**[ 2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]**

Practical: 1



# AIM-KOTLIN PROGRAMS

Submitted By: Maniya Omkumar S.

Enrollment number: 20012021012



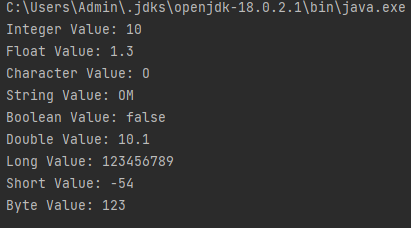
**Department of Computer Engineering/Information Technolo**

1. **Store & Display values in different variable of different type (Integer, Double, Float, Long, Short, Byte, Char, Boolean, String)**

**Answer:**

fun main()  
{  
 var i:Int = 10  
 *println*("Integer Value: $i")  
 var f:Float = 1.3F  
 *println*("Float Value: $f")  
 var c:Char = '0'  
 *println*("Character Value: $c")  
 var s:String = "OM"  
 *println*("String Value: $s")  
 var b:Boolean = false  
 *println*("Boolean Value: $b")  
 var d:Double = 10.1  
 *println*("Double Value: $d")  
 var l:Long = 123456789  
 *println*("Long Value: $l")  
 var sh:Short = -54  
 *println*("Short Value: $sh")  
 var by:Byte = 123  
 *println*("Byte Value: $by")  
}

**Output:**

****

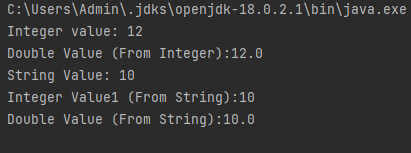
1. **Type conversion:**

**Integer to Double, String to Integer, String to Double.**

**Answer:**

fun main(){  
 var i:Int = 12  
 *println*("Integer value: $i")  
 var d:Double = i.toDouble()  
 *println*("Double Value (From Integer):$d")  
 var s:String = "10"  
 *println*("String Value: $s")  
 var i1:Int = s.*toInt*()  
 *println*("Integer Value1 (From String):$i1")  
 var d1:Double = s.*toDouble*()  
 *println*("Double Value (From String):$d1")  
}

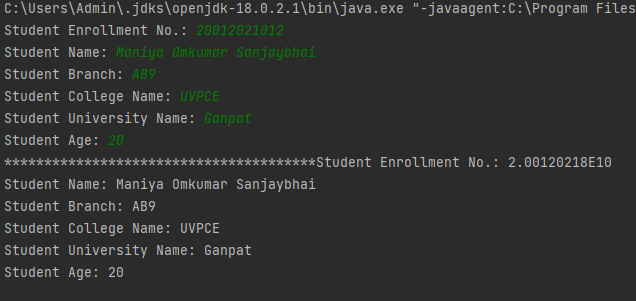
**Output:**

****

1. **Scan student’s information and display all the data.Answer:**

fun main(){  
 *print*("Student Enrollment No.: ")  
 var sn:Float = *readLine*()!!.*toFloat*()  
 *print*("Student Name: ")  
 var sname = *readLine*()  
 *print*("Student Branch: ")  
 var sb = *readLine*()  
 *print*("Student College Name: ")  
 var scn = *readLine*()  
 *print*("Student University Name: ")  
 var sun = *readLine*()  
 *print*("Student Age: ")  
 var sa:Int = *readLine*()!!.*toInt*()  
  
 *print*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
  
 *println*("Student Enrollment No.: $sn")  
 *println*("Student Name: $sname")  
 *println*("Student Branch: $sb")  
 *println*("Student College Name: $scn")  
 *println*("Student University Name: $sun")  
 *println*("Student Age: $sa")  
}

**Output:**

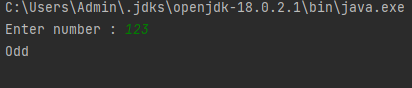
****

1. **Find the number is odd or even by using Control Flow inside println() method.** **Find the number is odd or even by using Control Flow inside println() method.**

