[2CEIT5PE18: MOBILE APPLICATION DEVELOPMENT]

Practical: 1

AIM- Develop a Kotlin program for demonstrating various programming concepts.

Submitted By: KANSAGARA KRISH KALPESHBHAI
Enrollment number: 23012011026

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

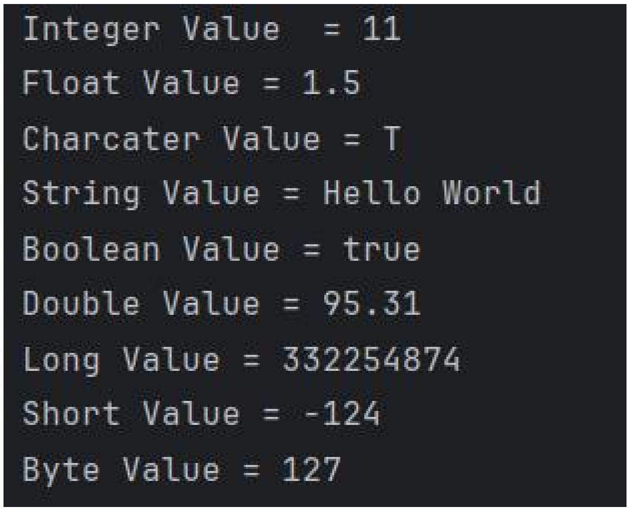
Department of Computer Engineering/Information Technology

1.1. Store & Display Values in Different Variables: Create and display variables of different data types, including Integer, Double, Float, Long, Short, Byte, Char, Boolean, and String

Answer:

```
fun main(){
    val intval : Int = 11
    println("Integer Value  = $intval")
    val floatval : Float = 1.5f
    println("Float Value = $floatval")
    val charValue : Char = 'T'
    println("Charcater Value = $charValue")
    val stringValue : String = "Hello World"
    println("String Value = $stringValue")
    val boolValue : Boolean = true
    println("Boolean Value = $boolValue")
    val doubleValue : Double = 95.31
    println("Double Value = $doubleValue")
    val longValue : Long = 332254874
    println("Long Value = $longValue")
    val shortValue : Short = -124
    println("Short Value = $shortValue")
    val byteValue : Byte = 127
    println("Byte Value = $byteValue")
}
```

Output:



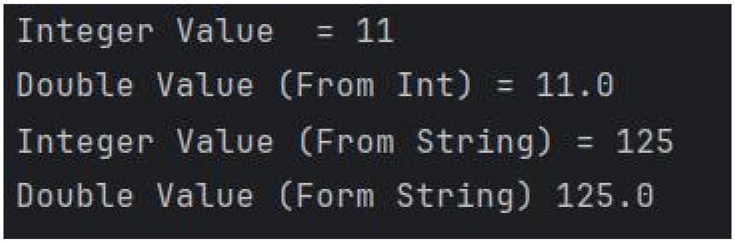
```
Integer Value  = 11
Float Value = 1.5
Charcater Value = T
String Value = Hello World
Boolean Value = true
Double Value = 95.31
Long Value = 332254874
Short Value = -124
Byte Value = 127
```

1.2.Store & Display Values in Different Variables: Create and display variables of different data types, including Integer, Double, Float, Long, Short, Byte, Char, Boolean, and String

Answer:

```
fun main(){
    var intvalue : Int = 11
    println("Integer Value  = $intvalue")
    var doubleValue : Double = intvalue.toDouble()
    println("Double Value (From Int) = $doubleValue")
    var stringValue : String = "125"
    var intValue : Int = stringValue.toInt()
    println("Integer Value (From String) =
$intValue")
    var doubleString : Double =
stringValue.toDouble()
    println("Double Value (Form String)
$doubleString")
}
```

Output:



```
Integer Value  = 11
Double Value (From Int) = 11.0
Integer Value (From String) = 125
Double Value (Form String) 125.0
```

1.3.Scan student’s information and display all the data: Input and display data of students, including their name, enrolment no, branch,etc.

