[2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]

Practical: 1

AIM - Kotlin Programs

Submitted By: Patel Krupa D.
Enrollment number: 21012021070

Practical: 1

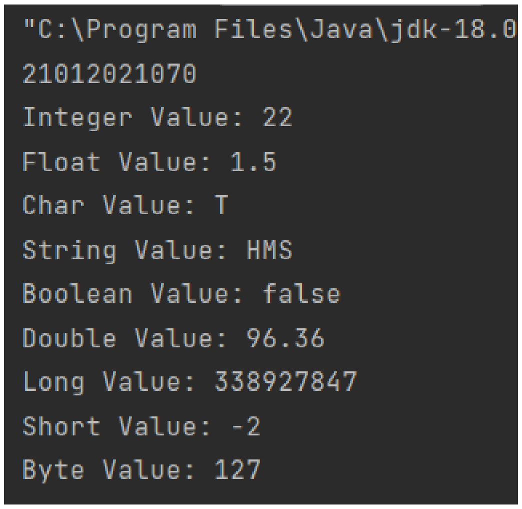
Ganpat University | U.V. Patel College of Engineering
॥ विद्यया समाजोत्कर्षः ॥

Department of
Information Technology

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

```
fun main()
{
println("21012021070")
var a: Int = 22
var b = 1.5
var c: Char = 'T'
var d: String = "HMS"
var e: Boolean = false
var f: Double = 96.36
val g: Long = 338927847
val h: Short = -2
val i: Byte = 127
println("Integer Value: $a")
println("Float Value: $b")
println("Char Value: $c")
println("String Value: $d")
println("Boolean Value: $e")
println("Double Value: $f")
println("Long Value: $g")
println("Short Value: $h")
println("Byte Value: $i")
}
```

Output:



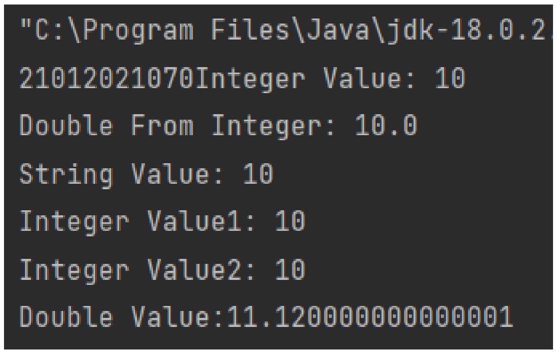
```
"C:\Program Files\Java\jdk-18.0
21012021070
Integer Value: 22
Float Value: 1.5
Char Value: T
String Value: HMS
Boolean Value: false
Double Value: 96.36
Long Value: 338927847
Short Value: -2
Byte Value: 127
```

1.2. **Type conversion:**
Integer to Double, String to Integer, String to Double.

Answer:

```
fun main()
{
 print("21012021070")
 var int_val : Int = 10
 var new_value : Double = int_val.toDouble()
 println("Integer Value: $int_val \nDouble From
Integer: $new_value")
 var string : String = "10"
 var new_string : Int = string.toInt()
 var dou_string : Double = string.toDouble()
 var new = dou_string+1.12
 println("String Value: $string \nInteger Value1:
$new_string \nInteger Value2: $new_string")
 println("Double Value:$new")
}
```

Output:



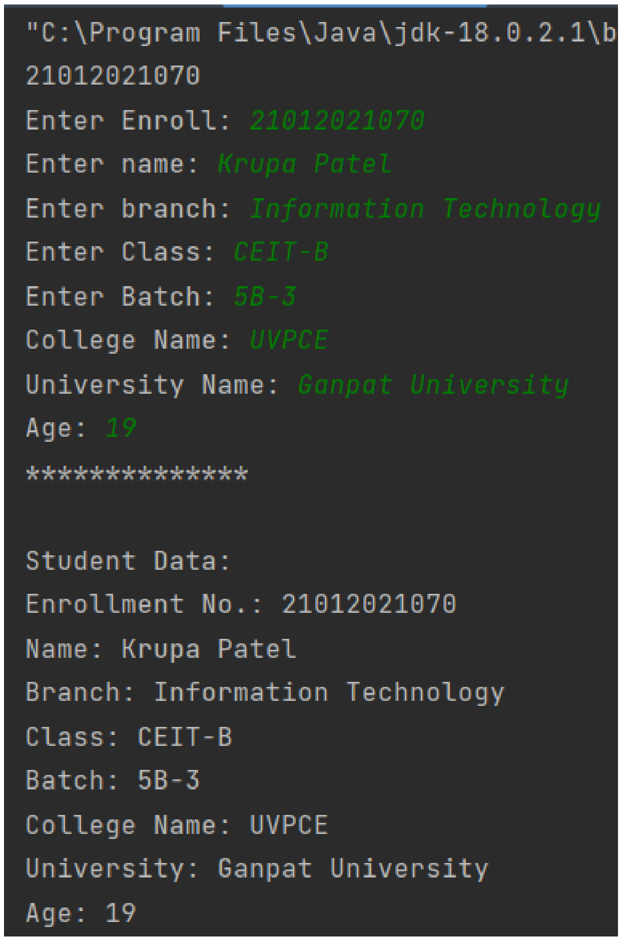
```
"C:\Program Files\Java\jdk-18.0.2
21012021070Integer Value: 10
Double From Integer: 10.0
String Value: 10
Integer Value1: 10
Integer Value2: 10
Double Value:11.120000000000001
```

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

Answer:

```
fun main()
{
    println("21012021070")
    print("Enter Enroll: ")
    var enroll = readLine()
    print("Enter name: ")
    var name= readLine().toString()
    print("Enter branch: ")
    var branch = readLine().toString()
    print("Enter Class: ")
    var Class = readLine()
    print("Enter Batch: ")
    var batch = readLine()
    print("College Name: ")
    var c_name = readLine()
    print("University Name: ")
    var u_name = readLine()
    print("Age: ")
    var age = readLine()
    println("**************\n\nStudent Data:")
    println("Enrollment No.: $enroll")
    println("Name: $name")
    println("Branch: $branch")
    println("Class: $Class")
    println("Batch: $batch")
    println("College Name: $c_name")
    println("University: $u_name")
    println("Age: $age")
}
```

Output:



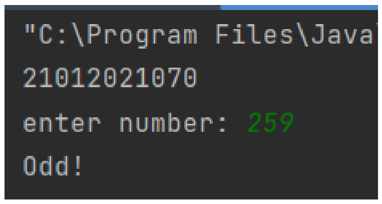
```
"C:\Program Files\Java\jdk-18.0.2.1\b
21012021070
Enter Enroll: 21012021070
Enter name: Krupa Patel
Enter branch: Information Technology
Enter Class: CEIT-B
Enter Batch: 5B-3
College Name: UVPCE
University Name: Ganpat University
Age: 19
**************

Student Data:
Enrollment No.: 21012021070
Name: Krupa Patel
Branch: Information Technology
Class: CEIT-B
Batch: 5B-3
College Name: UVPCE
University: Ganpat University
Age: 19
```

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

Answer:

```
fun main()
{
    println("21012021070")
    print("enter number: ")
    var num = readLine()!!.toInt()
    if(num%2==0)
    {
        println("Even!")
    }
    else
    {
        println("Odd!")
    }
}
```

Output:



```
"C:\Program Files\Java
21012021070
enter number: 259
Odd!
```

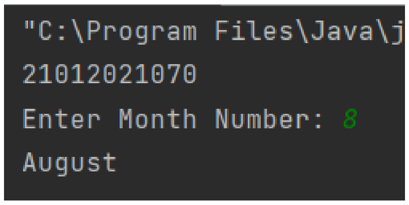
1.5. **Display month name using When.**

Answer:

```
fun main()
{
    println("21012021070")
    print("Enter Month Number: ")
    var num = readLine()
    when(num)
    {
        "1" -> println("January")
```

```
            "2" -> println("February")
            "3" -> println("March")
            "4" -> println("April")
            "5" -> println("May")
            "6" -> println("June")
            "7" -> println("July")
            "8" -> println("August")
            "9" -> println("September")
            "10" -> println("October")
            "11" -> println("November")
            "12" -> println("December")
            else -> println("Enter Valid Number!")
        }
    }
```

Output:



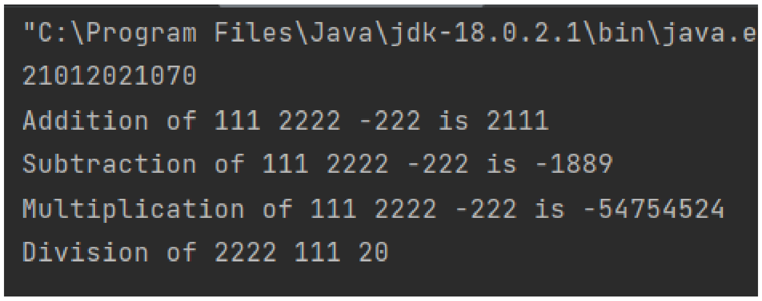
```
"C:\Program Files\Java\j
21012021070
Enter Month Number: 8
August
```

1.6. **By using a user defined function perform all arithmetic operations.**

Answer:

```
fun main()
{
    println("21012021070")
    var a= 111
    var b= 2222
    var c= -222
    var ans = a+b+c
    var ans1 = a-b-c
    var ans2 = a*b*c
    var ans3 = b/a
    println("Addition of $a $b $c is $ans")
    println("Subtraction of $a $b $c is $ans1")
    println("Multiplication of $a $b $c is $ans2")
    println("Division of $b $a $ans3")
}
```

Output:



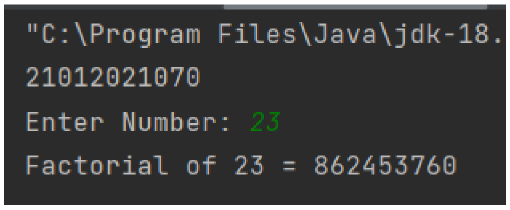
```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.e
21012021070
Addition of 111 2222 -222 is 2111
Subtraction of 111 2222 -222 is -1889
Multiplication of 111 2222 -222 is -54754524
Division of 2222 111 20
```

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

Answer:

```
fun main()
{
    println("21012021070")
    print("Enter Number: ")
    var number = readLine()!!.toInt()
    val factorial = fact(number)
    println("Factorial of $number = $factorial")
}
    tailrec fun fact(n: Int, temp: Int = 1): Int {
    return if (n == 1){
        temp
    }
    else {
        fact(n-1,temp*n)
         }
}
```

Output:



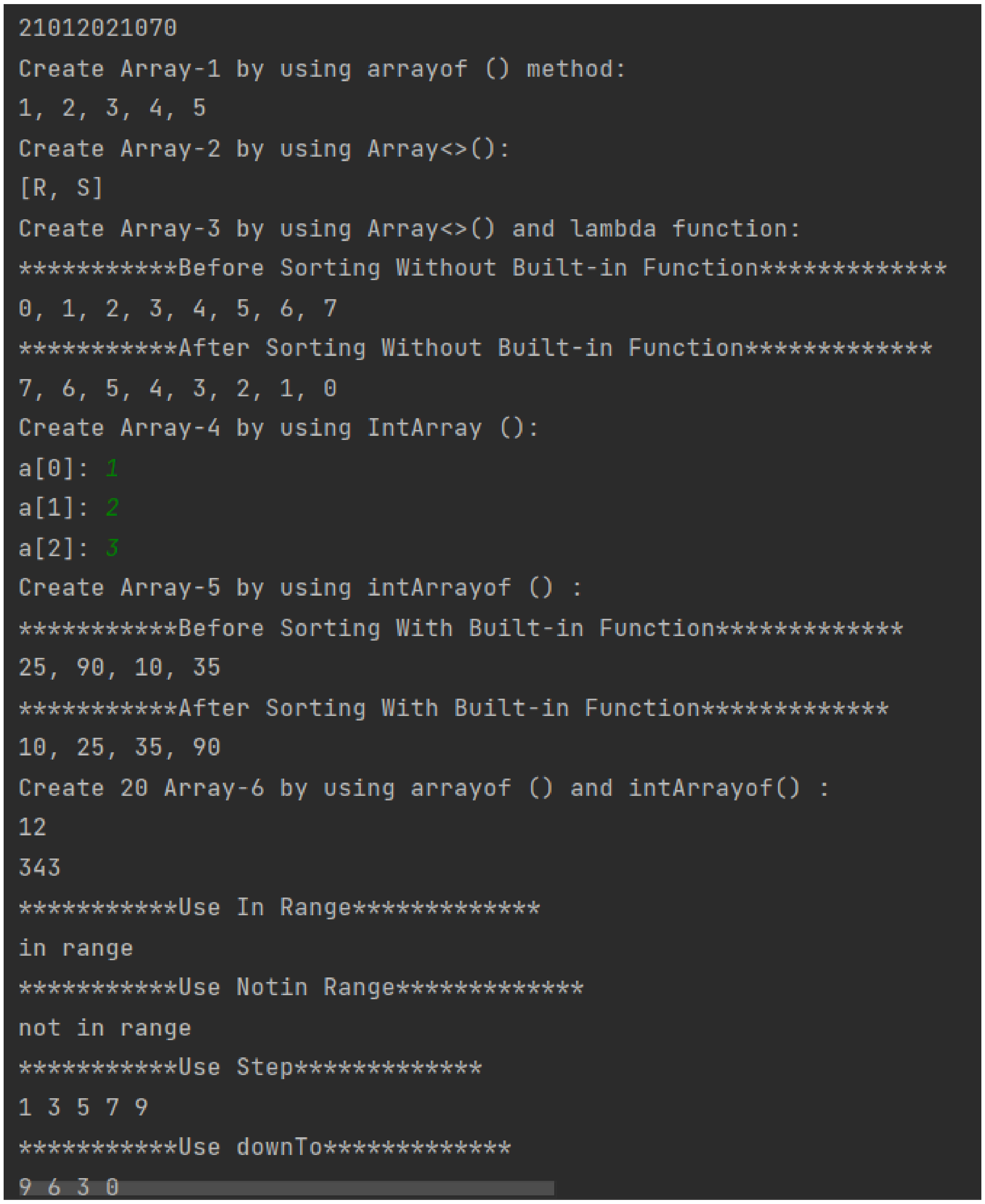
```
"C:\Program Files\Java\jdk-18.
21012021070
Enter Number: 23
Factorial of 23 = 862453760
```

1.8.**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.**

Answer:

```
fun main()
{
    println("21012021070")
    println("Create Array-1 by using arrayof ()
method:")
    var arr1 = arrayOf(1,2,3,4,5)
    println(arr1.joinToString())
    println("Create Array-2 by using Array<>():")
    var arr2 = arrayOf<String>("R","S")
    println(arr2.contentDeepToString())
    println("Create Array-3 by using Array<>() and
lambda function:")
    var arr3 = Array(8){i:Int->i}
    println("**********Before Sorting Without
Built-in Function************")
    println(arr3.joinToString())
    for(i in 0 until arr3.size)
    {
        for(j in i+1 until arr3.size)
        {
            if(arr3[i]<arr3[j])
            {
                arr3[j]=arr3[j]+arr3[i]
                arr3[i]=arr3[j]-arr3[i]
                arr3[j]=arr3[j]-arr3[i]
            }
        }
    }
    println("***********After Sorting Without Built-
in Function************")
    println(arr3.joinToString())
    println("Create Array-4 by using IntArray ():")
    var arr4 = IntArray(3)
    for(i in 0 until arr4.size)
    {
        print("a[$i]: ")
        arr4[i]= readLine()!!.toInt()
    }
    println("Create Array-5 by using intArrayof ()
:")
    var arr5 = intArrayOf(25,90,10,35)
    println("***********Before Sorting With Built-in
Function************")
    println(arr5.joinToString())
    arr5.sort()
    println("***********After Sorting With Built-in
Function************")
    println(arr5.joinToString())
    println("Create 20 Array-6 by using arrayof ()
and intArrayof() :")
    var arr6 = arrayOf(intArrayOf(1,2),
intArrayOf(3,4,3))
    for(i in 0 until arr6.size)
    {
        for(j in 0 until arr6[i].size)
        {
            print(arr6[i][j])
        }
        println()
    }
    val num = 10
    println("**********Use In Range************")
    if (num in 5..10) {
        println("in range")
    }
    println("**********Use Notin
Range************")
    if(num !in 5 .. 9)
    {
        println("not in range")
    }
    println("**********Use Step************")
    for (x in 1..10 step 2) {
        print("$x ")
    }
    println()
    println("**********Use downTo************")
    for (x in 9 downTo 0 step 3) {
        print("$x ")
    }
    println()
}
```

Output:



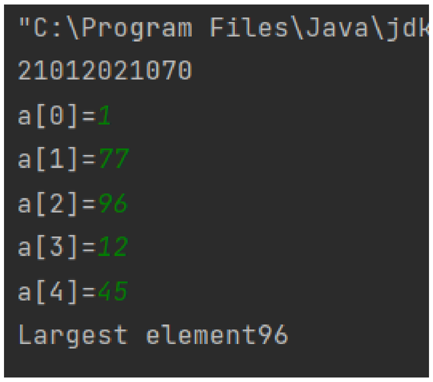
```
21012021070
Create Array-1 by using arrayof () method:
1, 2, 3, 4, 5
Create Array-2 by using Array<>():
[R, S]
Create Array-3 by using Array<>() and lambda function:
***********Before Sorting Without Built-in Function*************
0, 1, 2, 3, 4, 5, 6, 7
***********After Sorting Without Built-in Function*************
7, 6, 5, 4, 3, 2, 1, 0
Create Array-4 by using IntArray ():
a[0]: 1
a[1]: 2
a[2]: 3
Create Array-5 by using intArrayof () :
***********Before Sorting With Built-in Function*************
25, 90, 10, 35
***********After Sorting With Built-in Function*************
10, 25, 35, 90
Create 20 Array-6 by using arrayof () and intArrayof() :
12
343
***********Use In Range*************
in range
***********Use Notin Range*************
not in range
***********Use Step*************
1 3 5 7 9
***********Use downTo*************
9 6 3 0
```

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

Answer:

```
fun main() {
    println("21012021070")
    var array = Array<Int>(5){0}
    var x:Int = 0
    val abc:Int = array.size
    while( x < abc)
    {
        print("a[$x]=")
        array[x] = readLine()!!.toInt()
        x++
    }
    var largest = array[0]
    for (num in array) {
        if (largest < num)
            largest = num
    }
    println("Largest element$largest")
}
```

Output:



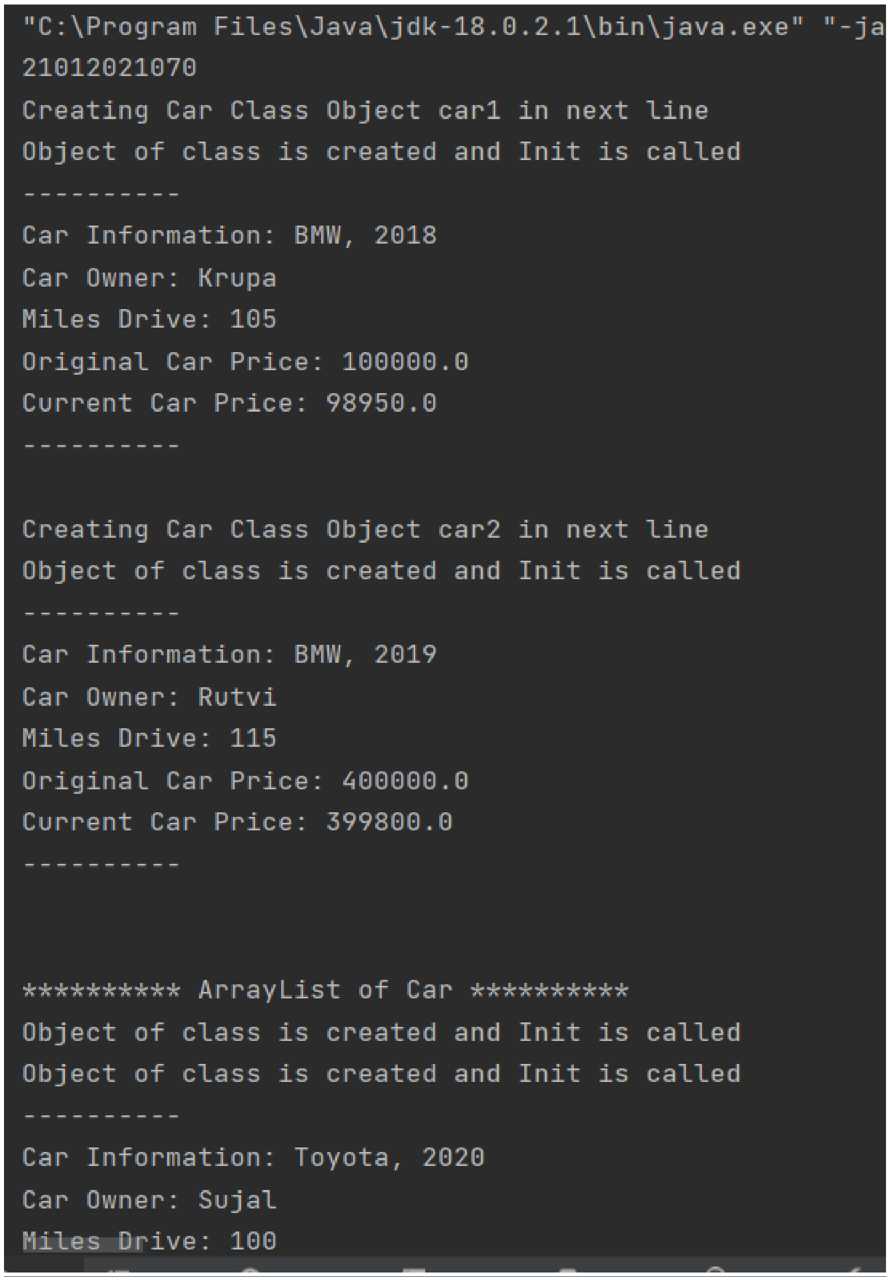
```
"C:\Program Files\Java\jdk
21012021070
a[0]=1
a[1]=77
a[2]=96
a[3]=12
a[4]=45
Largest element96
```

1.10. **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.)**

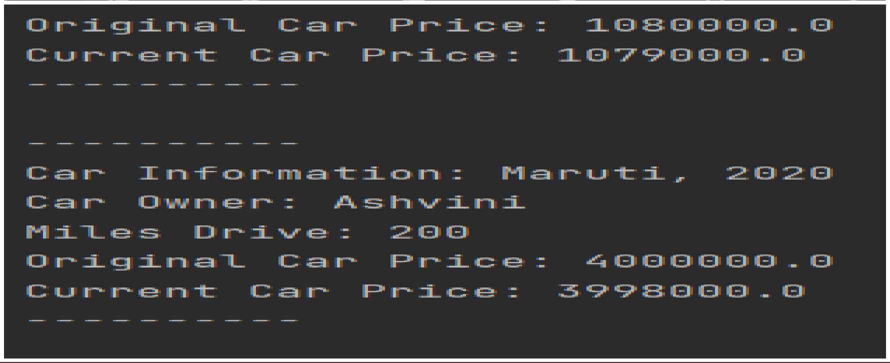
Answer:

```
class
Car(info:String,owner:String,miles:Int,originalPrice
:Double,currentPrice:Double){
        lateinit var info:String
    lateinit var owner:String
    var miles:Int
    var originalPrice:Double
    var currentPrice:Double
    init{
        println("Object of class is created and Init
is called")
        this.info = info
        this.owner = owner
        this.miles = miles
        this.originalPrice = originalPrice
        this.currentPrice = currentPrice
    }
    fun getCarInformation(): String {
        return info
    }
    fun getOriginalCarPrice(): Double {
        return originalPrice
    }
    fun getCurrentCarPrice(): Double {
        return currentPrice
    }
    fun displayCarInfo(){
        println("----------")
        println("Car Information:
${getCarInformation()}")
        println("Car Owner: $owner")
        println("Miles Drive: $miles")
        println("Original Car Price:
${getOriginalCarPrice()}")
        println("Current Car Price:
${getCurrentCarPrice()}")
        println("----------\n")
    }
}
fun main() {
    println("21012021070")
    println("Creating Car Class Object car1 in next
line")
    val car1 = Car("BMW, 2018","Krupa",105,
100000.0,98950.0)
    car1.displayCarInfo()
    println("Creating Car Class Object car2 in next
line")
    val car2 = Car("BMW, 2019","Rutvi",115,
400000.0,399800.0)
    car2.displayCarInfo()
    println("\n********** ArrayList of Car
**********")
    val carlist = ArrayList<Car>()
    carlist.add(Car("Toyota,
2020","Sujal",100,1080000.0,1079000.0))
    carlist.add(Car("Maruti,
2020","Ashvini",200,4000000.0,3998000.0))
    carlist[0].displayCarInfo()
    carlist[1].displayCarInfo()
    }
```

