World");

print("This is my first application");

}

**Operators in Dart**

**1) Multiplicative :- \* / % ~/**

**eg:- 1**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1\*num2;

print("${result}");

}

**eg:- 2**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1/num2;

print("${result}");

}

**eg:- 3**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1%num2;

print("${result}");

}

**eg:- 4**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1~/num2;

print("${result}");

}

**2) Additive :- + -**

**eg:- 1**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1+num2;

print("${result}");

}

**eg:- 2**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1-num2;

print("${result}");

}

**3) Unary postfix :- expr++ expr--**

**eg:- 1**

void main() {

var num1=10;

var result=0;

result=num1++

print("${result}");

}

**eg:- 2**

void main() {

var num2=20;

var result=0;

result=num2--;

print("${result}");

}

**4) Unary prefix :- -expr !expr ~expr ++expr --expr await**

**eg:-1**

void main() {

var num2=20;

var result=0;

result=--num2;

print("${result}");

}

**eg:-2**

void main() {

var num1=20;

var result=0;

result=++num1;

print("${result}");

}

**eg:-3**

void main() {

var num1=10;

var num2=-20;

var result=0;

result= -num2+num1;

print("${result}");

}

**eg:-4**

void main() {

var num1=true;

if(!num1==false){

print("hello");

}

}

**eg:-5**

void main() {

var num1=20;

if(~num1==20){

print("hello");

}

}

**eg:-6**

Future hello() async {

print('Hi,there');

}

void main() async{

await hello();

print('Enjoy Flutter and Dart');

}

**5) Shift :- << >> >>>**

**eg:-1**

void main() {

var num1=001001;

print(num1 >> 3);

}

**eg:-2**

void main() {

var num1=001001;

print(num1 << 3);

}

**6) Bitwise AND :- &**

**eg:-1**

void main() {

var num1=5;

var num2=8;

print(num1 & num2);

}

**7) Bitwise XOR :- ^**

**eg:-1**

void main() {

var num1=5;

var num2=8;

print(num1 ^ num2);

}

**8) Bitwise OR :- |**

**eg:-1**

void main() {

var num1=5;

var num2=8;

print(num1|num2);

}

**9) Relational and type test :- >= > <= < is is! as**

**eg:-1**

void main() {

var num1=12;

var num2=8;

if(num1>=num2){

print("greater than or eqaul");

}

}

**eg:-2**

void main() {

var num1=12;

var num2=8;

if(num1>num2){

print("greater than");

}

}

**eg:-3**

void main() {

var num1=12;

var num2=12;

if(num1<=num2){

print("less than or eqaul");

}

}

**eg:-4**

void main() {

var num1=4;

var num2=8;

if(num1<num2){

print("less than");

}

}

**eg:-5,6**

void main()

{

String a = 'hello';

double b = 8.8;

print(a is String);

print(b is !int);

}

**10) Equality :- == !=**

**eg:-1**

void main() {

var num1=4;

var num2=8;

print(num1==num2);

}

**eg:-2**

void main() {

var num1=4;

var num2=8;

print(num1!=num2);

}

**11) Logical AND :- &&**

**eg:-1**

void main() {

var num1=4;

var num2=8;

if(num2>=num1 && num1%2==0)

{

print("Logical AND");

}

}

**12) Logical OR :- ||**

**eg:-1**

void main() {

var num1=4;

var num2=8;

if(num2>=num1 || num1%2==0)

{

print("Logical AND");

}

}

**13) Conditional :- condition ? expr2 : expr3**

**eg:-1**

void main()

{

var a = 5;

var b = 7;

var c = (a < 10) ? "Statement is Correct : "Statement is Wrong";

print(c);

}

**14) if null :- ??**

**eg:-1**

void main()

{

int n;

var d = n ? ? "n has Null value";

print(d);

n = 10;

d = n ? ? "n has Null value";

print(d);

}

**15) Cascade:- .. ?..**

**eg:-1**

class hello {

var a;

var b;

void set(x, y)

{

this.a = x;

this.b = y;

}

void add()

{

var z = this.a + this.b;

print(z);

}

}

void main()

{

// Creating objects of class GFG

hello h1 = new hello();

GFG h2 = new hello();

// Without using Cascade Notation

h1.set(1, 2);

h1.add();

// Using Cascade Notation

h2..set(3, 4)

..add();

}

**16) Assignment :- = \*= /= += -= &= ^=**

**eg:-1**

void main()

{

var num1=10;

var num2=11;

var result=0;

result+=num1+num2; // result\*=num1\*num2;//result/=num1;

print(result);

}