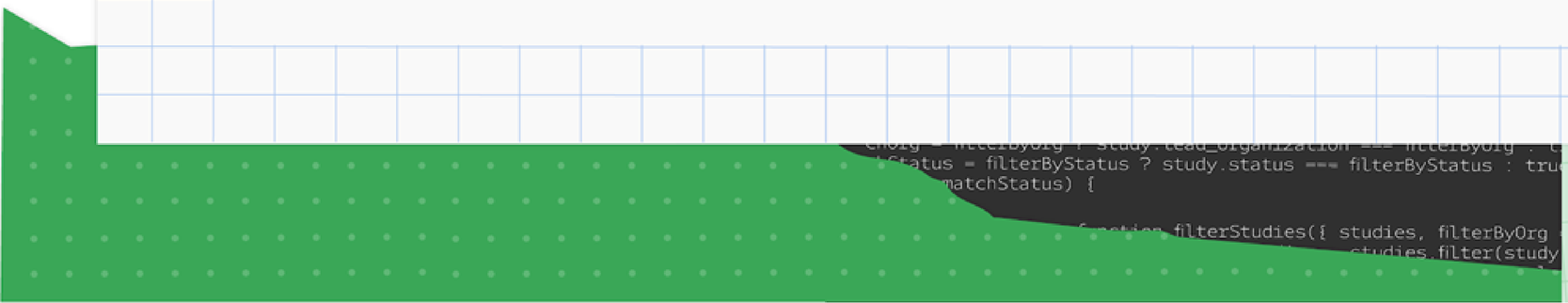
Google Developer Student Clubs

Object Oriented Programming

Everything is an object!



Classes and objects

And why?

Developers got tired rewriting same blocks of code again and again, so they invented this new way of programming, which is more readable, customizable, and efficient in time!

The syntax for classes in dart:

```
class class_name {
  // Body of class
}
```

```
class Student{
          // Fields defining  the
           // Properties of Class
          int? roll_no;
          String? name;

          void print_name(){ (##class method!##)
               print("Student Name: $name");
          }
}
```

Constructors

Is it mandatory?

A Constructor is a block of code that initializes the state and values during object creation. Constructor is name same as the class name and doesn't return any value.

Syntax:

```
class_name( [ parameters ] ){
    // Constructor Body
}
```

Objects

Is it mandatory?

Objects are the instance of the class and they are declared by using a new keyword followed by the class name.

Syntax:
var object_name = new class_name([arguments]);

```
// Creating Class named TechStream
class TechStream {

    // Creating Field inside the class
    String person = '';

    // Creating Function inside class
    void mee()
    {
        print("Welcome to $person");
    }
}

void main()
{
    // Creating Instance of class
    TechStream mee = new TechStream();

    // Calling field name person and assigning value
    // to it using object of the class TechStream
    mee.person = ‘sama’;

    // Calling function name mee using object of the class TechStream
    mee.mee();
}
```

this keyword

`This.`

- this keyword represents an implicit object pointing to the current class object. It refers to the current instance of the class in a method or constructor. The this keyword is mainly used to eliminate the ambiguity between class attributes and parameters with the same name. When the class attributes and the parameter names are the same this keyword is used to avoid ambiguity by prefixing class attributes with this keyword. this keyword can be used to refer to any member of the current object from within an instance method or a constructor

```dart
// Dart program to illustrate
// this keyword
void main()
{
Student s1 = new Student('S001');
}

class Student
{
// defining local st_id variable
var st_id;
Student(var st_id)
{
  // using this keyword
  this.st_id = st_id;
  print(" Dart THIS Example");
  print("The Student ID is : ${st_id}");
}
}
```

Concept of Inheritance

`creating daddy issues between classes`

- In Dart, one class can inherit another class i.e dart can create a new class from an existing class. We make use of extend keyword to do so.

- Terminology:
- Parent Class: It is the class whose properties are inherited by the child class. It is also known as a base class or superclass.

- Child Class: It is the class that inherits the properties of the other classes. It is also known as a deprived class or subclass.

Types of inheritance

`creating DIFFERENT daddy issues between classes`

- Single Inheritance: This inheritance occurs when a class inherits a single-parent class.
- Multiple Inheritance: This inheritance occurs when a class inherits more than one parent class. Dart doesn’t support this.
- Multi-Level Inheritance: This inheritance occurs when a class inherits another child class.
- Hierarchical Inheritance: More than one classes have the same parent class.
- Important Points:

- Child classes inherit all properties and methods except constructors of the parent class.
- Like Java, Dart also doesn’t support multiple inheritance.

```dart
// Dart program to show the single inheritance

// Creating parent class
class Parent{

// Creating a function
void output(){
  print("Welcome to Parent!!\nYou are inside output function.");
}
}

// Creating Child class
class Child extends Parent{
// We are not defining
// any thing inside it...
}
void main() {
// Creating object of Child class
var miki = new Child();

// Calling function
// inside Parent(Parent class)
miki.output();
}
```

```dart
// Dart program for heirarchial interitance
class A{

// Creating a function
void output1(){
  print("Welcome to A!!\nYou are inside the output function of A class.");
}
}

// Creating Child1 class
class ChildOfA1 extends A{

// Creating a function
void output2(){
  print("Welcome to A!!\nYou are inside the output function of ChildOfA1 class.");
}
}

// Creating Child2 class
class ChildofA2 extends ChildOfA1{
// We are not defining
// any thing inside it...
}

void main() {
// Creating object
// of ChildofA class
var miki = new ChildofA2();

// Calling function
// inside A
//(Parent class of Parent class)
miki.output1();

// Calling function
// inside ChildofA
// (Parent class)
miki.output2();
}
```

My links: