Name: Tip Dalin

Group : 3

Assignment

1. Type Inference

EXPLAIN: Explain how Dart infers the type of a variable.

+ Dart uses type inference to automatically determine the type of a variable based on the value that we assigned to it.

When you declare a variable using the 'var' keyword without explicitly specifying its type,

Dart infers the type from the initial value assigned to it.

```
void main() {
  // Dart infers the type as int based on the value assigned
  int number = 10;
  // Explicitly declaring the type as String
  String name = "Dalin!";
  // Print both variables to show their values
  print("The value of 'number' is: $number");
  print("The value of 'name' is: $name");
```

2.Nullable and Non Nullable Variable.

EXPLAIN : Explain nullable vs non-nullable variables.

-Nullable variables are declared using the ? syntax after the type. These variables can either hold a value of the specified type or null.

-Non Nullable variables in dart by default all variables are non nullable to ensures that no errors at compile time.

EXPLAIN : When is it useful to have nullable variables?

-Nullable variables are useful such as in optional values, initial state and external data.

```
// Declare a nullable integer variable and assign it a null value
int? nullableInt = null;
// Declare a non-nullable integer variable and assign it a value
```

int nonNullableInt = 5;

// Assign a new value to the nullable variable

nullableInt = 10;

3.Final and const

EXPLAIN : Describe the difference between final and const.

-Final:

-Final variable can be assigned only once and cannot be reassigned.

-The value of final variable is initialized at runtime.

-Const:

-A const variable is a compile-time constant, meaning its value must be known and assigned at compile time.

-Can only assign simple, known values (like numbers, strings, or literal values) to const variables.

// Declare a final variable and assign it the current date and time

final currentDateTime = DateTime.now();

/* Can you declare this variable as const? Why?

-No, we cannot declare `currentDateTime` as const

because it depends on runtime (DateTime.now() is evaluated at runtime).

*/

// Declare a const variable with a integer value

const num = 3;

// Can you reassign the value of this final variable? Why? ```

// -No, we cannot reassign a final variable once it is assigned. The following line would cause an error.

4. Strings:

//Declare two strings: firstName and lastName and an integer:age

String firstName = 'Dalin';

String lastName = 'Tip';

int age = 20;

```
// Concatenate the 2 strings and the age
   String fullInfo = 'Name: $firstName $lastName, Age: $age';
// Print result
   print(fullInfo);

//Lists:

// Create a list of integers
   List<int> numbers = [10, 20, 30, 40];

// Add a number to the list
   numbers.add(50);

// Remove a number from the list
   numbers.remove(20);

// Insert a number at a specific index in the list
   numbers.insert(1, 15);

// Iterate over the list and print each number
   print('List of numbers:');
  for (var number in numbers) {
    print(number);
  }

//Maps:
```

```
// Create a map with String keys and integer values
  Map<String, int> scoreMap = {
    'Math': 90,
    'Science': 85,
  };

// Add a new key-value pair to the map
    scoreMap['English'] = 88;

// Remove a key-value pair from the map
   scoreMap.remove('Science');
// Iterate over the map and print each key-value pair
    print('Score Map:');
  scoreMap.forEach((subject, score) {
    print('Subject: $subject, Score: $score');
  });
```

5.Loop and conditions

```
// Use a for-loop to print numbers from 1 to 5
  print('For-loop:');
  for (int i = 1; i <= 5; i++) {
    print(i);
  }
// Use a while-loop to print numbers while a condition is true
  print('While-loop:');
  int j = 1;
  while (j <= 5) {
    print(j);
    j++;
```

```
  }
// Use an if-else statement to check if a number is even or odd
   print('Even or Odd Check:');
   int n = 4; // Change this number to test with different values
  if (n % 2 == 0) {
    print('$n is even.');
  } else {
    print('$n is odd.');
  }
/*
```

6.Function

EXPLAIN : Compare positional and named function arguments

-Positional Arguments:

-Positional arguments are passed to a function in the order in which they are defined.

-The number of arguments passed must match the number of parameters defined in the function.

-Named Arguments:

-Named arguments are passed using the name of the parameter.

-Named arguments are wrapped in curly braces {} and can be made required using the required keyword.

EXPLAIN : Explain when and how to use arrow syntax for functions

-The arrow syntax is a shorthand for simple, single-expression functions.

It is used when the body of the function consists of a single expression or return statement.

```
*/
// Define a function that takes two integers and returns their sum
   int sum(int a, int b) {
    return a + b;
  }
// Call the function and print the result
```

```
//Positional
   void printSumPositional(int a, int b) {
    print('Sum (Positional): ${a + b}');
  }
// Named Arguments:
  void getArea({required int length, required int width}) {
    print('Area (Named): ${length * width}');
  }

// Call both functions with appropriate arguments
  printSumPositional(7, 3); // Positional arguments
  getArea(length: 5, width: 4); // Named arguments

/*
EXPLAIN : Can positional arguement be ommited? Show an example
  -No, positional arguments cannot be omitted.
  Example (this would cause an error):
  printSumPositional(7);  // Error: too few positional arguments

EXPLAIN : Can named arguement be ommited? Show an example
  -Named arguments can be omitted if not marked as required.
  Example:
    void getVolume({int length = 10, int width = 5, int height = 2}) {
    print('Volume: ${length * width * height}');}
    getVolume(width: 6); // Not passing all arguments, uses default values for others
*/
//Arrow Syntax:
// Define a function using arrow syntax that squares a number
```

```
int square(int number) => number * number;
// Call the arrow function and print the result
 print('Square of 4: ${square(4)}');
}
```