Answer:

```
fun main(){
    print("Student Enrollnment No : ")
    val studentTEnroll = readln()

    print("Student Name : ")
    val studentName = readln()

    print("Student Branch : ")
    val studentBranch = readln()

    print("Student Class : ")
    val studentClass = readln()

    print("Student Batch : ")
    val studentBatch = readln()

    print("Student Collage Name : ")
    val studentCollage = readln()

    print("Student University Name : ")
    val studentUniversity = readln()

    print("Student Age : ")
    val studentAge = readln().toInt()
    println("*********************")
    println("Student Data : ")
    println("Enrollment Number : $studentTEnroll")
    println("Student Name : $studentName")
    println("Student Age : $studentAge")
    println("Student Branch : $studentBranch")
    println("Student Class : $studentClass")
    println("Student Batch : $studentBatch")
    println("Student Collage : $studentCollage")
    println("Student University : 
$studentUniversity")




}
```

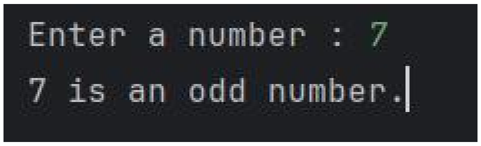
Output:

1.4.Check Odd or Even Numbers: Determine whether a number is odd or even using control flow within println() method.

Answer:

```
fun main(){
    print("Enter a number : ")
    val number = readln().toInt()
    if (number % 2 == 0) {
        println("$number is an even number.")
    } else {
        println("$number is an odd number.")
    }
}
```

Output:



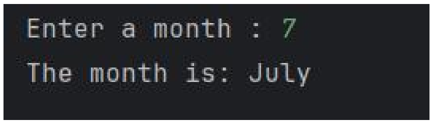
```
Enter a number : 7
7 is an odd number.
```

1.5. Display Month Name: Use a when expression to display the month name based on user input.

Answer:

```
fun main(){
    print("Enter a month : ")
    val month = readLine()?.toInt()

    val monthname = when (month){
        1 -> "January"
        2 -> "February"
        3 -> "March"
        4 -> "April"
        5 -> "May"
        6 -> "June"
        7 -> "July"
        8 -> "August"
        9 -> "September"
        10 -> "October"
        11 -> "November"
        12 -> "December"
        else -> "Invalid month"
    }
    println("The month is: $monthname")
}
```

Output:



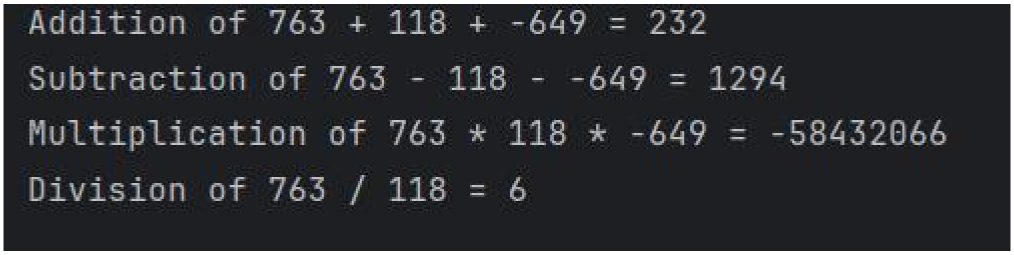
```
Enter a month : 7
The month is: July
```

1.6.User-Defined Function: Create a user-defined function to perform arithmetic operations (addition, subtraction, multiplication, division) on two numbers

Answer:

```
fun add(a: Int, b: Int, c: Int): Int {
    return a + b + c
}

fun subtract(a: Int, b: Int, c: Int): Int {
    return a - b - c
}

fun multiply(a: Int, b: Int, c: Int): Int {
    return a * b * c
}

fun divide(a: Int, b: Int): Int {
    return a / b
}

fun main() {
    val a = 763
    val b = 118
    val c = -649

    println("Addition of $a + $b + $c = ${add(a, b,
c)}")
    println("Subtraction of $a - $b - $c =
${subtract(a, b, c)}")
    println("Multiplication of $a * $b * $c =
${multiply(a, b, c)}")
    println("Division of $a / $b = ${divide(a, b)}")

}
```