Output:



```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-ja
21012021070
Creating Car Class Object car1 in next line
Object of class is created and Init is called
----------
Car Information: BMW, 2018
Car Owner: Krupa
Miles Drive: 105
Original Car Price: 100000.0
Current Car Price: 98950.0
----------

Creating Car Class Object car2 in next line
Object of class is created and Init is called
----------
Car Information: BMW, 2019
Car Owner: Rutvi
Miles Drive: 115
Original Car Price: 400000.0
Current Car Price: 399800.0
----------



********** ArrayList of Car **********
Object of class is created and Init is called
Object of class is created and Init is called
----------
Car Information: Toyota, 2020
Car Owner: Sujal
Miles Drive: 100
```



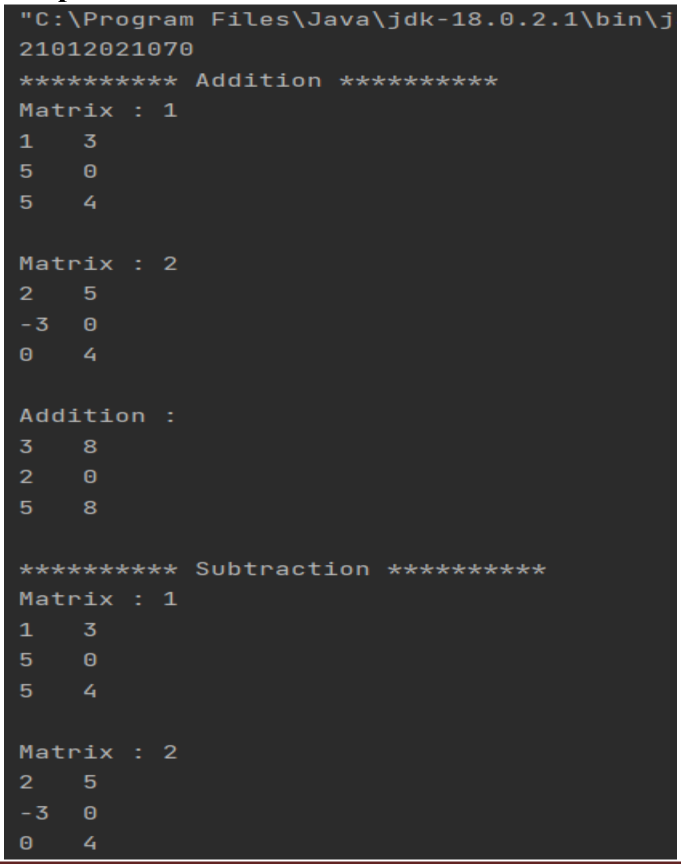
```
Original Car Price: 1080000.0
Current Car Price: 1079000.0
----------

----------
Car Information: Maruti, 2020
Car Owner: Ashvini
Miles Drive: 200
Original Car Price: 4000000.0
Current Car Price: 3998000.0
----------
```