**Answer:**

fun main(){  
 *print*("Enter number : ")  
 var x:Int = *readLine*()!!.*toInt*()  
  
 if(x % 2 == 0)  
 *println*("Even")  
 else  
 *println*("Odd")  
}

**Output:**

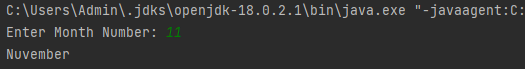
****

1. **Display month name using When**

**Answer:**

fun main(){  
 *print*("Enter Month Number: ")  
 var m:Int = *readln*()!!.*toInt*()  
  
 when(m){  
 1 -> *print*("January")  
 2 -> *print*("February")  
 3 -> *print*("March")  
 4 -> *print*("April")  
 5 -> *print*("May")  
 6 -> *print*("June")  
 7 -> *print*("July")  
 8 -> *print*("August")  
 9 -> *print*("September")  
 10 -> *print*("October")  
 11 -> *print*("Nuvember")  
 12 -> *print*("December")  
 else -> *print*("Enter proper number")  
 }  
}

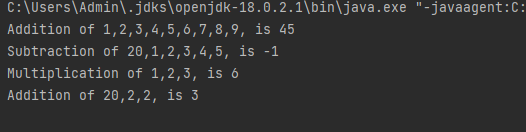
**Output:**

****

1. **By using a user defined function perform all arithmetic operations.Answer:**

fun main ()  
{  
 fun add(vararg x:Int){  
 var add:Int = x[0]  
 for(i in x[1]..x.size) {  
 add += i  
 }  
 var y:String = ""  
 for(i in x){  
 y += "$i,"  
 }  
 *println*("addition of $y is $add")  
 }  
  
 fun sub(vararg x:Int) {  
 var sub: Int = x[0]  
 for (i in x[1]..x.size){  
 sub -= i  
 }  
 var y:String = ""  
 for(i in x){  
 y += "$i,"  
 }  
 *println*("subtraction of $y is $sub")  
 }  
  
 fun mul(vararg x:Int){  
 var mul:Int = x[0]  
 for(i in x[1]..x.size) {  
 mul \*= i  
 }  
 var y:String = ""  
 for(i in x){  
 y += "$i,"  
 }  
 *println*("multiplication of $y is $mul")  
 }  
 fun div(vararg x:Int){  
 var div:Int = x[0]  
 for(i in x[1]..x.size) {  
 div /= i  
 }  
 var y:String = ""  
 for(i in x){  
 y += "$i,"  
 }  
 *println*("divison of $y is $div")  
 }  
 add (1,2,3,4,5,6)  
 sub(30,1,4,5)  
 mul(5,2)  
 div(20,2)  
}

**Output:**

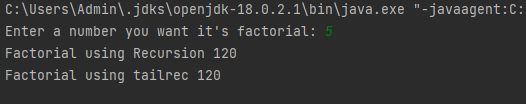
****

1. **Find the factorial of number by recursion. Explain "tailrec" keyword.**

**Answer:**

fun main() {  
 *print*("Enter a number you want it's factorial: ")  
 val n = *readLine*()!!.*toInt*()  
 *println*("Factorial using Recursion " + *recursionFact*(n))  
 *println*("Factorial using tailrec " + *fact*(n))  
}  
  
fun recursionFact(n: Int): Int {  
 if (n == 1 || n == 0) {  
 return 1  
 }  
 return n \* *recursionFact*(n - 1)  
}  
  
tailrec fun fact(n: Int, temp: Int = 1): Int {  
 return if (n == 1) {  
 temp  
 } else {  
 *fact*(n - 1, temp \* n)  
 }  
}