Output:



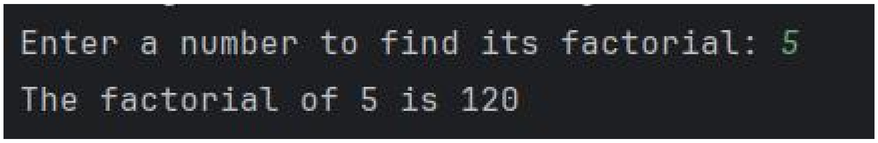
```
Addition of 763 + 118 + -649 = 232
Subtraction of 763 - 118 - -649 = 1294
Multiplication of 763 * 118 * -649 = -58432066
Division of 763 / 118 = 6
```

1.7.Factorial Calculation with Recursion: Calculate the factorial of a number using recursion.

Answer:

```
fun factorial( n : Int ) : Int{
    if (n == 0 || n == 1) {
        return 1
    } else {
        return n * factorial(n - 1)
    }
}

fun main(){
    print("Enter a number to find its factorial: ")
    val number = readln().toInt()
    val result = factorial(number)
    println("The factorial of $number is $result")
}
```

Output:



```
Enter a number to find its factorial: 5
The factorial of 5 is 120
```

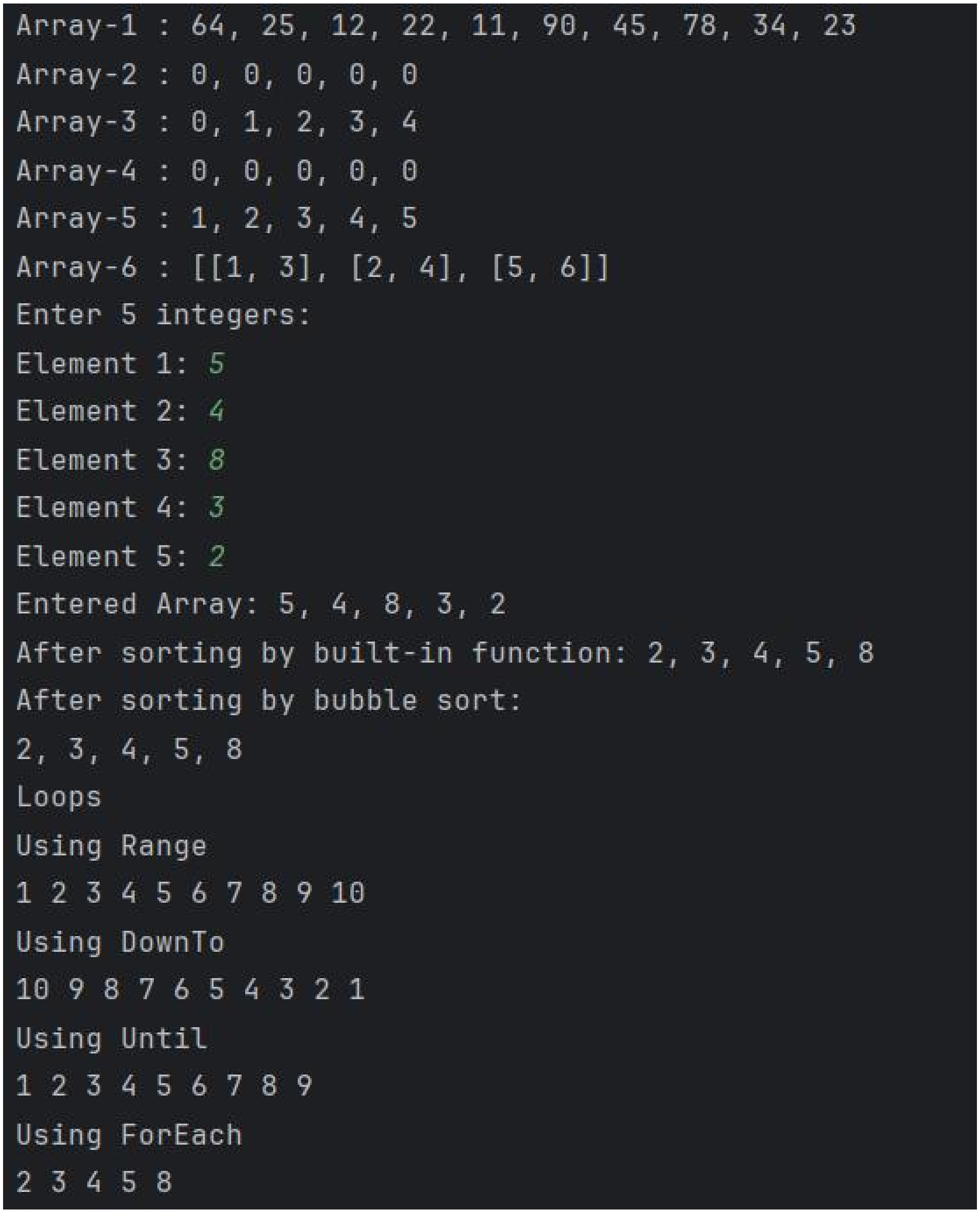
1.8.Working with Arrays: Explore array operations such as Arrays.deepToString(), contentDeepToString(), IntArray.joinToString(), and use them to print arrays. Utilize various

