[2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]

Practical: 1

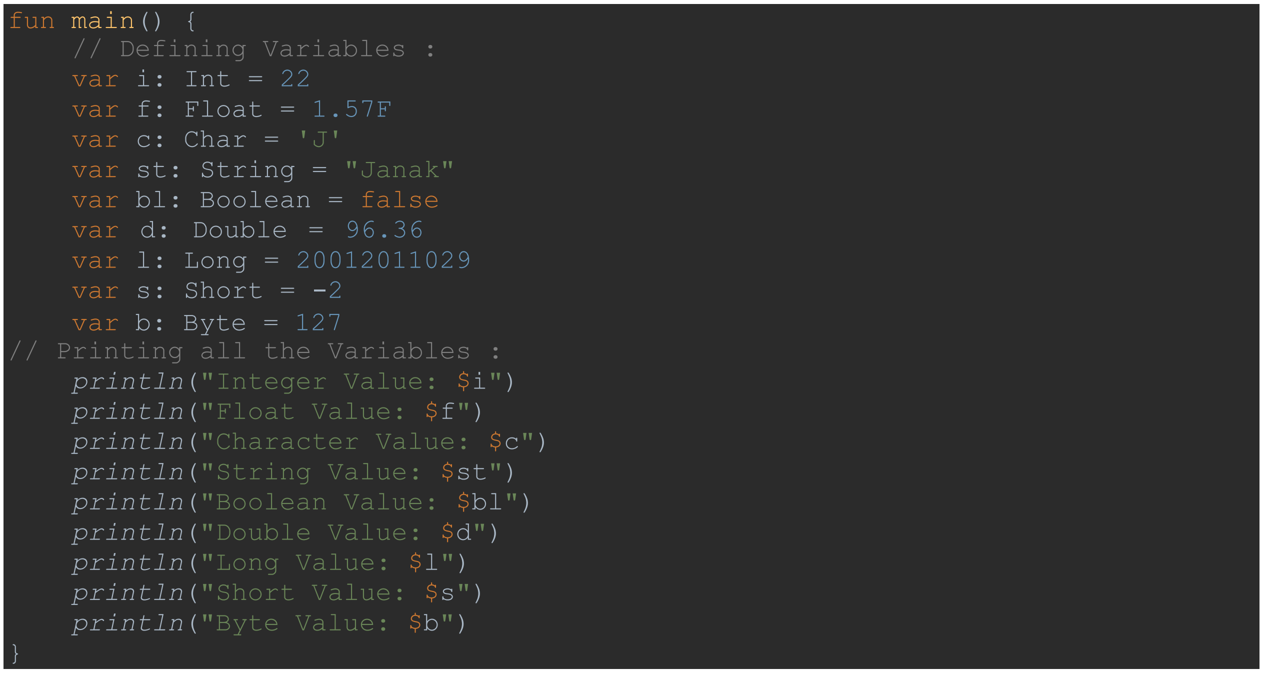
AIM-KOTLIN PROGRAMS

Submitted By: Patel Nishant J.
Enrollment number: 20012011115

Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering

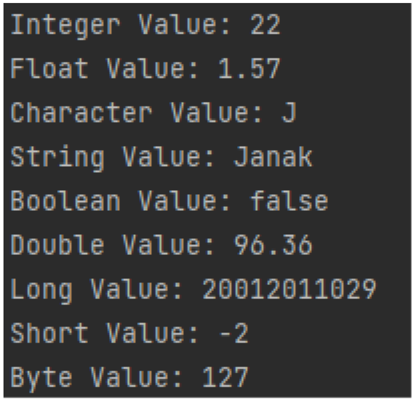
1. **Store & Display values in different variable of different type (Integer, Double, Float, Long, Short, Byte, Char, Boolean, String)**

Input code:-



```
fun main() {
    // Defining Variables :
    var i: Int = 22
    var f: Float = 1.57F
    var c: Char = 'J'
    var st: String = "Janak"
    var bl: Boolean = false
    var d: Double = 96.36
    var l: Long = 20012011029
    var s: Short = -2
    var b: Byte = 127
// Printing all the Variables :
    println("Integer Value: $i")
    println("Float Value: $f")
    println("Character Value: $c")
    println("String Value: $st")
    println("Boolean Value: $bl")
    println("Double Value: $d")
    println("Long Value: $l")
    println("Short Value: $s")
    println("Byte Value: $b")
}
```

output:

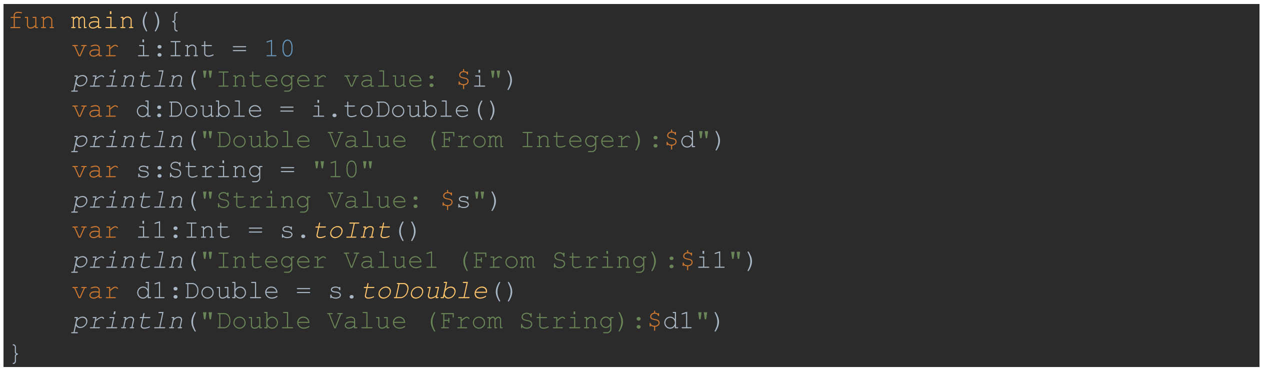


```
Integer Value: 22
Float Value: 1.57
Character Value: J
String Value: Janak
Boolean Value: false
Double Value: 96.36
Long Value: 20012011029
Short Value: -2
Byte Value: 127
```

2. Type conversion:

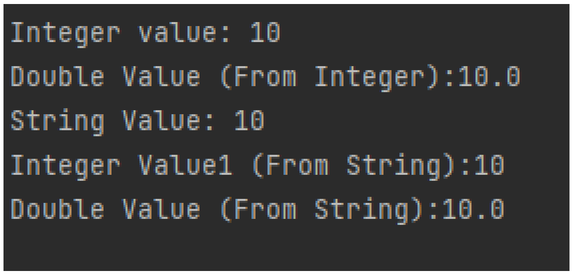
Integer to Double, String to Integer, String to Double.

