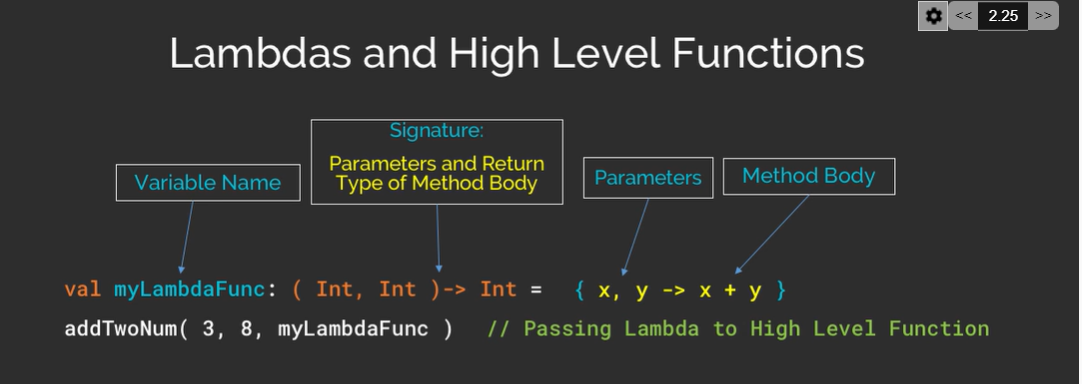
**Lambda Function**

**Lambda Function Syntax**



**Code Example**

**Example 1**

fun main() {  
 val lambdaFunction:(Int,Int)->Int=**{**x,y**->**x+y**}** *println*(lambdaFunction(10,20))  
}

**Example 2**

fun main() {  
 *println*(*credentials*("Vinay","Mandira"))  
}  
  
val *credentials*:(String,String)->String=**{**Fristname,Lastname**->** val passportname:String=Fristname+" "+Lastname;  
 passportname  
**}**

**Lambda Expression Can be Written in 2 ways**

1. With Type Annotation in Lambda Expression
2. Without Type Annotation in Lambda Expression

**With Type Annotation**

val *sum1* = **{** a: Int, b: Int **->** a + b **}**

**WithoutType Annotation**

val sum:(Int,Int)->Int=**{**a,b**->**a\*b**}**

**Return Type in Lambda**

// Return Type is Int  
val *sum*:(Int,Int)->Int=**{**a,b**->**a\*b**}**//Return Type is Unit i.e nothing return  
val *sum\_1*:(Int,Int)->Unit=**{**a,b**->**a+b**}**// If the return type is not written by default it will return the expected DataType.  
val *sum\_2*=**{**a:Int,b:Int**->**a\*b**}***/\*\*  
 \* Note:- If the operation is handled in a single statement the return type is automatically Given from Compiler.  
 \* Hover ing over the reference variable It can be found out.  
 \* In this return type is Int.  
 \*/*val *sum\_3*=**{**a:Int,b:Int**->** a+b**}***/\*\*  
 \* Note :- If the operation is not handled in a single line ,rather some extra operation is carried.  
 \* Then by default the return type is Unit.  
 \* But return type can be mentioned by declaring it.  
 \* In this return type is Int  
 \*/*val *sum\_4*=**{**a:Int,b:Int**->** val realProfit=a\*b  
 realProfit  
**}**