loop types like range, downTo, until, etc., to manipulate arrays. Sort an array of integers both without using built-in functions and with built-in functions.

Answer:

```
fun main() {
    val array1 = intArrayOf(64, 25, 12, 22, 11, 90,
45, 78, 34, 23)
    println("Array-1 : ${array1.joinToString()}")

    val array2 = Array(5) { 0 }
    println("Array-2 : ${array2.joinToString()}")

    val array3 = Array(5) { it }
    println("Array-3 : ${array3.joinToString()}")

    val array4 = IntArray(5)
    println("Array-4 : ${array4.joinToString()}")

    val array5 = intArrayOf(1, 2, 3, 4, 5)
    println("Array-5 : ${array5.joinToString()}")

    val array6 = arrayOf(
        intArrayOf(1,3),
        intArrayOf(2,4),
        intArrayOf(5,6),
    )
    println("Array-6 :
${array6.contentDeepToString()}")

    val inputArray = IntArray(5)
    println("Enter 5 integers:")
    for (i in inputArray.indices) {
        print("Element ${i + 1}: ")
        inputArray[i] = readLine()?.toIntOrNull() ?:
0
    }
    println("Entered Array:
${inputArray.joinToString()}")

    val builtInSorted = inputArray.sorted()
    println("After sorting by built-in function:
${builtInSorted.joinToString()}")

    println("After sorting by bubble sort:")
    for (i in inputArray.indices) {
        for (j in 0 until inputArray.size - i - 1) {
            if (inputArray[j] > inputArray[j + 1]) {
                val temp = inputArray[j]
                inputArray[j] = inputArray[j + 1]
                inputArray[j + 1] = temp
            }
        }
    }
    println(inputArray.joinToString())

    println("Loops")
    println("Using Range")
    for (i in 1..10) {
        print("$i ")
    }
    println()
    println("Using DownTo")
    for (i in 10 downTo 1) {
        print("$i ")
    }
    println()
    println("Using Until")
    for (i in 1 until 10) {
        print("$i ")
    }
    println()
    println("Using ForEach")
    inputArray.forEach { print("$it ") }
    println()
}
```

Output:



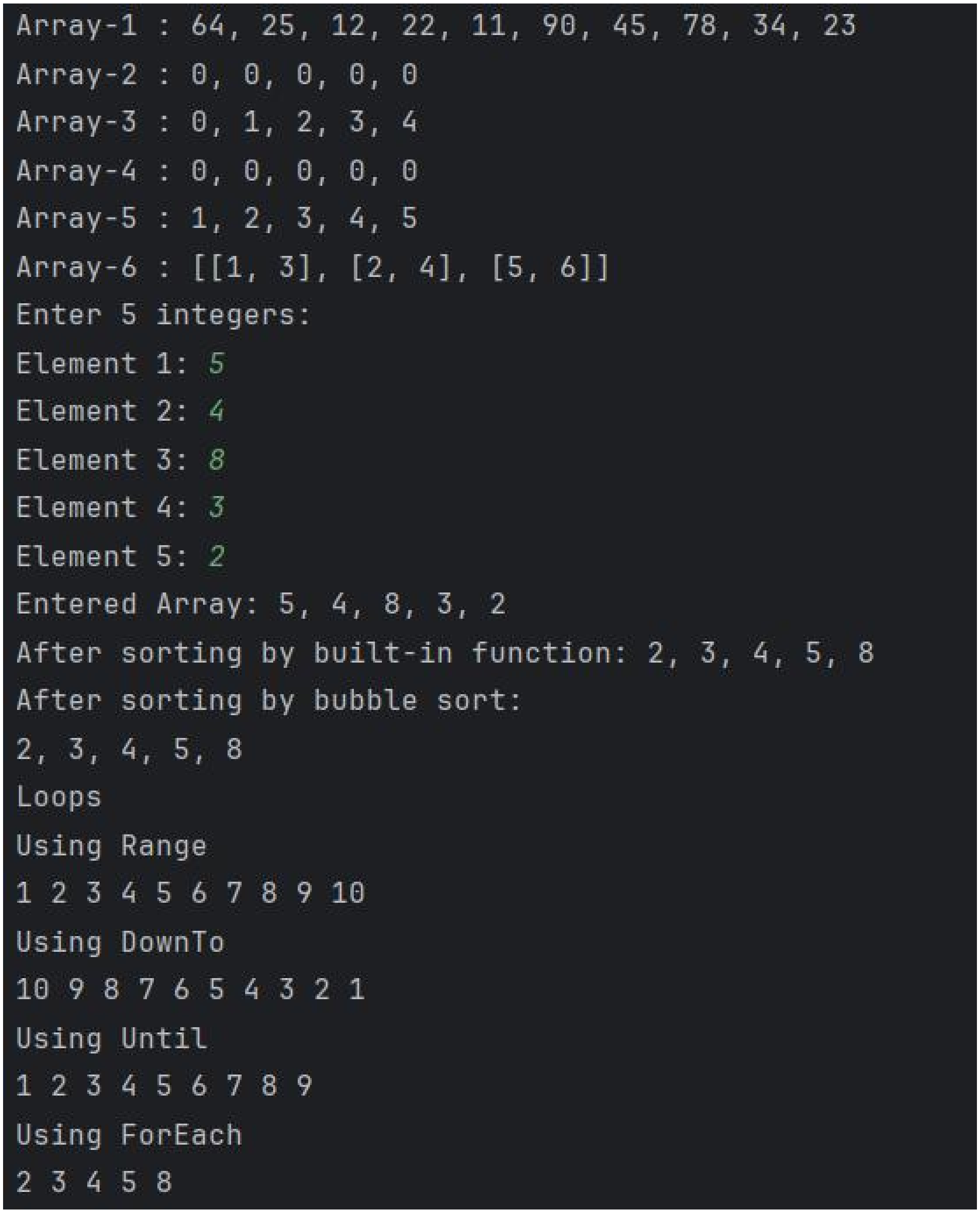
```
Array-1 : 64, 25, 12, 22, 11, 90, 45, 78, 34, 23
Array-2 : 0, 0, 0, 0, 0
Array-3 : 0, 1, 2, 3, 4
Array-4 : 0, 0, 0, 0, 0
Array-5 : 1, 2, 3, 4, 5
Array-6 : [[1, 3], [2, 4], [5, 6]]
Enter 5 integers:
Element 1: 5
Element 2: 4
Element 3: 8
Element 4: 3
Element 5: 2
Entered Array: 5, 4, 8, 3, 2
After sorting by built-in function: 2, 3, 4, 5, 8
After sorting by bubble sort:
2, 3, 4, 5, 8
Loops
Using Range
1 2 3 4 5 6 7 8 9 10
Using DownTo
10 9 8 7 6 5 4 3 2 1
Using Until
1 2 3 4 5 6 7 8 9
Using ForEach
2 3 4 5 8
```