Code:



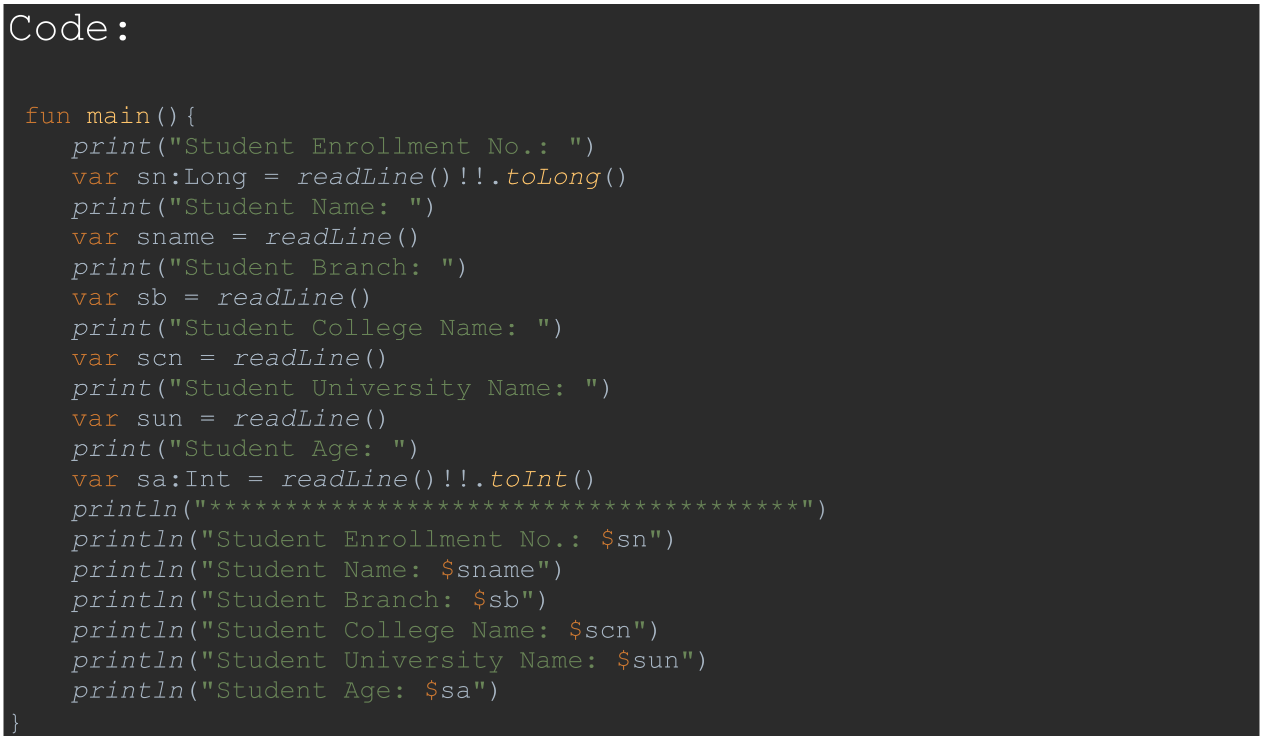
```
fun main(){
    var i:Int = 10
    println("Integer value: $i")
    var d:Double = i.toDouble()
    println("Double Value (From Integer):$d")
    var s:String = "10"
    println("String Value: $s")
    var i1:Int = s.toInt()
    println("Integer Value1 (From String):$i1")
    var d1:Double = s.toDouble()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

Output:



```
Integer value: 10
Double Value (From Integer):10.0
String Value: 10
Integer Value1 (From String):10
Double Value (From String):10.0
```

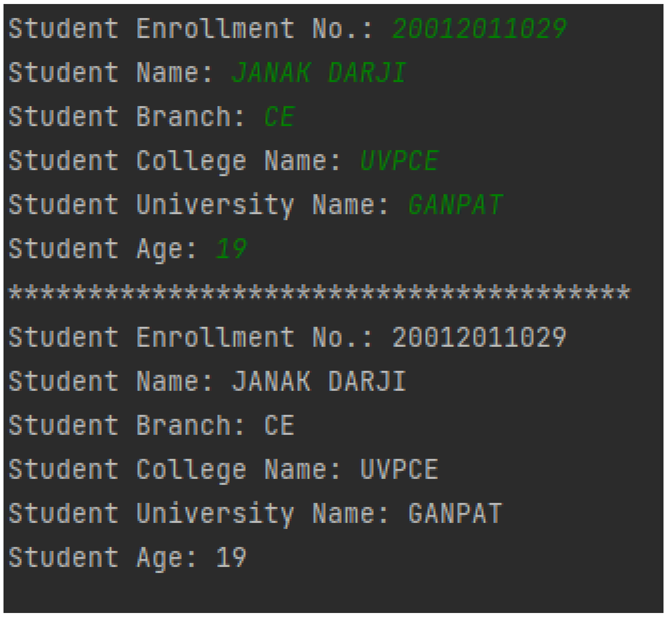
3.Scan **student's information and display all the data.**



Code:

```
 fun main(){
    print("Student Enrollment No.: ")
    var sn:Long = readLine()!!.toLong()
    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()
    println("*************************************")
    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")
}
```

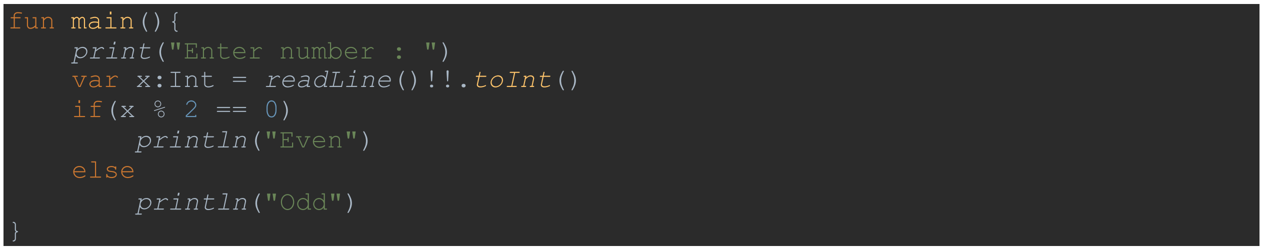
Output:



```
Student Enrollment No.: 20012011029
Student Name: JANAK DARJI
Student Branch: CE
Student College Name: UVPCE
Student University Name: GANPAT
Student Age: 19
***************************************
Student Enrollment No.: 20012011029
Student Name: JANAK DARJI
Student Branch: CE
Student College Name: UVPCE
Student University Name: GANPAT
Student Age: 19
```

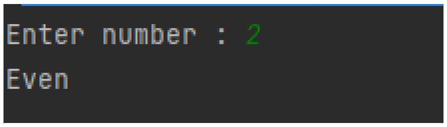
4.Find the number is odd or even by using Control Flow inside println() method

Code:

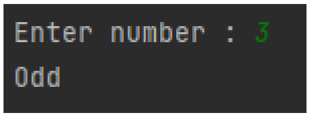


```
fun main(){
    print("Enter number : ")
    var x:Int = readLine()!!.toInt()
    if(x % 2 == 0)
        println("Even")
    else
        println("Odd")
}
```

Output:



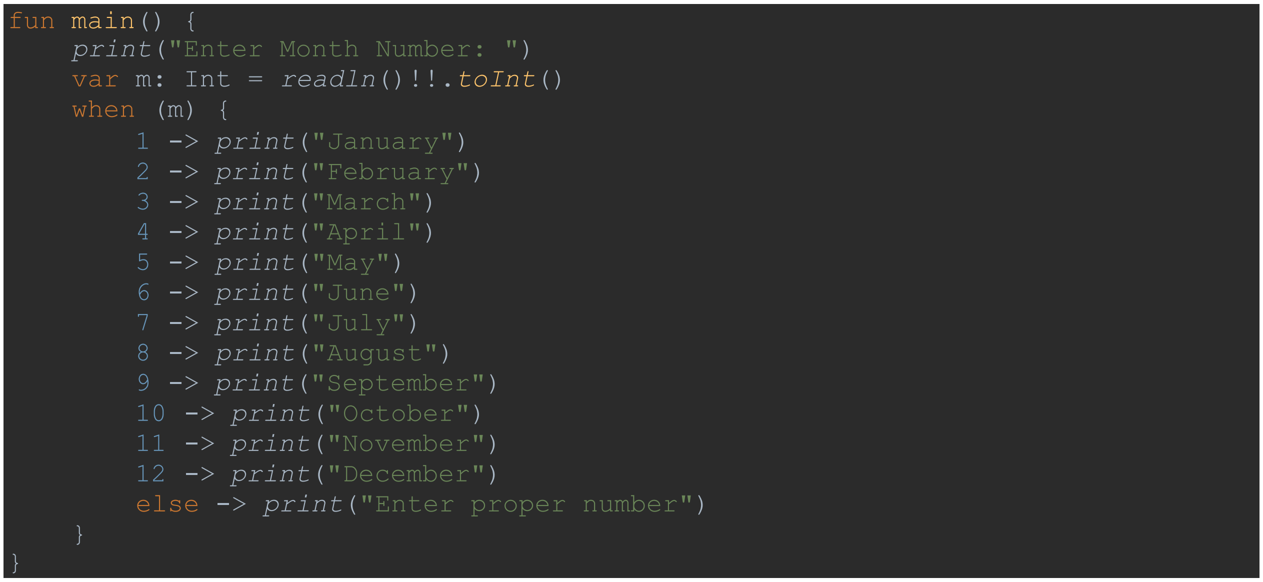
```
Enter number : 2
Even
```



