صلاح حسن السيد السيد حسين

Q1
code ::

```dart
void main()
{
  String myTeam = "Real Madrid" ;
  int numberOfMember = 11 ;
  int champions = 26 ;

  print("my Team is $myTeam , has $numberOfMember players and has $champions") ;

  // output??
  // my Team is Real Madried , has 11 players and has 26
}
```

Q2
Code::

```dart
void main()
{
  var x = 10;
  x = 20;
  print("$x") ;      // 10

  // x = "hello";   // error
  // مينفعش اغير النوع لان اصلا هو كان من نوع عدد صحيح انت حالا عايزه يكون نوعه نص او... ف هيديك ايرور
}
```

Q3
Code::

```dart
void main()
{
  dynamic x = 10 ;
  x = 20.3 ;
  x = true ;
  x = "salah" ;
  print("$x") ;       // salah

}
```

Q4

مقارنه بين ال

var , dynamic??

الاجابة هي

يحد القيمة مره واحده يعني مثلا لو انت عملت متغير من نوع var ==>

خلاص مينفعش تغير نوعه تاني الي اي نوع اخر int

علي عكس ال dynamic

لو عملت متغير من نوع int

تعرف تغيره الي اي نوع تاني وليكن مثلا

string , bool , double ,….

و بالتالي يكون استخدم

افضل var

و ذلك للامان والحفاظ علي الداتا لان الداتا لا تتغير بعد ذلك حيث يتم تحديدها مره واحده فقط

علي عكس الdynamic

ممكن يتم التلاعب بيها و تغير نوعها وبالتالي يكون الامان اقل

==========

Q5

اذكر الخطأ و صلح هذا الخطأ اذا كان يوجد؟

Code::

```
void main()
{
  final age = 25 ;
  age = 30 ;
  print("age is $age") ;
}
```

يوجد خطا ف البرنامج

Error

و ذلك لانه

final

تعني ان القيمه تتعرف مره واحده فقط و لا يجوز اسناد القيمه ليها مره تانيه

ف بكدا بيدي ايرور

كمان

final

تتحدد وقت التشغيل

To fix error

you should use

var instead of final

Q6

االفرق بين
const , final

قيمة ثابته لا تتغير==>const
يتم تحديدها اثناء وقت الترجمة
compile time

قيمة تتعين مره واحده فقط يتم تحديدها اثناء <== final
وقت التشغيل
run time
لا يمكن تغيرها بعد ذلك

Code::

```dart
void main() {
  const pi = 3.14;
  // pi = 3.15;           // Error
  print(pi); // 3.14
}
```

```dart
void main()
{
  final time = DateTime.now(); // true
  // const time = DateTime.now(); // error
  print("$time");
  print(time);
}
```

```dart
void main() {
  final age = 25;
  // age = 30;         // Error
  print(age);
}
```

Q7
List
Code::

```dart
void main()
{
  List<dynamic> l1 = ["Salah", 20, [50, 11, 30 , 76 , 82]];

  l1.add("Ahmed");

  l1.remove(50);

  l1[1] = "Ali";

  l1.add([10, 5, 7]);

  List l2 = l1[2];
  l2.sort();

  print(l1);
  // [Salah, Ali, [11, 30, 50, 76, 82], Ahmed, [10, 5, 7]]
}
```

Q8

Map
Code::

```dart
void main()
{
  Map<String, int> mp =
  {
    'Ali': 3000,
    'Sara': 4000,
    'Ahmed': 3500,
    'Laila': 4500,
  };

  mp['Khaled'] = 5555;

  if (mp.containsKey('Ali'))
  {
    print(mp['Ali']);
  }

  mp.remove('Ahmed');

  print(mp.keys.toList());
  print(mp.values.toList());

  Map<String, double> m2 = {};
  mp.forEach((k, v) {
    m2[k] = v * 1.10;
  });

  print(m2);
}

// output
//============================
// 3000
// [Ali, Sara, Laila, Khaled]
// [3000, 4000, 4500, 5555]
// {Ali: 3300.0000000000005, Sara: 4400.0, Laila: 4950.0, Khaled: 6110.500000000001}
```

Q9
المقارنة بين
Set && Map

Set ==> لا تكرر القيم
مثلا
لو عندي
Set = {1, 2 , 2 , 8 , 4 , 1 , 2 , 5 }
و تريد الطباعة
لايتم طباعهالقيم المكرره
لانها لا تكرر القيم

ولكن
Map
تتميز بوجود القيمة و المفتاح
Key , Value

Code::

```dart
void main()
{
  Map<String, int> mp = {'A': 1, 'B': 2};
  Set<int> st = {1, 2, 3};

  print(mp['A']);              // 1
  print(st.contains(2));      // true
}
```