1.9. Find Maximum Number from ArrayList: Write a program to find the maximum number from an ArrayList of integers.

Answer:

```
fun main(){
    val numbers = ArrayList<Int>()
    println("Enter the number of elements:")
    val n = readLine()?.toIntOrNull() ?: 0
    for (i in 1..n) {
        print("Enter element $i: ")
        val num = readLine()?.toIntOrNull() ?: 0
        numbers.add(num)
    }
    if (numbers.isNotEmpty()) {
        val maxNum = numbers.maxOrNull()
        println("Maximum number is: $maxNum")
    } else {
        println("No numbers entered.")
    }
}
```

Output:



```
Array-1 : 64, 25, 12, 22, 11, 90, 45, 78, 34, 23
Array-2 : 0, 0, 0, 0, 0
Array-3 : 0, 1, 2, 3, 4
Array-4 : 0, 0, 0, 0, 0
Array-5 : 1, 2, 3, 4, 5
Array-6 : [[1, 3], [2, 4], [5, 6]]
Enter 5 integers:
Element 1: 5
Element 2: 4
Element 3: 8
Element 4: 3
Element 5: 2
Entered Array: 5, 4, 8, 3, 2
After sorting by built-in function: 2, 3, 4, 5, 8
After sorting by bubble sort:
2, 3, 4, 5, 8
Loops
Using Range
1 2 3 4 5 6 7 8 9 10
Using DownTo
10 9 8 7 6 5 4 3 2 1
Using Until
1 2 3 4 5 6 7 8 9
Using ForEach
2 3 4 5 8
```

