SUMMARY OF OOP (Task-4)

1- Class and Object in Dart

✓ A Class is a blueprint used to create objects.
✓ An Object is an instance of a class that contains properties and methods

Ex:

```
class Car {
  String brand = "BMW";
  int speed = 200;
  void move() {
    print("Car is moving");
  }
}
void main() {
  Car c1 = Car();
  print(c1.brand);
  c1.move();
}
```

2-Constructor in Dart

✓ A Constructor is a special method used to initialize objects.
✓ It is called automatically when an object is created.
✓ The constructor name must be the same as the class name
✓ Doesn't need a return type

3- Default Constructor

✓ A Default Constructor has no parameters.
✓ It is automatically created if no constructor is defined

EX:

```
class Student {
  Student() {
    print("Default Constructor");
  }
}
```

4-Parameterized Constructor

✓ A Parameterized Constructor accepts parameters to initialize object values

EX:

```
class Student {
 String name;
  int age;
  Student(this.name, this.age);
}
```

5-Named Constructor

- improves code readability. It is useful when you want to create multiple constructors with the same name.

  EX:

  ```
  class Student {
   String? name ;

   Student.named(this.name); }
  ```

6- Encapsulation in Dart

- ✓ means hiding data within a library, preventing it from outside factors.
- ✓ It helps you control your program and prevent it from becoming too complicated.
- ✓ Dart doesn’t support keywords like public, private, and protected.
- ✓ Dart uses _ (underscore) to make a property or method private.

Ex:

```
class Person {
 String _name = "";
}
```

7-Getter and Setter

- Getter is used to get the value of a property. It is mostly used to access a private property’s value.
- Setter is used to set the value of a property. It is mostly used to update a private property’s value.

Ex:

```
class Person {
 String _name = "";

 String get name => _name; // getter

 set name(String value) { // setter
   _name = value;
 }}
```

8- Inheritance in Dart

- ✓ it always create a is-a relation between the parent and child class, like Student is a Person
- ✓ Inheritance allows a class to inherit properties and methods from another class.
- ✓ It uses the extends keyword

-Types Of Inheritance:

- **Single Inheritance** - In this type of inheritance, a class can inherit from only one class.

- **Multilevel Inheritance** - In this type of inheritance, a class can inherit from another class and that class can also inherit from another class.

- **Hierarchical Inheritance** - In this type of inheritance, parent class is inherited by multiple subclasses.

Ex:

```
class Animal {
  void eat() {
    print("Eating");
  }
}
class Dog extends Animal {}
```

9-inhertance of constructor

- ✓ is a process of inheriting the constructor of the parent class to the child class. It is a way of reusing the code of the parent class.
- ✓ The constructor of the parent class is called first and then the constructor of the child class is called.
- ✓ Constructors are not inherited directly.
- ✓ The child class must call the parent constructor.
- ✓ The super key is used to call the constructor of the parent class and pass the values of properties

Ex:

```
    class Animal {
      Animal(String name) {
        print(name);
      }
    }

    class Dog extends Animal {
      Dog() : super("cat");
}
```

Note!!

Super is used to refer to the parent class. It is used to call the parent class’s properties and methods