1. Definisi
   1. Nested Class: Adalah class yang ada dalam sebuah class. Misalnya kita memiliki class A dan membuat class baru bernama class B yang kita deklarasikan didalam class A maka class B adalah nested class.

Dalam kasus dibawah ini class Manusia merupakan nested class

class MakhlukHidup{  
 class Manusia(nama: String, umur: Int) {  
 val nama = nama  
 val umur = umur  
 fun berjalan(): String{  
 return "$nama sedang berjalan"  
 }  
 }  
}

* 1. Singleton: Singleton merupakan pendeklarasian object tanpa instantiasi. Fungsinya adalah Ketika kita ingin mengambil nilai yang sama disetiap class tanpa harus instantiasi klass tersebut maka kita menggunakan singleton. Dalam kotlin terdapat companion object

import java.io.FileInputStream  
import java.io.InputStream  
import java.sql.Connection  
import java.sql.DriverManager  
import java.sql.PreparedStatement  
  
class Database{  
 companion object {  
 val conn = DriverManager.getConnection( "jdbc:mysql://localhost:3306/app", "root", ""

)  
 }  
 fun tambah(nama: String, nim: String, foto: FileInputStream){  
  
 }  
}  
  
fun main(){  
 val statement = Database.conn.prepareStatement("INSERT INTO tugas (judul, matkul, deadline) values (?, ?, ?)")  
   
}

Dalam contoh diatas kita sudah menambahkan database conn sebagai koneksi kita ke dalam database. Nah disitu kita bisa mengambilnya kapanpun tanpa harus instansiasi class database. Contoh lain biasanya adalah Pi dalah Math.Pi sebagai nilai Pi

* 1. Interface: Interface adalah sebuah class yang digunakan sebagai acuan untuk membuat class. Jadi interface merupakan blueprint dari sebuah class. Menggunakan interface memudahkan kita membuat banyak class yang berasal dari satu sumber

interface MakhlukHidup {  
 val habitat:String  
 fun bergerak()  
}  
class Orang : MakhlukHidup {  
 override val habitat = "Darat"  
 override fun bergerak() {  
 *println*("Berjalan")  
 }  
}

Dalam kotlin interface di awali dengan keyword interface, maka MakhlukHidup adalah interface