Star Bawa

bCA 5 | 720058

DART FILE

submitted to amit srivastava

**//Q.1 Sort a list and print them according to length**

import 'dart:io';

import 'dart:math';

void main() {

var list = <int>[5, 4, 3, 2, 1];

list.sort();

print(list);

var rev = list.reversed;

print(rev);

var strlist = <String>[

'falcon',

'order',

'war',

'sky',

'ocean',

'blue',

'cloud',

'boy',

'by',

'raven',

'station',

'batallion'

];

strlist.sort((e1, e2) => e1.length.compareTo(e2.length));

strlist.forEach(print);

// print method in forEach helps in printing the total list;

}

[1, 2, 3, 4, 5]

(5, 4, 3, 2, 1)

by

war

sky

boy

blue

order

ocean

cloud

raven

falcon

station

batallion

**//Q2. print the unique number**

import "dart:io";

bool isUniqueNum(int num) {

var list = [];

int count = 0;

for (int i = num; i > 0; i = i ~/ 10) {

count++;

}

for (int i = num; i > 0; i = i ~/ 10) {

int d = i % 10;

if (list.contains(d)) {

return false;

} else

list.add(d);

}

return true;

}

void main() {

// print("Enter a number ");

stdout.writeln("Enter a number");

var num1 = stdin.readLineSync();

int num = int.parse(num1 ?? '0');

bool ch = isUniqueNum(num);

print(ch);

}

Output->

Enter a number

123

True

**//Q3 .Wap to show operator overloading**

import 'dart:io';

class OperatorOverloading {

double a = 101;

void input() {

print("Enter a number");

a = double.parse(stdin.readLineSync()!);

}

void display() {

print('$a \n');

}

OperatorOverloading operator +(OperatorOverloading b) {

OperatorOverloading ob = new OperatorOverloading();

ob.a = b.a + a;

return ob;

}

}

void main() {

OperatorOverloading ob = new OperatorOverloading();

OperatorOverloading ob2 = new OperatorOverloading();

OperatorOverloading ob3 = new OperatorOverloading();

ob3 = ob2 + ob;

ob3.display();

}

Output-> 202.0

**// Q4 .Wap to show constructor in dart**

class Abc {

var a, b, c;

Abc() {

a = 100;

b = 200;

c = 300;

}

Abc.namedconst(int x, int y, int z) {

//Abc named Constructor in java

a = x;

b = y;

c = z;

}

void show() {

print("$a,$b,$c");

}

}

void main() {

var p = new Abc();

// p.a = 100;//

// p.b = 200;//

// p.c = 300;//

p.show(); //default constructor

var q = new Abc.namedconst(100, 2, 4);

q.show(); //Named Constructor

}

**Q.5 Wap to EveryList method in dart**

// import 'dart-sdk/lib/html/dart2js/html\_dart2js.dart';

class User {

String firstName;

String LastName;

bool married;

User(this.firstName, this.LastName, this.married); //constructor paramaterised

bool get isMarried => this.married;

// String get firstName => this.LastName;

@override

String toString() => "${this.firstName} ${this.married} ${this.married}";

}

void main() {

var users = <User>[

User('John', 'Doe', true),

User('Jane', 'Doe', true),

User('Peter', 'Smith', false),

User('Roger', 'Roe', true),

User('Martin', 'Fonda', false),

];

var allmarried = users.every((e) => e.isMarried);

if (allmarried)

print('All users are married');

else

print('not all users are married');

users.sort((e1, e2) => e2.LastName.compareTo(e1.LastName));

users.forEach(print);

}

**Q6. Wap to implement List filter in dart**

void main() {

var vals = <int>[1, 2, 3, 4, 5];

print(vals);

var words = <String>["Wolf", "sky", "length", "cloud"];

print(words);

var w3 = words.where((e) => e.length == 3);

//words.where((element to be printed)=> condition)

print(w3);

var e1 = words.firstWhere((e1) => e1.startsWith('w'));

print(e1);

var e2 = words.lastWhere((element) => element.startsWith('w'));

print(e2);

var e3 = words.where((e) => e.length == 3);

print(e3);

}

1, 2, 3, 4, 5]

[Wolf, sky, length, cloud]

(sky)

**Q7. Wap to implement dynamic method in dart.**

void main() {

print(square(10.01231));

showOutput("hello world");

print(division(10, 2));

}

dynamic square(var num) {

return num \* num;

}

//the dynamic function in dart is used to return the

//dynamic values such that if we pass an integer we will reciecve

// an integer if we pass a double we recive a double

void showOutput(var msg) {

print(msg);

}

dynamic division(var a, var b) {

return a / b;

}

Output->

64

**Q.8 Wap To show the use of inheritance in dart**

class employee {

String ename = "", eid = "";

void get() {

ename = "Abc";

eid = "100";

}

void put() {

print("Employee name is $ename");

print("Employee id is $eid");

}

}

class data extends employee {

int sal = 0;

void input() {

sal = 1000;

super.get();

}

void output() {

print("output col");

super.put();

}

}

void main() {

data d = new data();

d.input();

d.output();

}

output col

Employee name is Abc

Employee id is 100

**Q9. Explain Exception handling through a piece of program**

//Exception handling

int mustGreaterThanZero(int val) {

  if (val <= 0) throw Exception('value must be greater than zero ');

  return 0;

}

void letVerifyTheValue(var val) {

  var valueVerification;

  try {

    valueVerification = mustGreaterThanZero(val);

  } catch (e) {

    print(e);

  } finally {

    if (valueVerification == null)

      print("Value is not accepted");

    else

      print("Value is valid $val");

  }

}

void main() {

  letVerifyTheValue(10);

}

Enter the number : 10

Value is valid 10;

Q.10 Wap to check a palindrome number in dart

import 'dart:io';

// import 'dart-sdk/lib/ffi/ffi.dart';

void main() {

  print("Enter a number");

  // var n = stdin.readLineSync();

  var n = "123321";

  var n1 = n.split('');

  var rev = "";

  for (var i in n1) {

    rev = i + rev;

  }

  if (rev == n)

    print("Palindrome");

  else

    print("not palindrome");

}

Output->

Palindrome