**Output:**

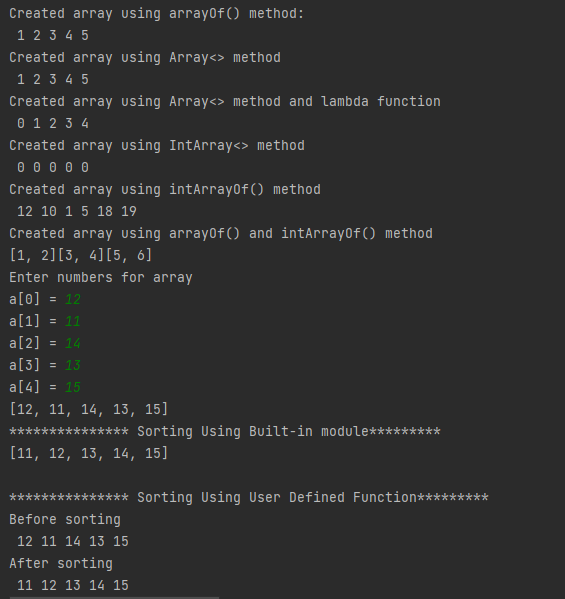
****

1. **Create different types of Array as shown in image. Explore Arrays.deepToString(), contentDeepToString() methods, IntArray variable .joinToString() and use in program to print Array. Explore range, downTo, until etc. for loop and use in this program. Sort Array of Integer data type without using inbuilt function & with using inbuilt function**

**Answer:**

fun main() {  
 val arr1 = *arrayOf*(1, 2, 3, 4, 5)  
 *println*("Created array using arrayOf() method: ")  
 *printArray*(arr1)  
 *println*()  
  
  
 val arr2: Array<Int> = *arrayOf*(1, 2, 3, 4, 5)  
 *println*("Created array using Array<> method")  
 *printArray*(arr2)  
 *println*()  
  
  
 *println*("Created array using Array<> method and lambda function")  
 val arr3 = Array(5) **{** i: Int **->** i **}** *printArray*(arr3)  
 *println*()  
  
  
 *println*("Created array using IntArray<> method")  
 val arr4 = IntArray(5)  
 *printArray*(arr4.*toTypedArray*())  
 *println*()  
  
  
 *println*("Created array using intArrayOf() method")  
 val arr5 = *intArrayOf*(12, 10, 1, 5, 18, 19)  
 *printArray*(arr5.*toTypedArray*())  
 *println*()  
  
 *println*("Created array using arrayOf() and intArrayOf() method")  
 val arr6 = *arrayOf*(*intArrayOf*(1, 2), *intArrayOf*(3, 4), *intArrayOf*(5, 6))  
 *printArray*(arr6)  
 *println*()  
  
  
 *println*("Enter numbers for array")  
 val a = ArrayList<Int>()  
 for (i in 0..4) {  
 *print*("a[$i] = ")  
 val temp: Int = *readLine*()!!.*toInt*()  
 a.add(temp)  
 }  
 *println*(a)  
  
 *println*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sorting Using Built-in module\*\*\*\*\*\*\*\*\*")  
 var c = ArrayList<Int>()  
 c = a.clone() as ArrayList<Int>  
 c.*sort*()  
 *println*(c)  
 *println*()  
  
 *println*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sorting Using User Defined Function\*\*\*\*\*\*\*\*\*")  
 *println*("Before sorting")  
 *printArray*(a.*toTypedArray*())  
 var b = ArrayList<Int>()  
 b = *sortArray*(a)  
 *println*()  
 *println*("After sorting")  
 *printArray*(b.*toTypedArray*())  
}  
  
fun printArray(arr: Array<Int>) {  
 for (element in arr) {  
 *print*(" $element")  
 }  
}  
  
  
fun printArray(arr: Array<IntArray>) {  
 for (row in arr) {  
 *print*(row.*contentToString*())  
 }  
}  
  
fun sortArray(arr: ArrayList<Int>): ArrayList<Int> {  
  
 var i: Int  
 var tmp: Int  
 for (i in 0 *until* arr.size) {  
 for (j in i + 1 *until* arr.size) {  
 if (arr[j] < arr[i]) {  
 tmp = arr[i]  
 arr[i] = arr[j]  
 arr[j] = tmp  
 }  
  
 }  
 }  
  
 return arr  
}

**Output:**

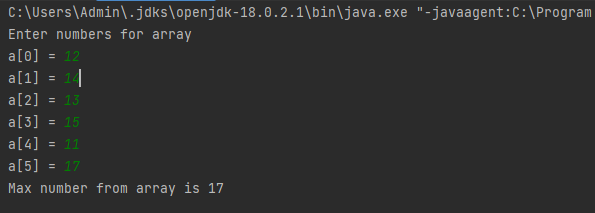
****

1. **Find the max number from ArrayList.**

**Answer:**

fun main() {  
 *println*("Enter numbers for array")  
 val a = ArrayList<Int>()  
 for (i in 0..5) {  
 *print*("a[$i] = ")  
 val temp: Int = *readLine*()!!.*toInt*()  
 a.add(temp)  
 }  
  
