Week2

Dart Language and Basics

Dart Basics

- print('hello'); hello is an expression and print is a statement
- 1.isEven or 'a'.isEmpty
- A variable is a reference to a value and it has a data type.
- Right click to the code and select Refactor. It is a tool that (introduce variable, rename a variable or extract a method from a piece of code etc.)
- You can declare a variable by var or String
- Declaring a function:

```
String concatStr(String a, [String b='']){
  return a+' '+ b;
}
void concatStr2 ({required String a, String b=' '}){
      print(a+' '+ b);
}
```

- Scope is defined by { and } and represents a domain for variables.
- Add final keyword before variables that won't change while running.
- Datatype int for integers and double for real valued numbers.
- Use datatype num for mixed values containing int and double's.
- You can use ' or '' for strings.
- Convert an integer to string by int.tostring method
- Ternary operator: print(messageCount>0:'You have message:'No message');
- Switch-case

```
switch (choice){
    case 0: {print("zero");}
    break;
    case 1: {print("one");}
    break;
    case 2: {print("two");}
    break;
    default: {print("no choice");}
    break;
  }
```

- While do-while and for loops are similar to C and Java.

Collections

- You can iterate on group of similar variables by using for in
- var msg=['a','b','c']; for (var m in msg) print($m);
- break commands ends a loop and continue passes the remaining code in that iteration.
- To access a lists element use msg[0],

- to add a value use msg.add('d'),
- to get the size of a list use print("size of msg is ${msg.length}");
- to check a list if it contains a particular element use if (msg.contains('a'))
- to remove an element from a list use msg.remove('a');
- to remove an element from a specific index from a list use msg.removeAt(1);
- Set is defined by { } instead of [ ]. In a set, an element can occur only once.
- Sets are not ordered thus elements can not be accessed by indices.
- You can intersect or combine sets e.g. set.intersection(newset)
- Map is similar to the set however each key is mapped to a value like a dictionary (key-value) such as map['tr']='türkiye';
- To iterate on a map use for(var m in map.entries) print('${m.key}=${m.value}');
- Declare a list by var l=[1,2,3];
- Declare a set var s={1,2,3};
- Declare a map by var m={'a':1,'b':2};
- Generics are a way to code a class or function so that it works with a range of data types instead of just one, while remaining type safe. Consider defining a variable by List<String> or List<Object>?
- List, Set and Map are reference type variables, and they are mutable. However, int, double and string are immutable. Consider this:

  var str = "This is a string."; // a string is placed in memory and the reference of that data is linked to the variable str.

  str = "This is another string."; //a whole new string is created in a different place of memory and the reference of new data is overwritten to the variable str.

Null Safety

- Null Safety in simple words means a variable cannot contain a 'null' value unless you initialized with null to that variable.
- String s; creates a non-nullable variable which is preferred mostly.
- String? s; creates a nullable variable. In this case the code is runed if you even don’t assign a value to the variable.
- To catch the exceptions use try-catch commands:

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

Classes

- Which one is better: using a map or class?
- There is no standardization and type restriction in maps.
- A class is a template contains a constructor, fields, methods, getters and setters.

```
class Student{
  String name;
  int grade;
  bool isGraduated;
  Student(this.name,this.grade) : isGraduated=false;
  String get getname {
    return name;
  }
  @override
  String toString(){
    return '$name $grade $isGraduated';
  }
}
```

- A static field of a class is a single variable and accessed through by using the class name not the object instance. Static methods are also similar behavior.
- Classes have fields and methods. To make a private field start the field name by _ and put the class in a separate file apart from the main.dart
- You can restrict or control the access to the private fields by utilizing getter and setter methods.
- Consider these declarations:

```
String ad; //non nullable: must be initiated by constructor
String? okul; //nullable: no need to init
late String adres; //non nullable need to be initiated before first use
```

- What is the difference between variables and references? Two references can point the same data (object)
- Classes can have variables with the type of their own (classes can be linked to each other by references) See Data Structures course.
- If a field is final, we cannot change it in the future (immutable fields)
- Put @immutable before class declaration to ensure that all fields will be final.
- Immutable classes can have const constructors.
- A property can be considered as a practical getter and setter.

```
int get grade{ return age-5; }
set grade(int s){age=s+5;}
```

- We use getter and setters to protect accessing a private field or to access/change the private field in a controlled way
- Inheritence in OOP: when a class derives from another class. The child class will inherit all the public and protected properties and methods from the parent class.

```
class B extends A{
 @override //polymorphism
 void theMethodOfAWithSameName(){}
 }
```

Generics and Iterables

- Generic classes allow to define fields with specific types in a class that the user can specify when creating an instance.
- List, Set and Map are iterable collections.

- List and Set implements Iterable abstract class but Map not.
- However, Map's entries, keys and values fields implements Iterable.
- Iterable is a higher type that we can iterate on that group of variables.
- To iterate on a collection use for(final e in iterable){}
- To get the first element of a collection use iterable.first,
- To get the last element of a collection use iterable.last,
- To get the size of a collection use iterable.length,
- To check if a collection includes a specific element use iterable.contains(e),
- To run a command on each element easily use iterable.forEach((e) => print(e));
- To create a new list from a collection List l=iterable.toList();
- An iterator can be created from a collection which has useful navigation methods:

```
var i=iterable.iterator();
while(i.moveNext()){ print(i.current);}
```

- Assume a variable defined as *final s={1,2,3};* You can refer this elements by print({3,5,7,..s,6,4});
- Collection if and for are used when declaring collections:
  var s=[0, if(condition) 1, for(var i=2;i<5;i++) i, 5];
- iterable.firstWhere((e) => e>3);
- iterable.any((e) => e>3); at least one element must comply the condition
- iterable.every((e) => e>3); all elements must comply the condition
- iterable.where((e) => e>3); returns a sub collection that complies the condition
- iterable.map((e) => '$e value'); converts values and builds a new collection

Other Topics

- Linter is used in Android Studio to improve the code with checks. Please see files: pubspec.yaml dev dependency and analysis_options.yaml
- It underlines some codes needs to be improved. Use alt+enter to see the quick fix suggestions.
- External libraries can be downloaded from pub.dev website.
- Use command flutter pub add english_words to add a package containing the most ~5000 used English words and some utility functions.

References

- https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/
- https://developers.google.com/community/experts/directory/profile/profile-gazihan-alankus