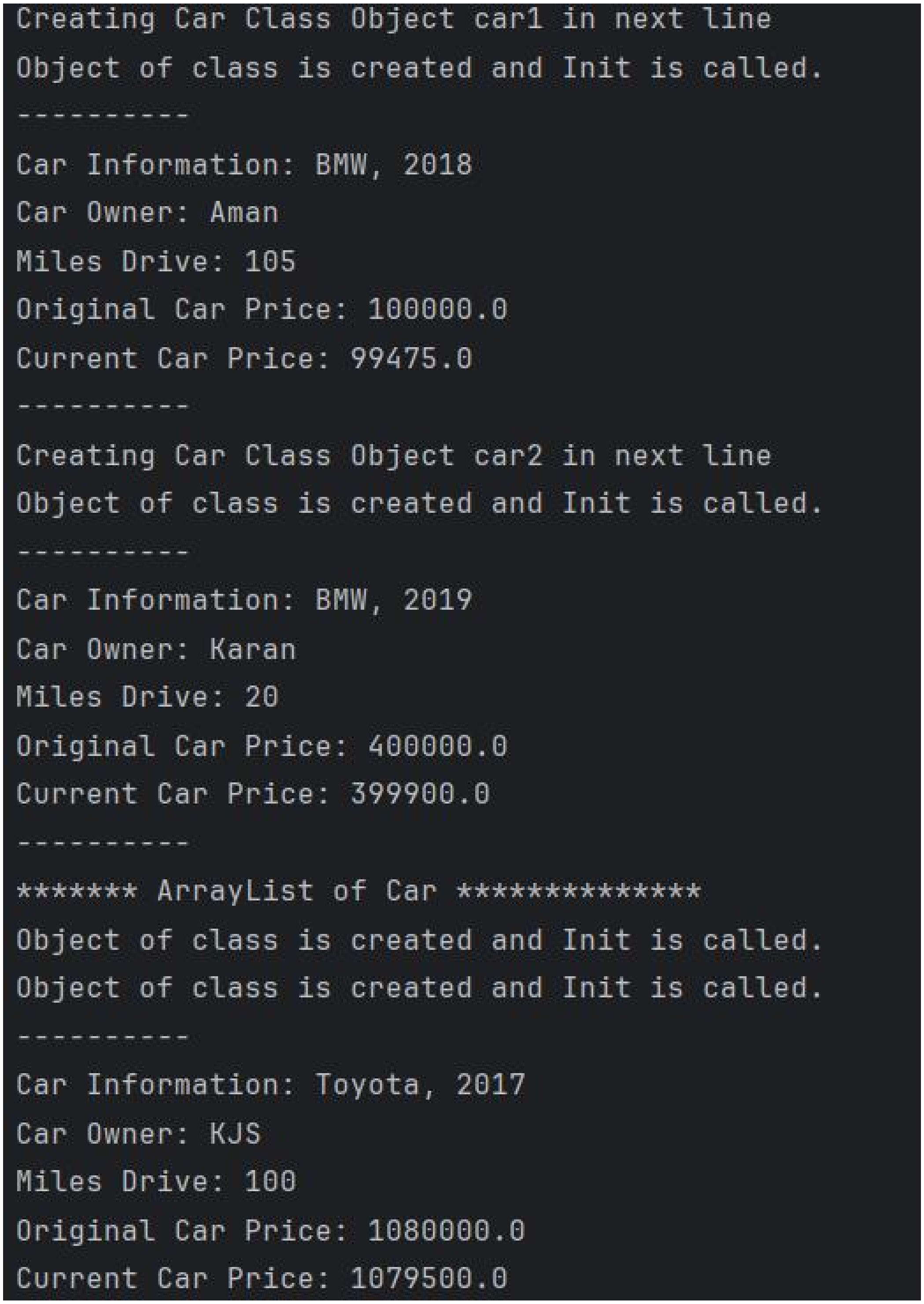
1.10. Class and Constructor Creation: Define different classes and constructors. Create a "Car" class with properties like type, model, price, owner, and miles driven. Implement functions to get car information, original car price, current car price, and display car information.

Answer:

```
class Car(
    var type: String,
    var model: Int,
    var price: Double,
    var owner: String,
    var milesDriven: Int
) {
    init {
        println("Object of class is created and Init
is called.")
    }

    fun getCarInfo(): String {
        return "Car Information: $type, $model\nCar
Owner: $owner\nMiles Drive: $milesDriven"
    }

    fun getOriginalCarPrice(): Double {
        return price
    }

    fun getCurrentPrice(): Double {
        return price - (milesDriven * 5)
    }

    fun displayCarInfo() {
        println("----------")
        println(getCarInfo())
        println("Original Car Price:
${getOriginalCarPrice()}")
        println("Current Car Price:
${getCurrentPrice()}")
        println("----------")
    }
```

```
}

fun main() {
    println("Creating Car Class Object car1 in next
line")
    val car1 = Car("BMW", 2018, 100000.0, "Aman",
105)
    car1.displayCarInfo()

    println("Creating Car Class Object car2 in next
line")
    val car2 = Car("BMW", 2019, 400000.0, "Karan",
20)
    car2.displayCarInfo()

    println("******* ArrayList of Car
**************")
    val carList = ArrayList<Car>()
    carList.add(Car("Toyota", 2017, 1080000.0, "KJS",
100))
    carList.add(Car("Maruti", 2020, 4000000.0, "NPP",
200))

    for (car in carList) {
        car.displayCarInfo()
    }
}
```