 *println*("Max number from array is ${*maxNum*(a)}")  
}  
  
fun maxNum(arr: ArrayList<Int>): Int {  
 var temp = arr[0]  
 for (i in 0..5) {  
 if (arr[i] > temp) {  
 temp = arr[i]  
 }  
 }  
 return temp  
}

**Output:**

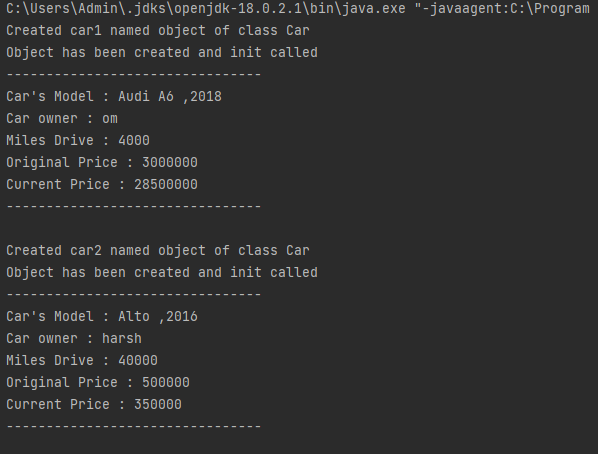
****

1. **Write Different types of Class & Constructor. Create a class Car and set various members like type, model, price, owner, milesDrive. add the function getCarPrice in it. Create an object of Car class and access property of it. (getCarInformation(), getOriginalCarPrice(), getCurrentCarPrice(), displayCarInfo() etc.)**

**Answer:**

class Car constructor(cmodel: String, oprice: Int, cowner: String, mdrive: Int, cprice: Int) {  
 var model: String  
 var originalPrice: Int  
 var owner: String  
 var mileDrive: Int  
 var currentPrice: Int  
  
 init {  
 this.model = cmodel  
 this.originalPrice = oprice  
 this.owner = cowner  
 this.mileDrive = mdrive  
 this.currentPrice = cprice  
 *println*("Object has been created and init called")  
 *println*("--------------------------------")  
 *println*("Car's Model : $model")  
 *println*("Car owner : $owner")  
 *println*("Miles Drive : $mileDrive")  
 *println*("Original Price : $originalPrice")  
 *println*("Current Price : $currentPrice")  
 *println*("--------------------------------\n")  
  
 }  
  
 fun getCarInformation() {  
 *println*("Car's Model : $model")  
 }  
  
 fun carOwner() {  
 *println*("Car owner : $owner")  
 }  
  
 fun mileDrove() {  
 *println*("Miles Drive : $mileDrive")  
 }  
  
 fun getOriginalCarPrice() {  
 *println*("Original Price : $originalPrice")  
 }  
  
 fun getCurrentCarPrice() {  
 *println*("Current Price : $currentPrice")  
 }  
  
 fun displayCarInfo() {  
 getCarInformation()  
 carOwner()  
 mileDrove()  
 getOriginalCarPrice()  
 getCurrentCarPrice()  
 }  
  
  
}  
  
fun main() {  
 *println*("Created car1 named object of class Car")  
 var car1 = Car("Audi A6 ,2018", 3000000, "om", 4000, 28500000)  
  
 *println*("Created car2 named object of class Car")  
 var car2 = Car("Alto ,2016", 500000, "harsh", 40000, 350000)  
  
  
  
