Dart Programming Quiz

Part 1: Multiple Choice Questions (QCM)

1. What is the correct way to declare a constant value in Dart?

   A) `var pi = 3.14;`

   B) `final pi = 3.14;`

   C) `const pi = 3.14;`

   D) `let pi = 3.14;`

2. Which of the following is NOT a valid Dart data type?

   A) `int`

   B) `double`

   C) `decimal`

   D) `String`

3. What will the following code print?

```dart
void main() {
  var a = 5;
  var b = 2;
  print(a ~/ b);
}
```

   A) `2.5`

   B) `2`

   C) `3`

   D) `Error`

4. How do you define a named parameter in Dart?

   A) `void greet(String name) {}`

   B) `void greet({String name}) {}`

   C) `void greet([String name]) {}`

   D) `void greet(*String name*) {}`

5. Which of the following is correct for creating a List in Dart?

   A) `var list = [1, 2, 3];`

   B) `var list = (1, 2, 3);`

   C) `var list = {1, 2, 3};`

   D) `var list = <1, 2, 3>;`

6. What is the output of this code?

```
void main() {
  var name;
  print(name ?? "Guest");
}
```

   A) `null`
   B) `Guest`
   C) `""` (empty string)
   D) `Error`

7. What is the default value of an uninitialized `int` variable in Dart?

   A) 0
   B) `null`
   C) undefined
   D) 1

8. Which of the following is true about `final` in Dart?

   A) Must be assigned at compile time
   B) Can only be assigned once at runtime
   C) Can be reassigned multiple times
   D) Must be mutable

9. Which operator is used for **type casting** in Dart?

   A) `as`
   B) `is`
   C) `is!`
   D) `->`

10. What is the output?

```
void main() {
  print(3 ~/ 2);
}
```

   A) 1.5
   B) 1
   C) 2
   D) Error

11. How do you declare an optional positional parameter?

   A) `void f(int x)`
   B) `void f([int x])`
   C) `void f({int x})`
   D) `void f(*int x*)`

12. Which collection in Dart is **unordered and does not allow duplicates**?

    A) `List`
    B) `Map`
    C) `Set`
    D) `Queue`

13. How do you create a Map in Dart?

    A) `var m = {1, 2}`
    B) `var m = {1: 'one', 2: 'two'}`
    C) `var m = [1: 'one', 2: 'two']`
    D) `var m = Map(1,2)`

14. How do you check if a variable is null?

    A) `x == null`
    B) `x ? null`
    C) `x ?? null`
    D) `x.isNull()`

15. Which keyword is used to create an asynchronous function?

    A) `async`
    B) `await`
    C) `future`
    D) `yield`

16. What is the output?

```dart
void main() {
  List<int> l = [1,2,3];
  print(l.contains(2));
}
```

    A) 0
    B) true
    C) 2
    D) false

17. How do you define a **constant constructor** in a class?

    A) `const ClassName()`
    B) `ClassName.const()`
    C) `final ClassName()`
    D) `static ClassName()`

18. Which of the following is a **null-aware operator** in Dart?

    A) `?.`
    B) `!`

C) `==`

D) `=>`

19. Which of these is valid string interpolation?

A) `"Hello $name"`

B) `"Hello {name}"`

C) `"Hello + name"`

D) `"Hello #name"`

20. How do you define a **getter** in Dart?

A) `int get age => _age;`

B) `int age() => _age;`

C) `get int age => _age;`

D) `getter int age => _age;`

21. What is the type of `var x = 3.14;`?

A) `int`

B) `double`

C) `num`

D) `var`

22. How do you catch exceptions in Dart?

A) `try { ... } catch(e) { ... }`

B) `try { ... } except(e) { ... }`

C) `try { ... } error(e) { ... }`

D) `catch { ... }`

23. What is the output?

```
void main() {
  var l = [1,2,3];
  l.add(4);
  print(l.length);
}
```

A) 3

B) 4

C) Error

D) 0

24. Which keyword makes a class **abstract**?

A) `abstract`

B) `interface`

C) `final`

D) `virtual`

25. How do you mark a parameter as **required** in a named parameter?

    A) `void f({required int x}) {}`
    B) `void f([required int x]) {}`
    C) `void f(int x!)  {}`
    D) `void f(required int x) {}`

26. How do you call a superclass constructor?

    A) `super()`
    B) `base()`
    C) `parent()`
    D) `this()`

27. How do you create a **constant list**?

    A) `var l = [1,2,3];`
    B) `final l = [1,2,3];`
    C) `const l = [1,2,3];`
    D) `List l = [1,2,3];`

28. What is the type of `List<int>`?

    A) `dynamic`
    B) `generic list of int`
    C) `Set`
    D) `Map`

29. Which function executes **after a future completes**?

    A) `then()`
    B) `catch()`
    C) `async()`
    D) `await()`

30. Which of the following **iterates a map** correctly?

    A) `for(var k in map) {}`
    B) `for(var e in map.entries) {}`
    C) `map.foreach((k,v){});`
    D) `map.forin()`

31. What is printed?

```dart
void main() {
  var s = "Dart";
  print(s.substring(1,3));
}
```

    A) `Da`
    B) `ar`

C) `rt`

D) `Dar`

32. How do you make a variable **late-initialized**?

A) `late int x;`

B) `final x;`

C) `var x;`

D) `int?  x;`

33. Which statement is true about **extension methods**?

A) Can add methods to existing classes

B) Can override private fields

C) Can change the original class

D) Only works with List

34. Which keyword is used to **pause a function until a Future completes**?

A) `await`

B) `async`

C) `then`

D) `yield`

35. Which of the following **creates a set of integers**?

A) `var s = {1,2,3};`

B) `var s = [1,2,3];`

C) `var s = (1,2,3);`

D) `var s = <1,2,3>;`

36. Which of these is a valid **cascade operator usage**?

A) `myList..add(1)..add(2);`

B) `myList.>add(1).>add(2);`

C) `myList::add(1)::add(2);`

D) `myList**add(1)**add(2);`

Part 2: Exercises

E1. Write a Dart function `factorial(int n)` that returns the factorial of `n` using recursion.

E2. Create a Dart program that reads a list of integers and prints the **largest number**.

E3. Write a Dart class `Person` with `name` and `age` fields, and a method `introduce()` that prints: `"Hi, my name is <name> and I am <age> years old."`.

E4. Implement a Dart function `isPalindrome(String s)` that checks if a string is a palindrome (reads the same backward as forward).

E5. Write a Dart function that **reverses a List** of integers.

E6. Create a Dart program that counts the **number of vowels** in a string.

E7. Implement a Dart class `Rectangle` with `width` and `height`, and a method `area()` that returns the area.

E8. Write a Dart function `fibonacci(int n)` that returns the nth Fibonacci number **recursively**.

E9. Create a Dart program that **sorts a list of strings alphabetically**.

E10. Write a Dart class `BankAccount` with methods `deposit(amount)` and `withdraw(amount)` and a `balance` field.

E11. Implement a Dart function that **removes duplicates from a list**.

E12. Write a Dart program that reads a list of numbers and prints the **average**.

E13. Create a Dart class `Car` with `brand` and `year` fields, and override the `toString()` method.

E14. Write a Dart function that **checks if a number is prime**.