1. Triple:
   - Kotlin’s ‘Triple‘ is a convenient data structure that allows you to store three values together as a single object.
   - Example: ‘val triple = Triple("A", 1, 3.14)‘
   - It provides ‘component1()‘, ‘component2()‘, and ‘component3()‘ functions for accessing its elements.

2. Inline Function:
   - An inline function in Kotlin is a function that gets "inlined" at the call site, reducing the overhead of function calls.
   - Useful for higher-order functions to avoid runtime overhead.

3. Object Overloading:
   - However, you can achieve similar behavior using named and default arguments in functions.
   - This allows you to define multiple functions with the same name but different parameters or default values.
   - Example:

```
fun greet(name: String) {
    println("Hello, $name!")
}

fun greet(name: String, age: Int) {
    println("Hello, $name! You are $age years old.")
}
```

4. Destructuring Objects:
   - Kotlin supports destructuring declarations, which allow you to break an object into its component parts and assign them to variables in a single line.
   - This works for data classes, arrays, collections, and more.
   - Example:

```
data class Person(val name: String, val age: Int)

val person = Person("Aljay", 30)
val (name, age) = person
```

5. Regex :
   - Kotlin provides native support for regular expressions through its ‘Regex‘ class.
   - You can create regex patterns, match them against strings, extract substrings, and replace parts of strings.
   - Example:

```
val regex = Regex("[a-zA-Z]+")
val matchResult = regex.find("Hello, Kotlin!")
matchResult?.value // Returns "Hello"
```