}

**Output:**

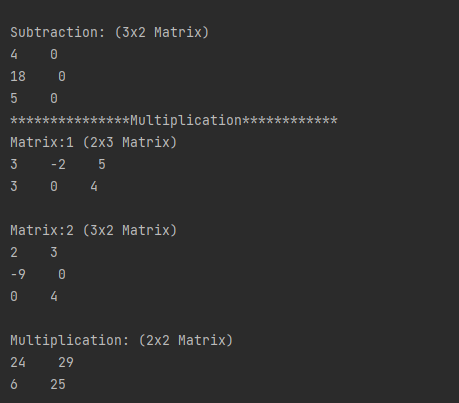
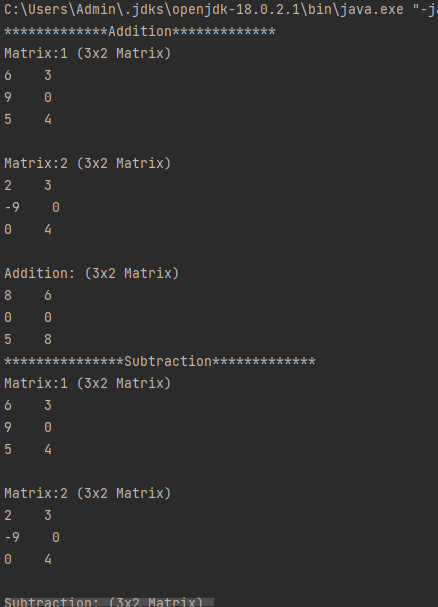
****

1. **Write about Operator Overloading. Perform Matrix Addition, Subtraction & Multiplication using Class Matrix & operator overloading. Overload toString() function in Matrix class.**

**Answer:**

import java.util.Arrays  
  
  
class Matrix(arrs: Array<IntArray>, noOfRow: Int, noOfCol: Int) {  
 var r:Int = noOfRow  
 var c:Int = noOfCol  
 var m = arrs  
  
 fun displayMat(final: Array<IntArray>){  
 for (row in final) {  
 for (column in row) {  
 *print*("$column ")  
 }  
 *println*()  
 }  
 }  
  
 operator fun plus(b: Matrix) {  
 // Adding Two matrices  
 val sum = Array(r) **{** IntArray(c) **}** for (i in 0..r - 1) {  
 for (j in 0..c - 1) {  
 sum[i][j] = m[i][j] + b.m[i][j]  
 }  
 }  
 return displayMat(sum)  
 }  
  
 operator fun minus(b: Matrix) {  
 // Subtracting Two matrices  
 val sub = Array(r) **{** IntArray(c) **}** for (i in 0..r - 1) {  
 for (j in 0..c - 1) {  
 sub[i][j] = m[i][j] - b.m[i][j]  
 }  
 }  
 return displayMat(sub)  
 }  
  
 operator fun times(b: Matrix) {  
 // Subtracting Two matrices  
 val t = Array(r) **{** IntArray(b.c) **}** for (i in 0..r - 1) {  
 for (j in 0..b.c - 1) {  
 for (k in 0 .. c - 1){  
 t[i][j] += m[i][k] \* b.m[k][j]  
 }  
 }  
 }  
 return displayMat(t)  
 }  
}  
  
fun displayarray(a: Array<IntArray>){  
 for (row in a) {  
 for (column in row) {  
 *print*("$column ")  
 }  
 *println*()  
 }  
 *println*()  
}  
  
fun main() {  
 val firstMatrix = Matrix(*arrayOf*(*intArrayOf*(3, -2, 5), *intArrayOf*(3, 0, 4)), 2, 3)  
 val secondMatrix = Matrix(*arrayOf*(*intArrayOf*(2, 3), *intArrayOf*(-9, 0), *intArrayOf*(0, 4)), 3, 2)  
 val secondMatrix1 = Matrix(*arrayOf*(*intArrayOf*(6, 3), *intArrayOf*(9, 0), *intArrayOf*(5, 4)), 3, 2)  
  
 *println*("\*\*\*\*\*\*\*\*\*\*\*\*\*Addition\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 *println*("Matrix:1 (3x2 Matrix)")  
 *displayarray*(secondMatrix1.m)  
 *println*("Matrix:2 (3x2 Matrix)")  
 *displayarray*(secondMatrix.m)  
 *println*("Addition: (3x2 Matrix)")  
 val add = secondMatrix1 + secondMatrix  
  
