خب حالا که با چند تا از ویژگی های Lambda آشنا شدین خوبه که سراغ بحث دیگه ای به نام Closure ها بریم:

fun unaryOperation**(**x:Int,op: **(**Int**) ->** Int**)**{  
 op(x)  
}

fun outsideFunction**()** {  
 val number = 10  
 unaryOperation**(**20**) {** it \* number **}**}

یک تابع نوشتیم که یک تابع High-Order داخلش استفاده شده. (همون unaryOperation که توی جلسه قبلی ساختیمش) و دیدین که این تابع High-Order به متغیر خارج عبارت لاندایی که تعریف کردیم دسترسی داره! و اگه با IntelliJ IDEA این کد رو بزنین متوجه میشین که با نگه داشتن موشواره بر روی number بهمون میگه که “variable captured in a closure” و درواقع این بهمون میفهمونه که ما به مقدار متغییری دسترسی داشتیم که خارج عبارت lambda تعریف شده و مشکلی هم نداره.

بذارین یه حلقه for داشته باشیم

fun unaryOperation**(**x:Int,op: **(**Int**) ->** Int**)**{  
 op(x)  
}

fun outsideFunction**()** {  
 for **(**number in 1..30**)** {  
 unaryOperation**(**20**) {** println**(**number**)  
 it** \* number  
 **}** }  
}

fun main(args: Array<String>){

outsideFunction()

}

که بیاد و در هر مرتبه اجرا مقدار number رو پرینت کنه. اگه توی main تابع outsideFunction رو صدا بزنیم متوجه میشیم که مقادیر تغییر میکنن و لازم به گفتنه که در بعضی زبان‌ها اینجوری تعریف شدن که مقدار متغییرها ثابت بمونن ولی در کاتلین میتونن تغییر کنند.