Q10
Switch

Code::

```dart
import 'dart:io';
void main()
{
  int x = int.parse(stdin.readLineSync()!);
  switch (x)
  {
    case 1:
      print("Sunday");
      break;
    case 2:
      print("Monday");
      break;
    case 3:
      print("Tuesday");
      break;
    case 4:
      print("Wednesday");
      break;
    case 5:
      print("Thursday");
      break;
    case 6:
      print("Friday");
      break;
    case 7:
      print("Saturday");
      break;
    default:
      print("رقم غير صالح");
  }
}
// 1 ==> Sunday ,,, 2 ==> Monday ,,, 3 ==> Tuesday ....
```

Q11
Multiplictian
جدول الضرب

```dart
import 'dart:io';

void main()
{
  int x = int.parse(stdin.readLineSync()!);
  print("============================") ;
  for (int i = 1; i <= 12; i++)
  {
    print("$x x $i = ${x * i}");
  }
}
// output

// 128
// ============================
// 128 x 1 = 128
// 128 x 2 = 256
// 128 x 3 = 384
// 128 x 4 = 512
// 128 x 5 = 640
// 128 x 6 = 768
// 128 x 7 = 896
// 128 x 8 = 1024
// 128 x 9 = 1152
// 128 x 10 = 1280
// 128 x 11 = 1408
// 128 x 12 = 1536
```

Q12
OOP

Code::

```
void main()
{
  Counter c = Counter(5);
  c.inc();          // 6
  c.dec();         // 5
  c.reset();       // 0
  print(c.x);      // 0
}

class Counter
{
  int x;

  Counter(this.x);

  void inc()
  {
    x++;
  }

  void dec()
  {
    x--;
  }

  void reset()
  {
    x = 0;
  }
}
```

==========
OOP ===>Book
==========

```dart
class Book
{
  String title;
  String author;
  String genre;
  int pages;
  int availableCopies;
  double rating;
  bool onLoan;

  Book(
    this.title,
    this.author,
    this.genre,
    this.pages,
    this.availableCopies,
    this.rating,
    this.onLoan,
  );

  void borrow()
  {
    if (availableCopies > 0 && onLoan == false)
    {
      availableCopies--;
      onLoan = true;
      print("Done the book is borrow successfully");
    }
    else
    {
      print("No Books is available Now");
    }
  }

  void returnBook()
  {
    availableCopies++;
    onLoan = false;
    print(" Done the book is return successfull");
  }

  void printBook()
  {
    print("$title");
    print("$author");
    print("$genre");
    print("$pages");
    print("$availableCopies");
    print("$rating");
    print("$onLoan");
    print("============================") ;
  }
}
```

```dart
void main()
{
  Book b1 = Book("Data Structure", "Salah", "Programming", 210, 3, 4.5, false);
  Book b2 = Book("Flutter Book", "Sara", "Technology", 370, 2, 4.8, false);
  Book b3 = Book("Clean Code", "Ahmed", "Software", 420, 0, 4.9, false);

  b1.printBook();
  b1.borrow();
  b1.printBook();
  b1.returnBook();
  b1.printBook();

  b2.borrow();
  b2.borrow();

  b3.borrow();
}
```

Output

```
Data Structure
Salah
Programming
210
3
4.5
false
==============================
Done the book is borrow successfully
Data Structure
Salah
Programming
210
2
4.5
true
==============================
 Done the book is return successfull
Data Structure
Salah
Programming
210
3
4.5
false
==============================
Done the book is borrow successfully
No Books is available Now
No Books is available Now
```

Error! Number cannot be represented in specified format.

Big-O Notation

دي طريقه عشان اعرف بيها مدي كفاءة الخوارزمية
فيه نوعين
Time Complexity
هو الوقت الي بيستغرقه الخوارزميه ف التنفيذ
لكن
Space Complexity
هو المساحه التي ياخذها الخوارزميه من الميموري

طب انا يهمني اي؟؟
اكيد يهمني ال
Time
لان معظم الاجهزه حالا بقت مساحته كبيره ف اكيد هبص لحته الوقت اكتر من المساحة

المهم

انواع ال
Big O Notation
اشهر الانواع
Big O(1)

مثال لو انا عندي ليست و عايز اول عنصر ف القايمه ف اكيد ده وقت ثابت
مش هحتاج اخد منه تايم
ف بيكون هو
O(1)

Big O(n)
لما يكون الزمن خطي معاك ليست و بتلف علي الليست كلها ف اكيد هتمشي عدد خطوة n

Big O(n^2)
لما يكون عندك
Nested For
For جوا For

O(log n)

تستخدمم اكثر ف خوارزمية البحث
Binary Search