  
 *println*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Subtraction\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 *println*("Matrix:1 (3x2 Matrix)")  
 *displayarray*(secondMatrix1.m)  
 *println*("Matrix:2 (3x2 Matrix)")  
 *displayarray*(secondMatrix.m)  
 *println*("Subtraction: (3x2 Matrix)")  
 val sub = secondMatrix1 - secondMatrix  
  
 *println*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Multiplication\*\*\*\*\*\*\*\*\*\*\*\*")  
 *println*("Matrix:1 (2x3 Matrix)")  
 *displayarray*(firstMatrix.m)  
 *println*("Matrix:2 (3x2 Matrix)")  
 *displayarray*(secondMatrix.m)  
 *println*("Multiplication: (2x2 Matrix) ")  
 val multiplication = firstMatrix \* secondMatrix  
}

**Output:**

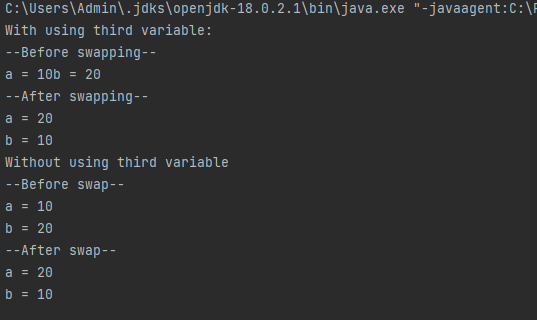
****

**E.1 Swap Value of two variables without using third variable and with using third variable.**

**Answer:**

fun main() {  
  
 *println*("With using third variable:")  
 var a = 10  
 var b = 20  
 *println*("--Before swapping--")  
 *print*("a = $a")  
 *println*("b = $b")  
 val temporary = a  
 a = b  
 b = temporary  
 *println*("--After swapping--")  
 *println*("a = $a")  
 *println*("b = $b")  
  
 *println*("Without using third variable")  
 var a1 = 10  
 var a2 = 20  
  
 *println*("--Before swap--")  
 *println*("a = $a1")  
 *println*("b = $a2")  
  
 a1 = a1 - a2  
 a2 = a1 + a2  
 a1 = a2 - a1  
  
 *println*("--After swap--")  
 *println*("a = $a1")  
 *println*("b = $a2")  
}

**Output:**

****

**E.2 Create two class named as Product and Laptop. Inherit with this information: Product class should be parent and child class should be Laptop class.**

**Add Product Name, Quantity, Amount per Quantity in Product class. In Laptop class add CPU name, RAM size, HDD Size, etc. of Laptop configuration.**

**Create primary and secondary Constructor of both class.**

**If Primary constructor is there then can we create secondary constructor in inheritance?**

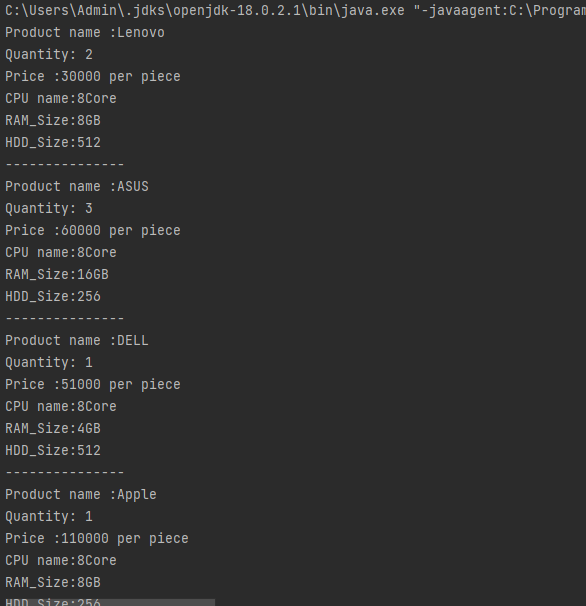
**If we can create secondary and primary constructor both in child class then what is restriction if parent have more than two different secondary constructor?**

**Create List of 5 laptops in ArrayList and display all objects information.**

**Answer:**

open class Product(name: String,quantity: Int){  
 var p:Int=-1  
 init {  
 *println*("Product name :$name")  
 *println*("Quantity: $quantity")  
  
