**/ primary with secondary constructor using this**

class Kotlin(var language:String,var type:String){

init{

println("${language} ${type}")

}

constructor( language:String, type:String, password:String):this( language, type){

println("${language} ${type} ${password}")

}

}

fun main(){

var c = Kotlin("kotlin","OOp")

var d = Kotlin("java","oop","java123")

}

class Person(number:Int, image:String){

init{

println("The value of data is ${number} and ${image}")

}

constructor(number:Int, image:String, properties:String):this(number, image){

println("${number} ${image} ${properties}")

}

}

fun main(){

val c = Person(3,"ZOhaib")

val d = Person(23,"Zaibi","hello")

println(c)

}

**Simple Inheritance**

open class Person(var name:String,var directory:String){

fun statement(){

print("${name} ${directory}")

}

}

class ClassA(var name1:String,var directory1:String):Person("Zohaib","C"){

fun statement1(){

println("The value is ${name1} and ${directory1}")

}

}

class ClassB(var name2:String,var directory2:String):Person("Zohaib","C"){

fun statement2(){

println("The value is ${name2} and ${directory2}")

}

}

fun main(){

val c = ClassA("Zohaib","Muhammad")

c.statement()

println(c)

val d = ClassB("Hello","D")

d.statement2()

}

**Method overloading**

class Function(){

fun overloading(name:String , id:String){

println(" ${name} ${id}")

}

fun overloading(name:Int, id:String, number:String){

println("${name} ${id} ${number}")

}

}

fun main(){

val c= Function()

c.overloading("Hello","ifelse")

c.overloading(2,"hhillo","3")

}