```
Enter number : 3
Odd
```

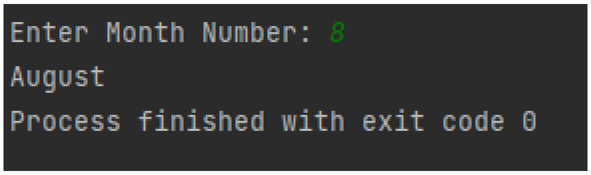
5. Display month name using When

Code:



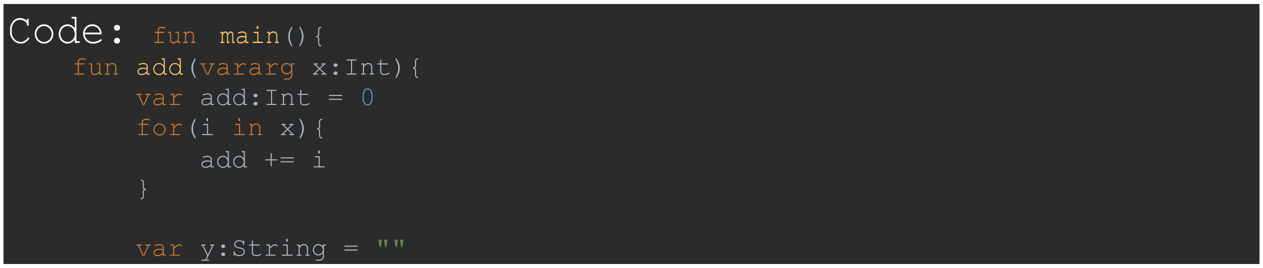
```
fun main() {
    print("Enter Month Number: ")
    var m: Int = readln()!!.toInt()
    when (m) {
        1 -> print("January")
        2 -> print("February")
        3 -> print("March")
        4 -> print("April")
        5 -> print("May")
        6 -> print("June")
        7 -> print("July")
 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

Output:



```
Enter Month Number: 8
August
Process finished with exit code 0
```

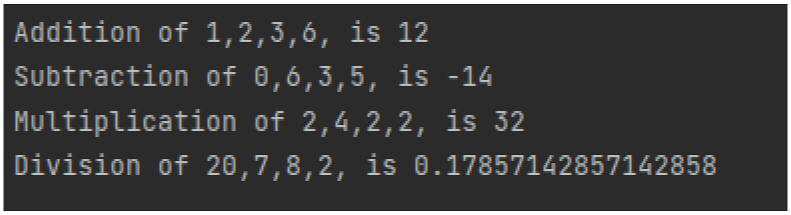
6. By using a user defined function perform all arithmetic operations



Code:

```
fun main(){
    fun add(vararg x:Int){
        var add:Int = 0
        for(i in x){
            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 = 0
        var flag = 0
        for(i in x){
            if (flag == 0){
                flag = 1
                continue
            }
            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 = 1
        for(i in x){
            mul *= i
        }
        var y:String = ""
        for(i in x){
            y += "$i,"
        }
        println("Multiplication of $y is $mul")
    }

    fun div(vararg x:Int) {
        var div:Double = x[0].toDouble()
        for (i in x) {
            if(i == x[0]) {
                continue
            }
            div /= i
        }
        var y: String = ""
        for (i in x) {
            y += "$i,"
        }
        println("Division of $y is $div")
    }
    add(1,2,3,6)
    sub(0,6,3,5)
    mul(2,4,2,2)
    div(20,7,8,2)
}
```

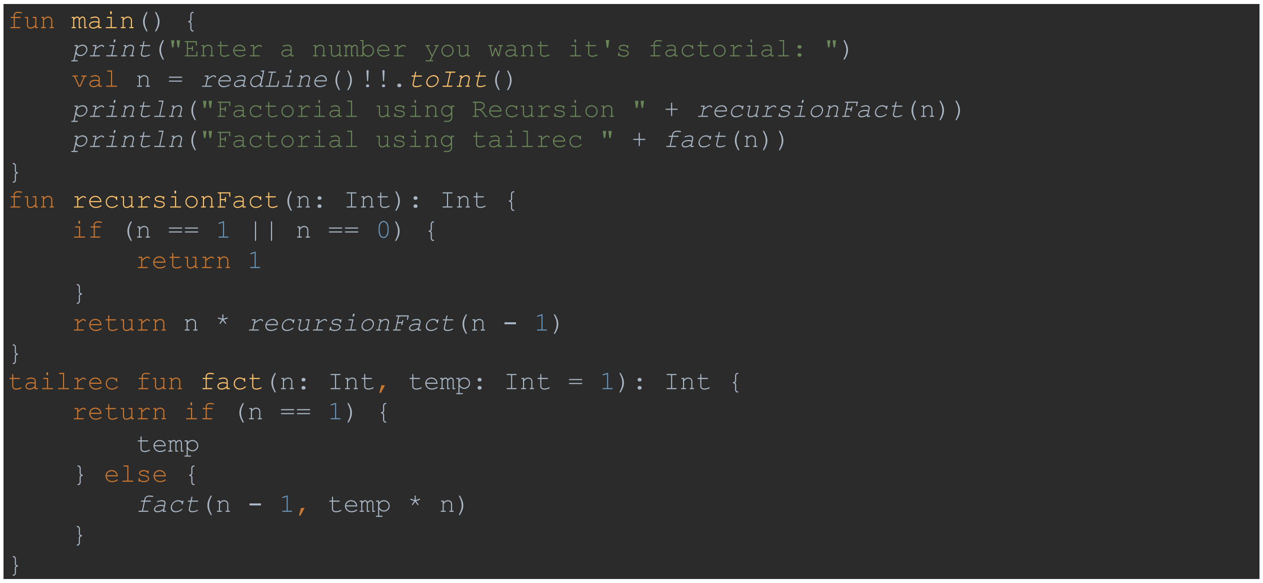
output:



```
Addition of 1,2,3,6, is 12
Subtraction of 0,6,3,5, is -14
Multiplication of 2,4,2,2, is 32
Division of 20,7,8,2, is 0.17857142857142858
```

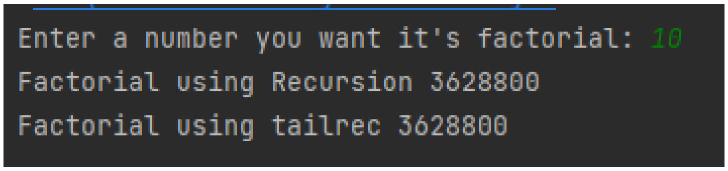
7.Find the factorial of number by recursion. Explain "tailrec" keyword