 }  
 constructor(name: String,quantity: Int,p: Int) : this(name,quantity){  
 this.p=p  
 *println*("Price :$p per piece")  
 }  
}  
class Laptop(name: String,quantity: Int,price : Int,cpu\_n:String,RAM\_s:String):Product(name,quantity,price){  
 var hdd:String=""  
 init{  
 *println*("CPU name:$cpu\_n")  
 *println*("RAM\_Size:$RAM\_s")  
  
 }  
 constructor(name: String,quantity:Int,price: Int,cpu\_n: String,RAM\_s: String,hdd:String):this(name,quantity,price,cpu\_n,RAM\_s){  
 this.hdd=hdd  
 *println*("HDD\_Size:$hdd")  
 *println*("---------------")  
 }  
  
}  
  
fun main(){  
 var l1=Laptop("Lenovo",2,30000,"8Core","8GB","512")  
 var l2=Laptop("ASUS",3,60000,"8Core","16GB","256")  
 var l3=Laptop("DELL",1,51000,"8Core","4GB","512")  
 var l4=Laptop("Apple",1,110000,"8Core","8GB","256")  
 var l5=Laptop("HP",2,4500,"8Core","16GB","512")  
}

**Output:**

****

**E.3 Create two class named as Person and Student. Inherit with this information: Person class should be parent and child class should be Student class.**

**Add first name, last name, age in Person class. In Laptop class add enrollment no, branch, class, lab batch, etc.**

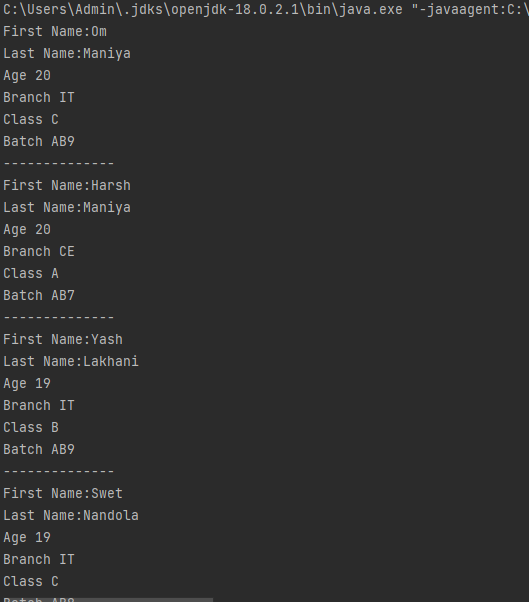
**Create primary and secondary Constructor of both class.**

**Create List of 5 students in ArrayList and display all objects information.**

**Answer:**

open class Person(firstname:String,lastname:String){  
 var age:Int=-1  
 init{  
 *println*("First Name:$firstname")  
 *println*("Last Name:$lastname")  
 }  
 constructor(firstname: String,lastname: String,age:Int):this(firstname,lastname){  
 this.age=age  
 *println*("Age $age")  
 }  
}  
class Student(firstname:String,lastname:String,age: Int,eno:Long,branch:String,Class:String):Person(firstname,lastname,age){  
 var batch:String=""  
 init{  
 *println*("Branch $branch")  
 *println*("Class $Class")  
 }  
 constructor(firstname:String,lastname:String,age: Int,eno:Long,branch:String,Class:String,batch:String):this(firstname,lastname,age,eno,branch,Class){  
 this.batch=batch  
 *println*("Batch $batch")  
 *println*("--------------")  
 }  
  
}  
fun main(){  
 var s1=Student("Om","Maniya",20,20012021012,"IT","C","AB9")  
 var s2=Student("Harsh","Maniya",20,20012011059,"CE","A","AB7")  
 var s3=Student("Yash","Lakhani",19,20012021042,"IT","B","AB9")  
 var s4=Student("Swet","Nandola",19,20012021015,"IT","C","AB8")  
 var s5=Student("Hari","Talaviya",20,20012021063,"IT","A","AB2")  
  
}

**Output:**

****