Name: Mujeeb Sheikh

Assignment no 6

- **Create a list of names and print all names using list.**

```
void main(){
List <String> names = ['Mujeeb', 'Daniyal', 'Owais', 'Abdullah'];
print('All Names of List is');
print([names]);
}
```

```
All Names of List is
[[Mujeeb, Daniyal, Owais, Abdullah]]
```

- **Create an empty list of type string called days. Use the add method to add names of 7 days and print all days.**

```
 void main(){
List <String> days = [];
days.add("Monday");
days.add("Tuesday");
days.add("Wednesday");
days.add("Thursday");
days.add("Friday");
days.add("Saturday");
days.add("Sunday");
print("Days Of The Week is");
print([days]);
 }
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main15.dart
Days Of The Week is
[[Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday]]
```

Create a list of Days and remove one by one from the end of list.

```
void main(){
  List <String> days =
['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday'];
  print("The List of Days is");
  print([days]);
  days.remove('Monday');
  print([days]);
  days.remove('Tuesday');
  print([days]);
  days.remove('Wednesday');
  print([days]);
  days.remove('Thursday');
  print([days]);
  days.remove('Friday');
  print([days]);
  days.remove('Saturday');
  print([days]);
  days.remove('Sunday');
  print([days]);

}
```

```
The List of Days is
[[Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday]]
[[Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday]]
[[Wednesday, Thursday, Friday, Saturday, Sunday]]
[[Thursday, Friday, Saturday, Sunday]]
[[Friday, Saturday, Sunday]]
[[Saturday, Sunday]]
[[Sunday]]
[[]]
```

Create a list of numbers & write a program to get the smallest & greatest number from a list.

```
void main(){
  List <int> numbers=[10,8,9,2,3,1];
  print('The List of Numbers is $numbers');
  numbers.sort();
  print("The Sorted Numbers $numbers");
  int smallest = numbers[0];

  int greatest = numbers[numbers.length - 1];

  print("The smallest number is: $smallest");
  print("The greatest number is: $greatest");
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main17.dart
The List of Numbers is [10, 8, 9, 2, 3, 1]
The Sorted Numbers [1, 2, 3, 8, 9, 10]
The smallest number is: 1
The greatest number is: 10
```

- **Create Map variable name world then inside it create countries Map, Key will be the name country & country value will have another map having capitalCity, currency and language to it. by using any country key print all the value of Capital & Currency.**

```
void main() {

  Map<String, Map<String, String>> world = {
    'USA': {'capitalCity': 'Washington D.C.', 'currency': 'US Dollar',
'language': 'English'},
    'France': {'capitalCity': 'Paris', 'currency': 'Euro', 'language': 'French'},
    'Japan': {'capitalCity': 'Tokyo', 'currency': 'Japanese Yen', 'language':
'Japanese'},
    // Add more countries as needed
  };

  // Specify the country key for which you want to print details
  String countryKey = 'USA';

  // Get the details map for the specified country
  Map<String, String>? countryDetails = world[countryKey];

  // Print the capital city and currency of the specified country
```

```
  print('Capital: ${countryDetails?['capitalCity']}, Currency:
${countryDetails?['currency']}');
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
Capital: Washington D.C., Currency: US Dollar
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Check if "fri" exist in expanses; if exist change it's value to 5000.0 otherwise add 'fri' to expenses and set its value to 5000.0 then print expenses.**

```
void main() {
  Map<String, double> expenses = {
    'sun': 3000.0,
    'mon': 3000.0,
    'tue': 3234.0,
  };

  if (!expenses.containsKey('fri')) {
    expenses['fri'] = 5000.0;
  } else {

    expenses['fri'] = 5000.0;
  }
  print(expenses);
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
{sun: 3000.0, mon: 3000.0, tue: 3234.0, fri: 5000.0}
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Given a list of integers, write a dart code that returns the maximum value from the list**

```
void main() {
  List<int> numbers = [10, 5, 20, 15, 30];

  // Find the maximum value using reduce
  int max = numbers.reduce((value, element) => value > element ? value :
element);

  print("Maximum value in the list: $max");
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
Maximum value in the list: 30
```

- **Write a Dart code that takes in a list of strings and removes any duplicate elements, returning a new list without duplicates. The order of elements in the new list should be the same as in the original list**

```
List<String> removeDuplicates(List<String> inputList) {
  return inputList.toSet().toList();
}

void main() {
  List<String> originalList = ['apple', 'banana', 'apple', 'orange', 'banana',
'grape'];
  List<String> newList = removeDuplicates(originalList);
  print("Original List: $originalList");
  print("List without duplicates: $newList");
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
Original List: [apple, banana, apple, orange, banana, grape]
List without duplicates: [apple, banana, orange, grape]
```

- **Write a Dart code that takes in a list of strings and prints a new list with the elements in reverse order. The original list should remain unchanged.**

```
void main() {
  List<String> originalList = ['apple', 'banana', 'orange', 'grape'];

  // Create a copy of the original list and reverse it
  List<String> reversedList = List.from(originalList.reversed);

  // Print the reversed list
  print("Reversed List: $reversedList");
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
Reversed List: [grape, orange, banana, apple]
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Implement a code that takes in a list of integers and returns a new list containing only the unique elements from the original list. The order of elements in the new list should be the same as in the original list.**