Code:



```
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:

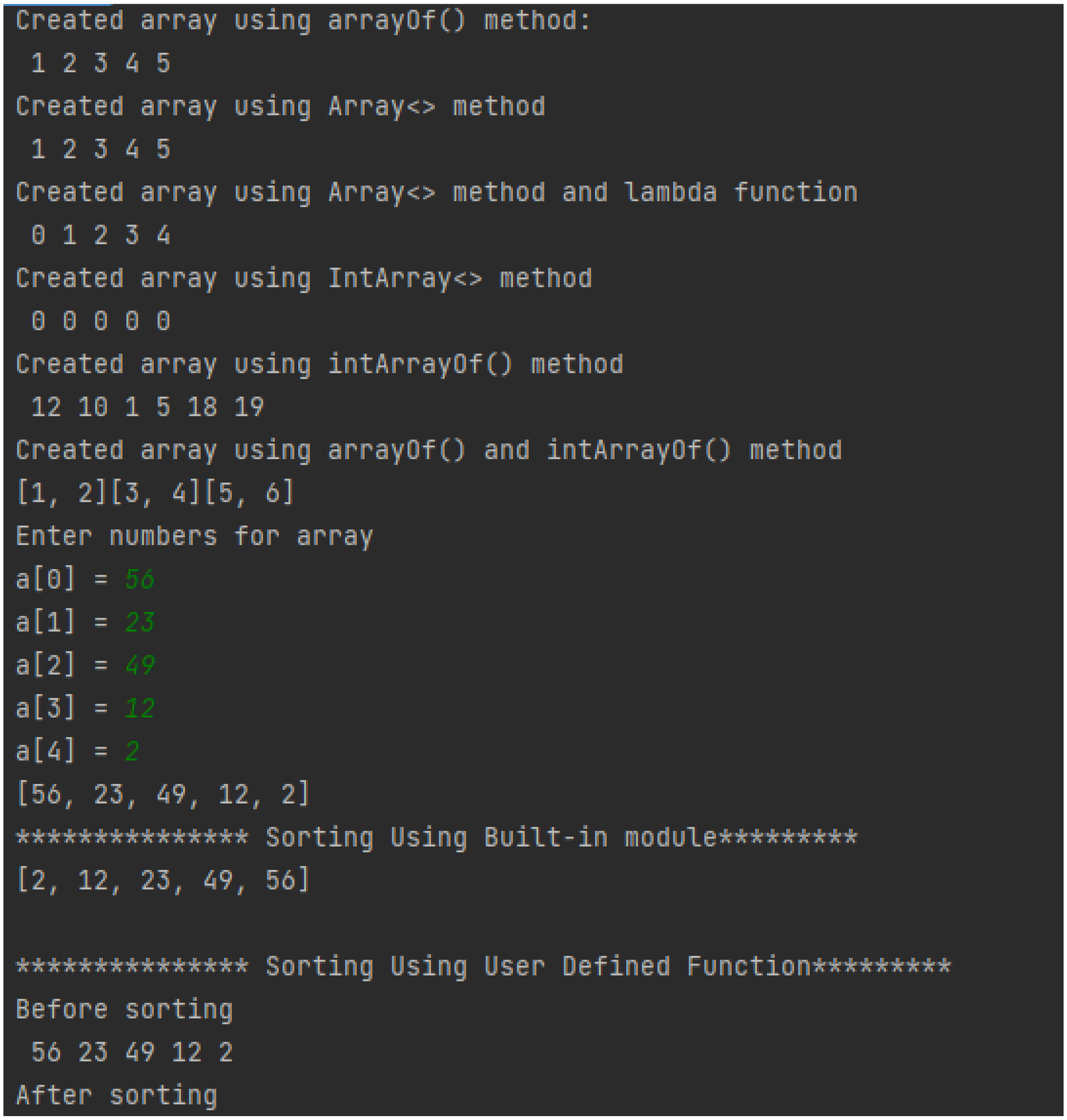


```
Enter a number you want it's factorial: 10
Factorial using Recursion 3628800
Factorial using tailrec 3628800
```

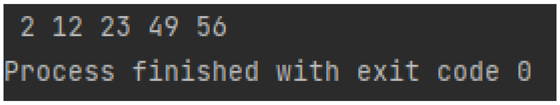
8.Create different types of Array as shown in image. Explore Arrays.deepToString(), contentDeepToString() methods, IntArray variable .joinToString() and use in program to print Array. Explore range, downTo, until etc. for loop and use in this program. Sort Array of Integer data type without using inbuilt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

Code:

Output:



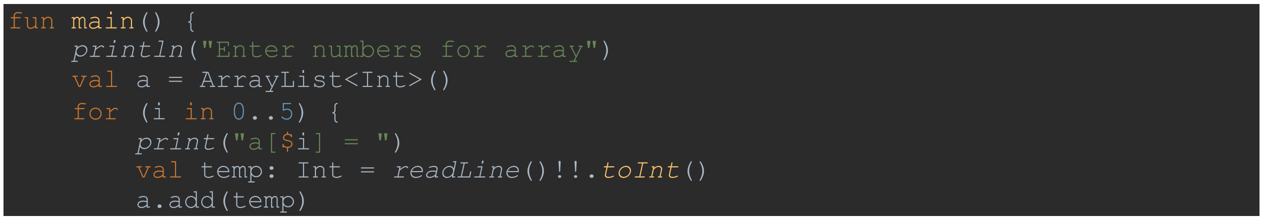
```
Created array using arrayOf() method:
 1 2 3 4 5
Created array using Array<> method
 1 2 3 4 5
Created array using Array<> method and lambda function
 0 1 2 3 4
Created array using IntArray<> method
 0 0 0 0 0
Created array using intArrayOf() method
 12 10 1 5 18 19
Created array using arrayOf() and intArrayOf() method
[1, 2][3, 4][5, 6]
Enter numbers for array
a[0] = 56
a[1] = 23
a[2] = 49
a[3] = 12
a[4] = 2
[56, 23, 49, 12, 2]
*************** Sorting Using Built-in module*********
[2, 12, 23, 49, 56]

*************** Sorting Using User Defined Function*********
Before sorting
 56 23 49 12 2
After sorting
```



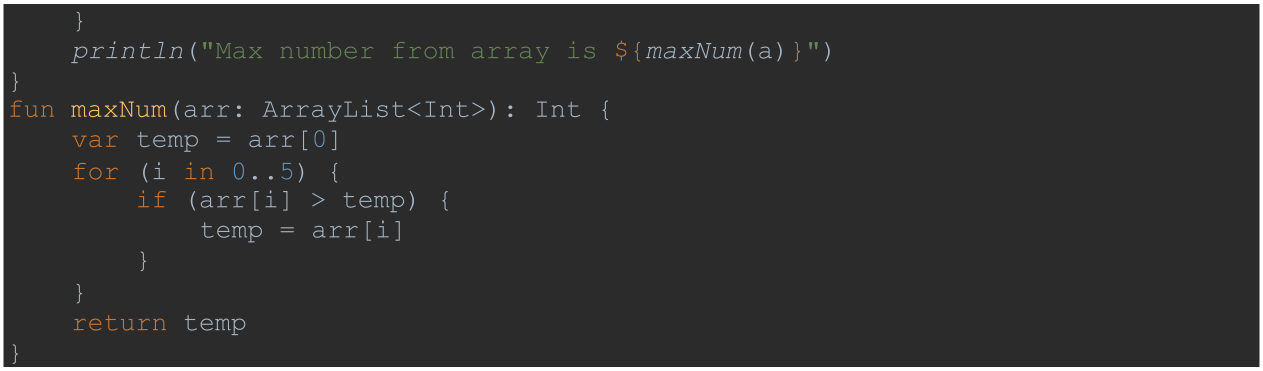
```
 2 12 23 49 56
Process finished with exit code 0
```

9.Find the max number from ArrayList

Code:

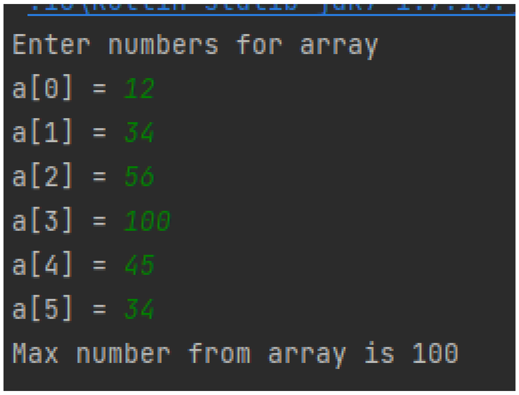