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

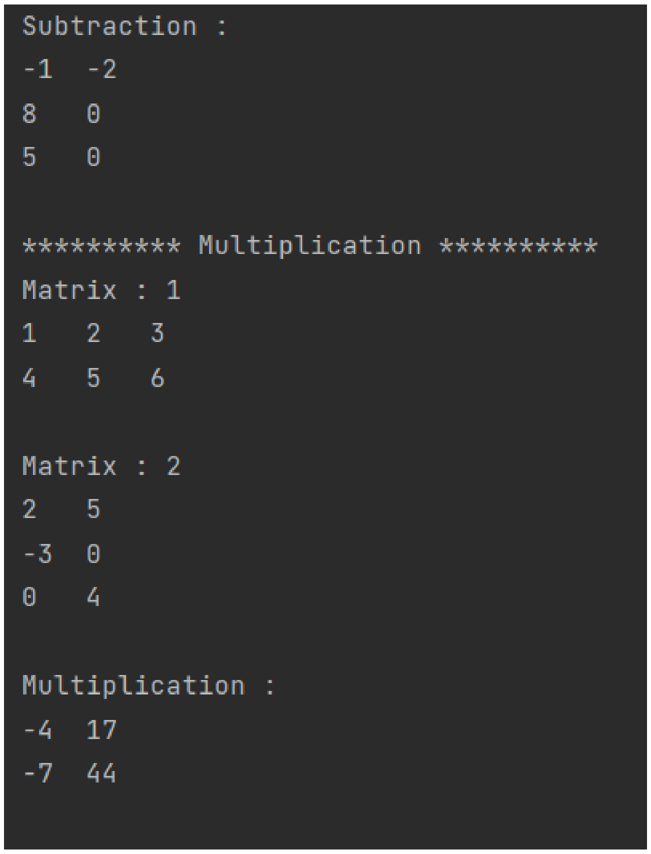
Answer:

```
class Matrix(var data: Array<IntArray>) {
    val rows: Int = data.size
    val columns: Int = data[0].size
    operator fun plus(other: Matrix): Matrix {
        val resultData  = Array(rows)
{IntArray(columns)}
        for (i in 0 until rows) {
            for (j in 0 until columns) {
                resultData[i][j] = data[i][j] +
other.data[i][j]
            }
        }
        return Matrix(resultData)
    }
    operator fun minus(other: Matrix): Matrix {
        val resultData  = Array(rows)
{IntArray(columns)}
        for (i in 0 until rows) {
            for (j in 0 until columns) {
                resultData[i][j] = data[i][j] -
other.data[i][j]
            }
        }
        return Matrix(resultData)
    }
    operator fun times(other: Matrix): Matrix {
        val resultData  = Array(rows)
{IntArray(other.columns) {0} }
        for (i in 0 until rows) {
            for (j in 0 until other.columns) {
                for (k in 0 until columns)
                    resultData[i][j] += data[i][k] *
other.data[k][j]
            }
        }
        return Matrix(resultData)
    }
    override fun toString(): String {
        val sb = StringBuilder()
        for (i in 0 until rows) {
            for (j in 0 until columns) {
                sb.append("${data[i][j]}\t")
            }
            sb.append("\n")
        }
        return sb.toString()
    }
}
fun main() {
    val firstMatrix = Matrix(arrayOf(intArrayOf(1,
2, 3), intArrayOf(4, 5, 6)))
    val secondMatrix = Matrix(arrayOf(intArrayOf(2,
5), intArrayOf(-3, 0), intArrayOf(0, 4)))
    val secondMatrix1 = Matrix(arrayOf(intArrayOf(1,
3), intArrayOf(5, 0), intArrayOf(5, 4)))
    println("21012021070")
    println("********** Addition **********")
    println("Matrix : 1 ")
    println(secondMatrix1)
    println("Matrix : 2 ")
    println(secondMatrix)
    val thirdMatrix = secondMatrix1 + secondMatrix
    println("Addition : ")
    println(thirdMatrix)
    println("********** Subtraction **********")
    println("Matrix : 1 ")
    println(secondMatrix1)
    println("Matrix : 2 ")
    println(secondMatrix)
    val subtractMatrix = secondMatrix1 -
secondMatrix
    println("Subtraction : ")
    println(subtractMatrix)
    println("********** Multiplication **********")
    println("Matrix : 1 ")
    println(firstMatrix)
    println("Matrix : 2 ")
    println(secondMatrix)
    val multiplication = firstMatrix * secondMatrix
    println("Multiplication : ")
    println(multiplication)
}
```

Output:



```
"C:\Program Files\Java\jdk-18.0.2.1\bin\j
21012021070
********** Addition **********
Matrix : 1
1   3
5   0
5   4

Matrix : 2
2   5
-3  0
0   4

Addition :
3   8
2   0
5   8

********** Subtraction **********
Matrix : 1
1   3
5   0
5   4

Matrix : 2
2   5
-3  0
0   4
```



```
Subtraction :
-1  -2
8   0
5   0

********** Multiplication **********
Matrix : 1
1   2   3
4   5   6

Matrix : 2
2   5
-3  0
0   4

Multiplication :
-4  17
-7  44
```