```dart
List<int> uniqueElements(List<int> inputList) {
  return inputList.toSet().toList();
}

void main() {
  List<int> originalList = [1, 2, 3, 2, 4, 3, 5, 6, 1];

  List<int> uniqueList = uniqueElements(originalList);

  print("Original List: $originalList");
  print("List with Unique Elements: $uniqueList");
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
Original List: [1, 2, 3, 2, 4, 3, 5, 6, 1]
List with Unique Elements: [1, 2, 3, 4, 5, 6]
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Write a Dart code that takes in a list of integers and prints a new list with the elements sorted in ascending order. The original list should remain unchanged.**

```dart
void main() {
  List<int> originalList = [3, 1, 4, 1, 5, 9, 2, 6];
  print('The Original List is : $originalList');

  List<int> sortedList = List.from(originalList)..sort();
  print("Sorted List: $sortedList");
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
The Original List is : [3, 1, 4, 1, 5, 9, 2, 6]
Sorted List: [1, 1, 2, 3, 4, 5, 6, 9]
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Implement a Dart code that uses the where() method to filter out negative numbers from a list of integers. The program should take in the original list as a parameter and print a new list containing only the positive numbers.**

```
void main() {
  List<int> originalList = [1, -2, 3, -4, 5, -6, 7, -8, 9];

  List<int> positiveNumbers = originalList.where((number) => number >
0).toList();

  print("List with positive numbers: $positiveNumbers");
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
List with positive numbers: [1, 3, 5, 7, 9]
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Implement a Dart code that uses the where() method to filter out odd numbers from a list of integers. The program should take in the original list as a parameter and print a new list containing only the even numbers.**

```
void main() {
  List<int> originalList = [1, 2, 3, 4, 5, 6, 7, 8, 9];

  List<int> evenNumbers = originalList.where((number) => number % 2 ==
0).toList();
    print("List with even numbers: $evenNumbers");
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
List with even numbers: [2, 4, 6, 8]
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Given a list of integers, write a Dart code that uses the map() method to create a new list with each value squared. The program should take in the original list as a parameter and print the new list.**

```dart
void main() {
  List<int> originalList = [1, 2, 3, 4, 5];

  List<int> squaredList = originalList.map((number) => number * number).toList();

  print("New list with squared values: $squaredList");
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
New list with squared values: [1, 4, 9, 16, 25]
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Create a map named "person" with the following key-value pairs: "name" as "John", "age" as 25, "isStudent" as true. Write a Dart code to check if the person is both a student and over 18 years old. Print "Eligible" if both conditions are true, otherwise print "Not eligible".**

```dart
void main() {
  Map<String, dynamic> person = {
    'name': 'John',
    'age': 25,
    'isStudent': true,
  };
  if (person['isStudent'] && person['age'] > 18) {
    print("Eligible");
  } else {
    print("Not eligible");
  }
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
Eligible
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Given a map representing a product with keys "name", "price", and "quantity", write Dart code to check if the product is in stock. If the quantity is greater than 0, print "In stock", otherwise print "Out of stock".**

```dart
void main() {
  Map<String, dynamic> product = {
    'name': 'Apple',
    'price': 1.99,
    'quantity': 10,
  };

  // Check if the product is in stock
  if (product['quantity'] > 0) {
    print("In stock");
  } else {
    print("Out of stock");
  }
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
In stock
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Create a map named "car" with the following key-value pairs: "brand" as "Toyota", "color" as "Red", "isSedan" as true. Write Dart code to check if the car is a sedan and red in color. Print "Match" if both conditions are true, otherwise print "No match".**

```dart
void main() {
  Map<String, dynamic> car = {
    'brand': 'Toyota',
    'color': 'Red',
    'isSedan': true,
  };

  // Check if the car is a sedan and red in color
  if (car['isSedan'] && car['color'] == 'Red') {
    print("Match");
  } else {
    print("No match");
  }
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
Match
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Given a map representing a user with keys "name", "isAdmin", and "isActive", write Dart code to check if the user is an active admin. If the user is both an admin and active, print "Active admin", otherwise print "Not an active admin".**

```dart
void main() {
  Map<String, dynamic> user = {
    'name': 'John Doe',
    'isAdmin': true,
    'isActive': true,
  };

  // Check if the user is an active admin
  if (user['isAdmin'] && user['isActive']) {
    print("Active admin");
  } else {
    print("Not an active admin");
  }
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
Active admin
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```

- **Given a map representing a shopping cart with keys as product names and values as quantities, write Dart code to check if a product named "Apple" exists in the cart. Print "Product found" if it exists, otherwise print "Product not found".**

```dart
void main() {
  Map<String, int> cart = {
    'Apple': 5,
    'Banana': 3,
    'Orange': 2,
  };

  // Check if "Apple" exists in the cart
  if (cart.containsKey('Apple')) {
    print("Product found");
  } else {
    print("Product not found");
  }
}
```

```
PS C:\Users\mujee\OneDrive\Desktop\Dart> dart main18.dart
Product found
PS C:\Users\mujee\OneDrive\Desktop\Dart>
```