```
fun main() {
    println("Enter numbers for array")
    val a = ArrayList<Int>()
    for (i in 0..5) {
        print("a[$i] = ")
        val temp: Int = readLine()!!.toInt()
        a.add(temp)
```



```
    }
    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
}
```

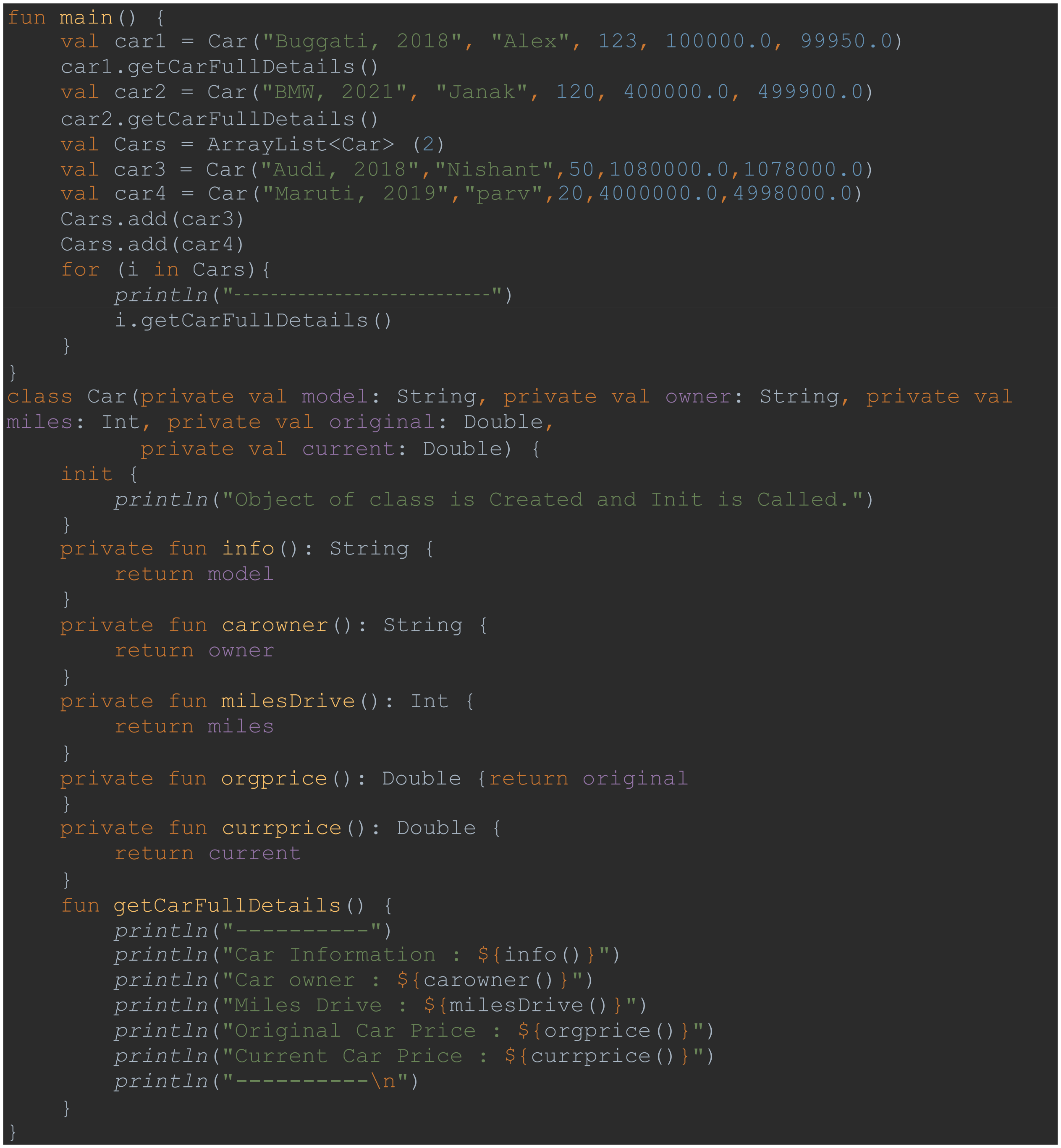
Output:



```
Enter numbers for array
a[0] = 12
a[1] = 34
a[2] = 56
a[3] = 100
a[4] = 45
a[5] = 34
Max number from array is 100
```

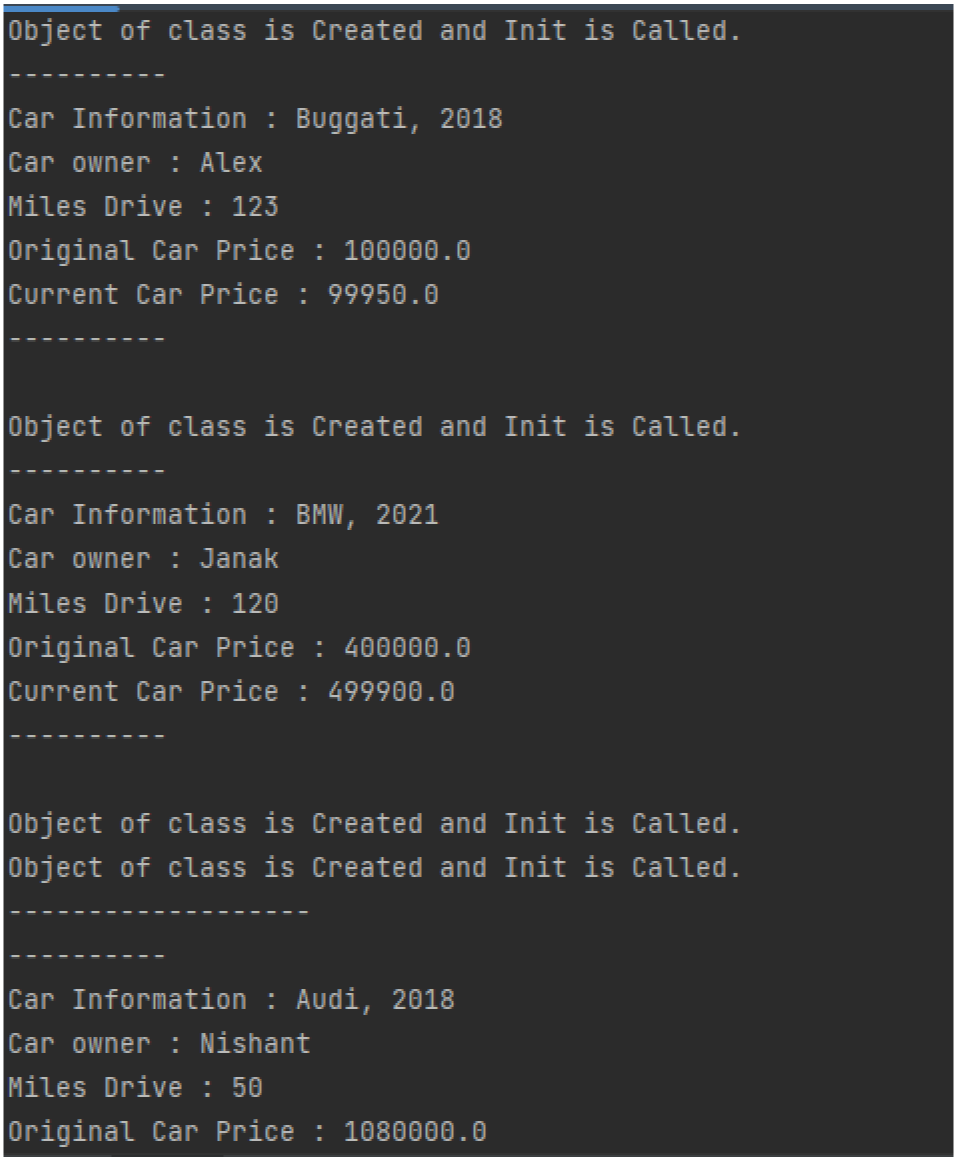
10. Write Different types of Class & Constructor. Create a class Car and set various members like type, model, price, owner, milesDrive. add the function getCarPrice in it. Create an object of Car class and access property of it. (getCarInformation(), getOriginalCarPrice(), getCurrentCarPrice(), displayCarInfo()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

Code:

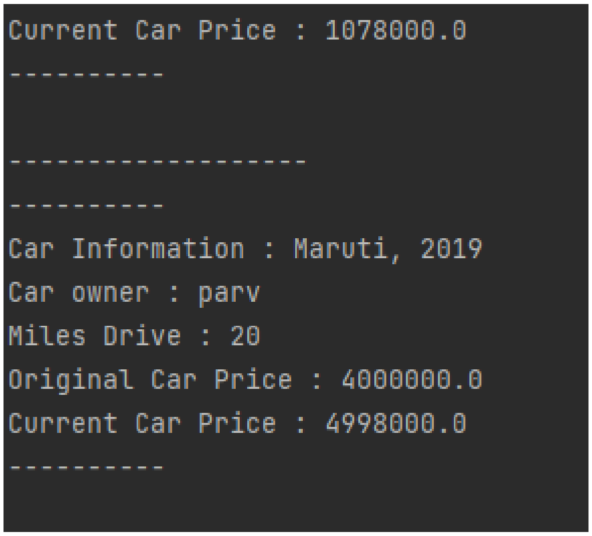


```
fun main() {
    val car1 = Car("Buggati, 2018", "Alex", 123, 100000.0, 99950.0)
    car1.getCarFullDetails()
    val car2 = Car("BMW, 2021", "Janak", 120, 400000.0, 499900.0)
    car2.getCarFullDetails()
    val Cars = ArrayList<Car> (2)
    val car3 = Car("Audi, 2018","Nishant",50,1080000.0,1078000.0)
    val car4 = Car("Maruti, 2019","parv",20,4000000.0,4998000.0)
    Cars.add(car3)
    Cars.add(car4)
    for (i in Cars){
        println("-------------------")
        i.getCarFullDetails()
    }
}
class Car(private val model: String, private val owner: String, private val
miles: Int, private val original: Double,
          private val current: Double) {
    init {
        println("Object of class is Created and Init is Called.")
    }
    private fun info(): String {
        return model
    }
    private fun carowner(): String {
        return owner
    }
    private fun milesDrive(): Int {
        return miles
    }
    private fun orgprice(): Double {return original
    }
    private fun currprice(): Double {
        return current
    }
    fun getCarFullDetails() {
        println("----------")
        println("Car Information : ${info()}")
        println("Car owner : ${carowner()}")
        println("Miles Drive : ${milesDrive()}")
        println("Original Car Price : ${orgprice()}")
        println("Current Car Price : ${currprice()}")
        println("----------\n")
    }
}
```

Output:



```
Object of class is Created and Init is Called.
----------
Car Information : Buggati, 2018
Car owner : Alex
Miles Drive : 123
Original Car Price : 100000.0
Current Car Price : 99950.0
----------

Object of class is Created and Init is Called.
----------
Car Information : BMW, 2021
Car owner : Janak
Miles Drive : 120
Original Car Price : 400000.0
Current Car Price : 499900.0
----------

Object of class is Created and Init is Called.
Object of class is Created and Init is Called.
-------------------
----------
Car Information : Audi, 2018
Car owner : Nishant
Miles Drive : 50
Original Car Price : 1080000.0
```

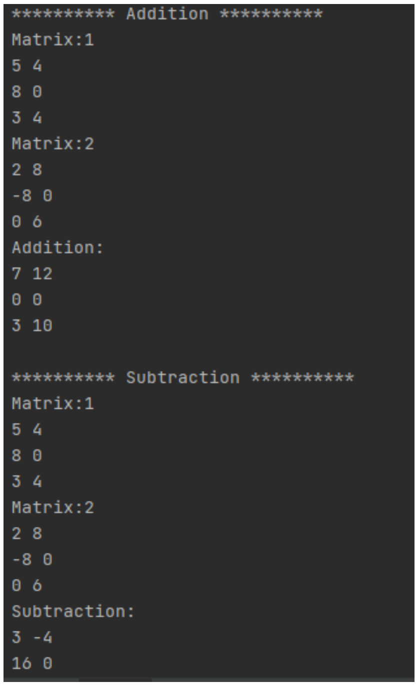


```
Current Car Price : 1078000.0
----------


-------------------
----------
Car Information : Maruti, 2019
Car owner : parv
Miles Drive : 20
Original Car Price : 4000000.0
Current Car Price : 4998000.0
----------
```

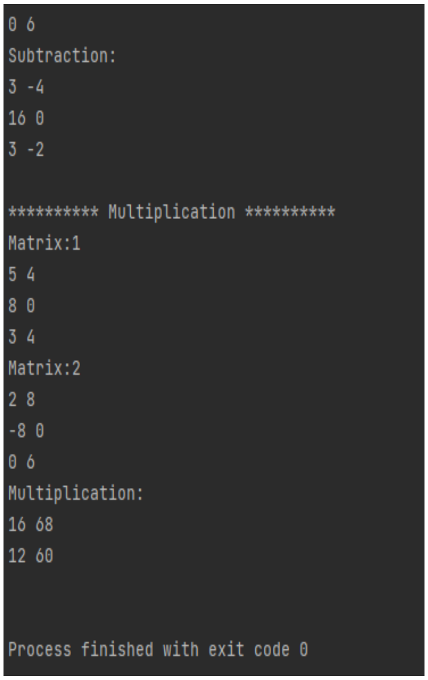
11. Write about Operator Overloading. Perform Matrix Addition, Subtraction & Multiplication using Class Matrix & operator overloading. Overload toString() function in Matrix class.

Code:

Output:



```
********** Addition **********
Matrix:1
5 4
8 0
3 4
Matrix:2
2 8
-8 0
0 6
Addition:
7 12
0 0
3 10

********** Subtraction **********
Matrix:1
5 4
8 0
3 4
Matrix:2
2 8
-8 0
0 6
Subtraction:
3 -4
16 0
```

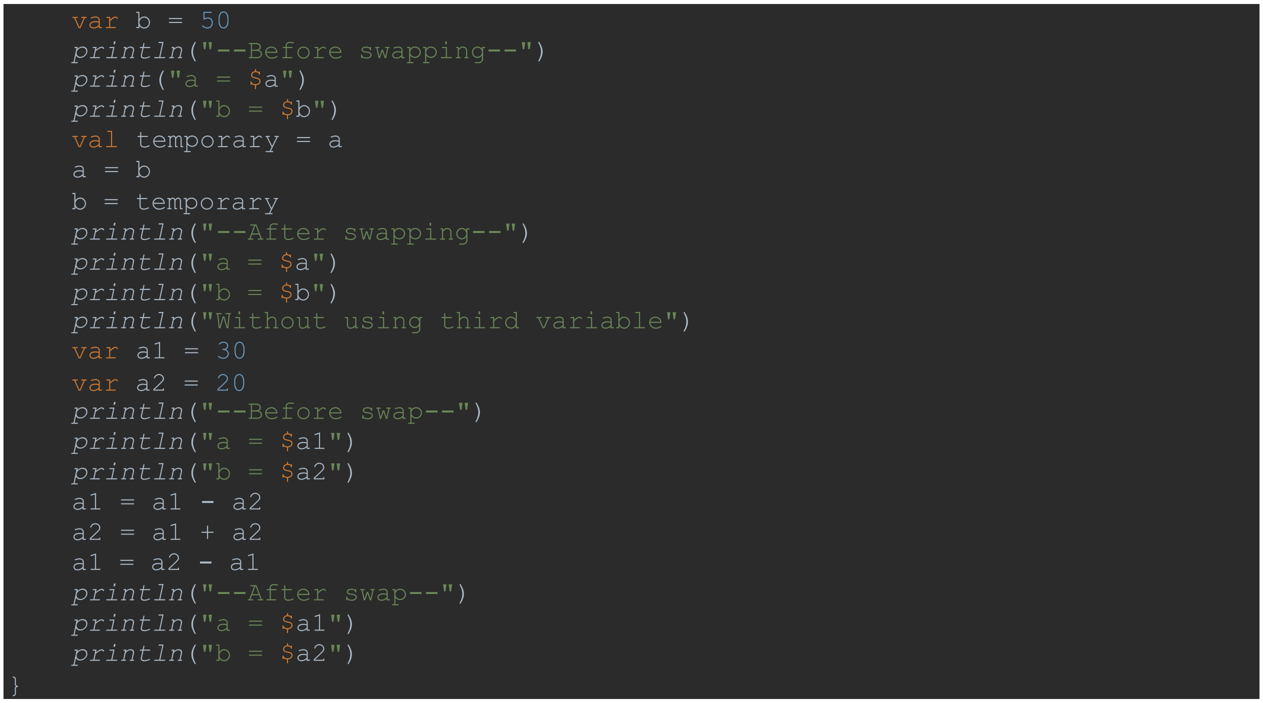


```
0 6
Subtraction:
3 -4
16 0
3 -2

********** Multiplication **********
Matrix:1
5 4
8 0
3 4
Matrix:2
2 8
-8 0
0 6
Multiplication:
16 68
12 60


Process finished with exit code 0
```

Exercise:-

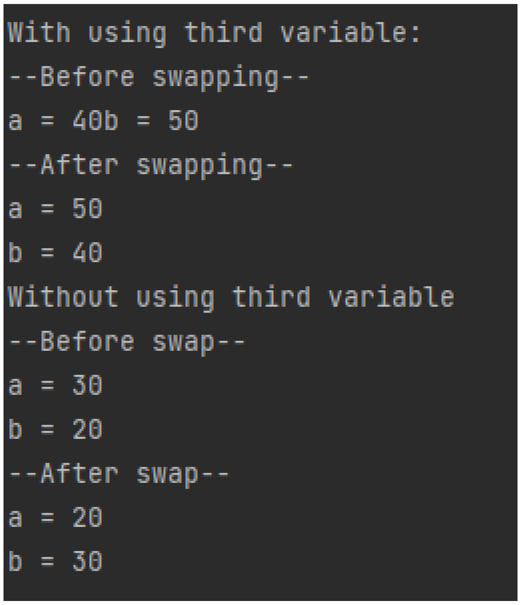
1.Swap Value of two variables without using third variable and with using third variable.

```
Code: fun main() {
    println("With using third variable:")
    var a = 40
```



```
    var b = 50
    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 = 30
    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:



```
With using third variable:
--Before swapping--
a = 40b = 50
--After swapping--
a = 50
b = 40
Without using third variable
--Before swap--
a = 30
b = 20
--After swap--
a = 20
b = 30
```

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.

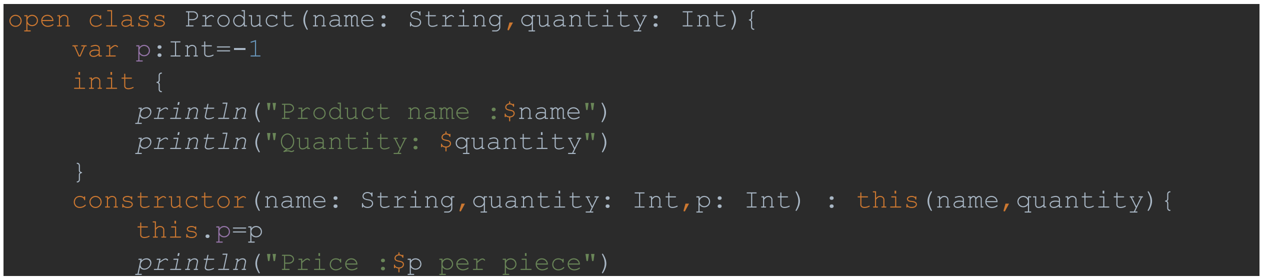
Create primary and secondary Constructor of both class.

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

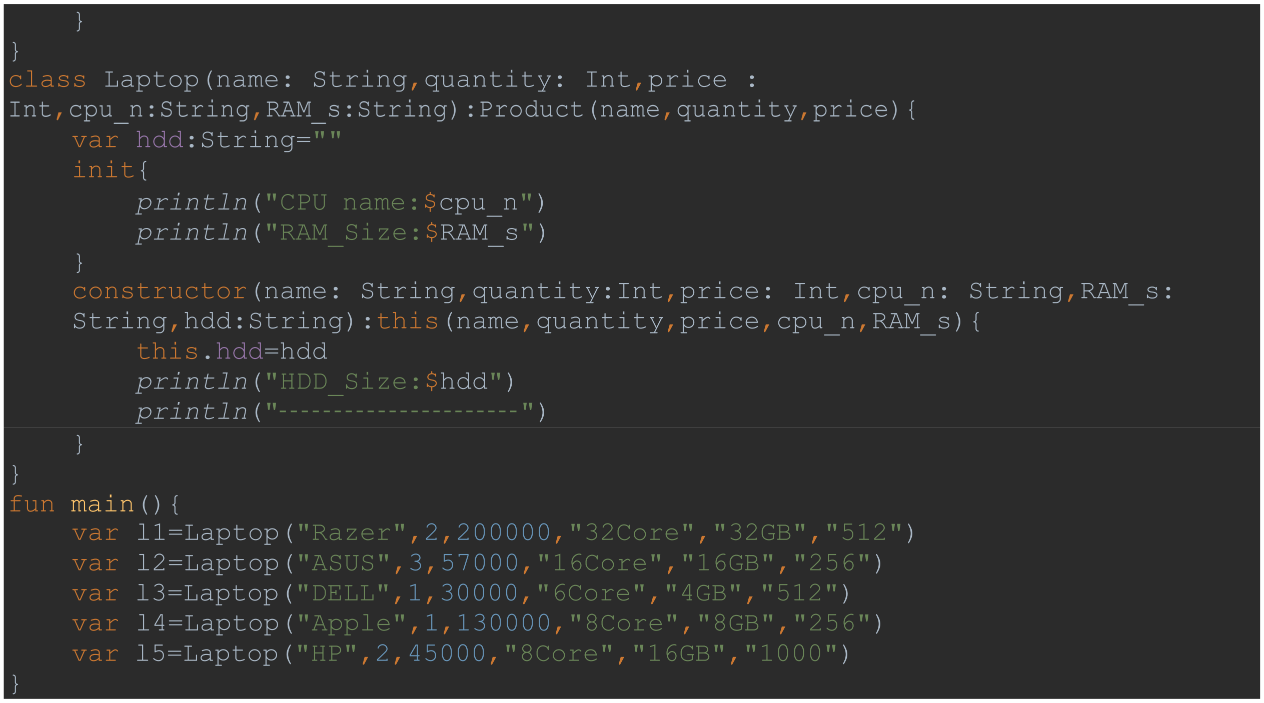
If we can create secondary and primary constructor both in child class then what is restriction if parent have more than two different secondary constructor?

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

CODE:

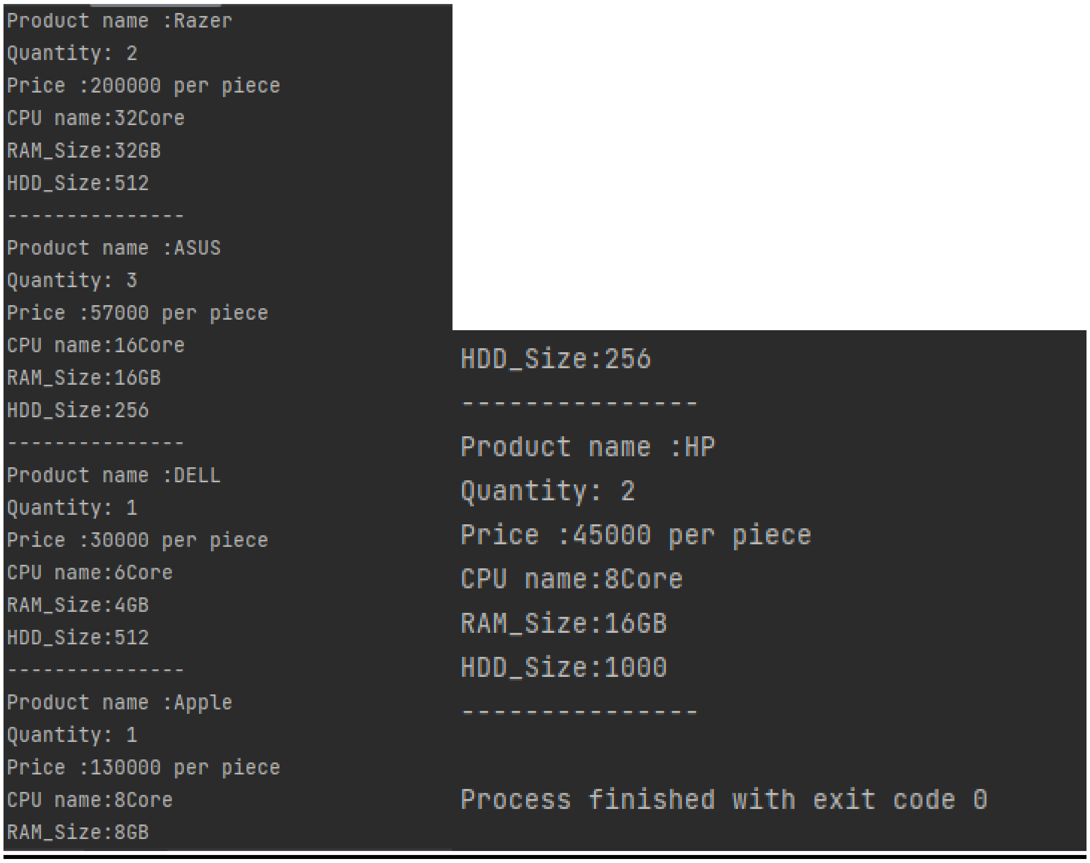


```
open class Product(name: String,quantity: Int){
    var p:Int=-1
    init {
        println("Product name :$name")
        println("Quantity: $quantity")
    }
    constructor(name: String,quantity: Int,p: Int) : this(name,quantity){
        this.p=p
        println("Price :$p per piece")
```



```
    }
}
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("Razer",2,200000,"32Core","32GB","512")
    var l2=Laptop("ASUS",3,57000,"16Core","16GB","256")
    var l3=Laptop("DELL",1,30000,"6Core","4GB","512")
    var l4=Laptop("Apple",1,130000,"8Core","8GB","256")
    var l5=Laptop("HP",2,45000,"8Core","16GB","1000")
}
```

OUTPUT:



```
Product name :Razer
Quantity: 2
Price :200000 per piece
CPU name:32Core
RAM_Size:32GB
HDD_Size:512
---------------
Product name :ASUS
Quantity: 3
Price :57000 per piece
CPU name:16Core
RAM_Size:16GB
HDD_Size:256
---------------
Product name :DELL
Quantity: 1
Price :30000 per piece
CPU name:6Core
RAM_Size:4GB
HDD_Size:512
---------------
Product name :Apple
Quantity: 1
Price :130000 per piece
CPU name:8Core
RAM_Size:8GB
HDD_Size:256
---------------
Product name :HP
Quantity: 2
Price :45000 per piece
CPU name:8Core
RAM_Size:16GB
HDD_Size:1000
---------------

Process finished with exit code 0
```

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.

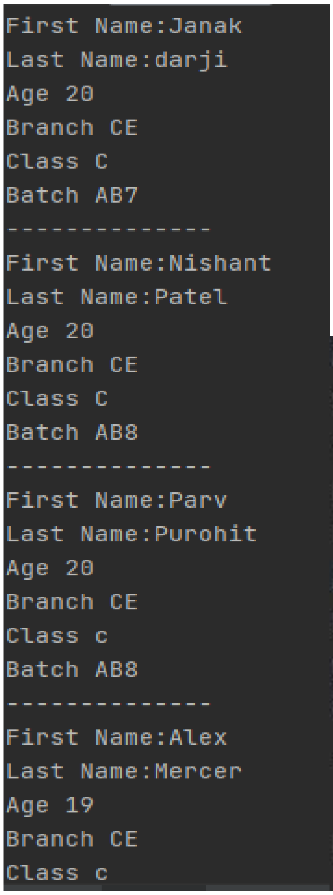
Create primary and secondary Constructor of both class.

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

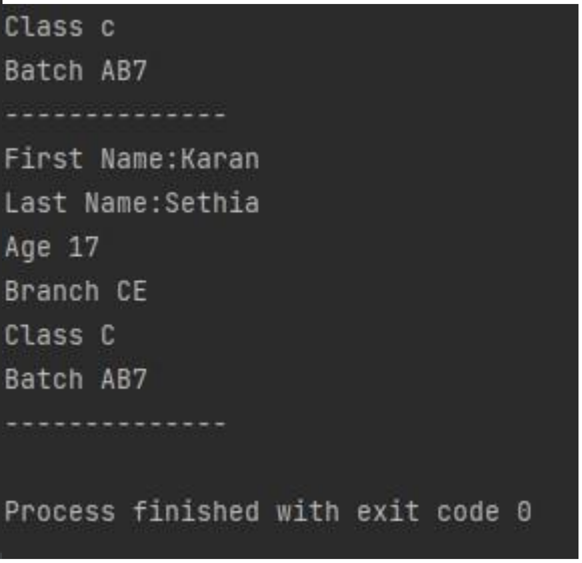
Code:

```
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("Janak","darji",20,20012011029,"CE","C","AB7")
    var s2=Student("Nishant","Patel",20,20012011115,"CE","C","AB8")
    var s3=Student("Parv","Purohit",20,20012011141,"CE","c","AB8")
    var s4=Student("Alex","Mercer",19,20012021011,"CE","c","AB7")
    var s5=Student("Karan","Sethia",17,2001201006,"CE","C","AB7")
}
```

Output:



```
First Name:Janak
Last Name:darji
Age 20
Branch CE
Class C
Batch AB7
--------------
First Name:Nishant
Last Name:Patel
Age 20
Branch CE
Class C
Batch AB8
--------------
First Name:Parv
Last Name:Purohit
Age 20
Branch CE
Class c
Batch AB8
--------------
First Name:Alex
Last Name:Mercer
Age 19
Branch CE
Class c
```



```
Class c
Batch AB7
--------------
First Name:Karan
Last Name:Sethia
Age 17
Branch CE
Class C
Batch AB7
--------------

Process finished with exit code 0
```