Output:



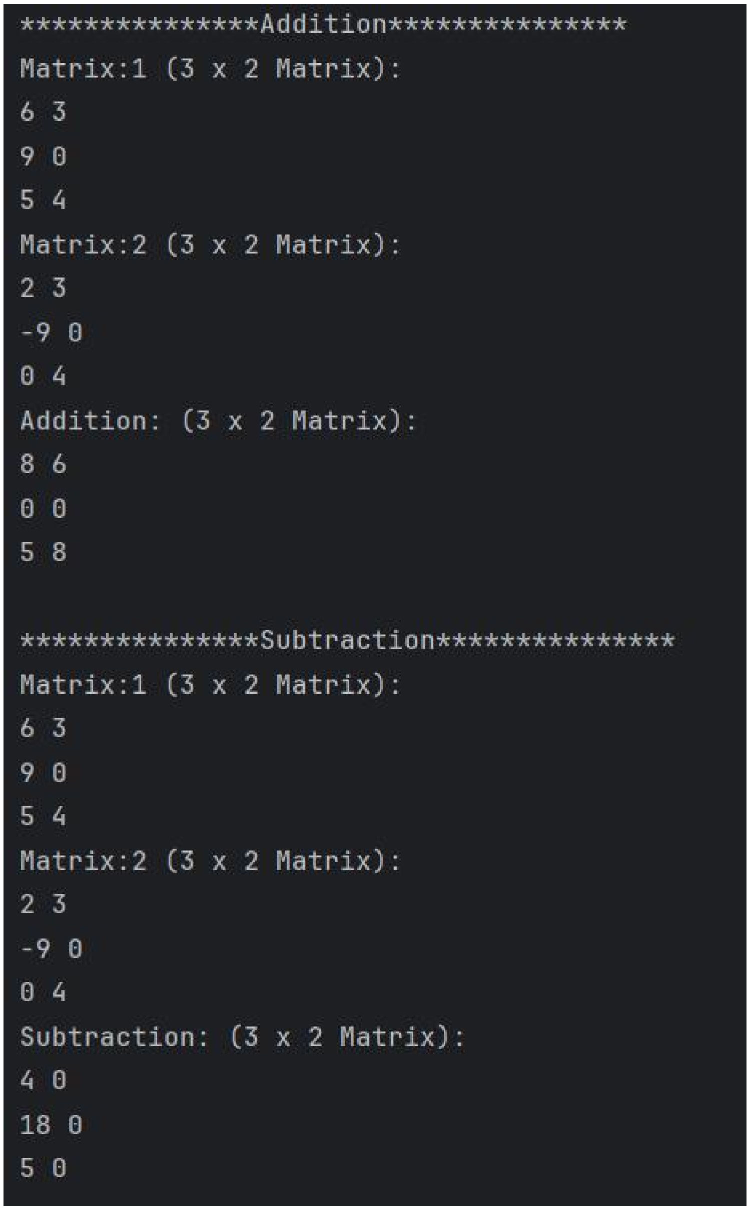
```
Creating Car Class Object car1 in next line
Object of class is created and Init is called.
----------
Car Information: BMW, 2018
Car Owner: Aman
Miles Drive: 105
Original Car Price: 100000.0
Current Car Price: 99475.0
----------
Creating Car Class Object car2 in next line
Object of class is created and Init is called.
----------
Car Information: BMW, 2019
Car Owner: Karan
Miles Drive: 20
Original Car Price: 400000.0
Current Car Price: 399900.0
----------
******* ArrayList of Car **************
Object of class is created and Init is called.
Object of class is created and Init is called.
----------
Car Information: Toyota, 2017
Car Owner: KJS
Miles Drive: 100
Original Car Price: 1080000.0
Current Car Price: 1079500.0
```

1.11. Operator Overloading and Matrix Operations: Explain operator overloading and implement matrix addition, subtraction, and multiplication using a "Matrix" class. Overload the toString() function in the "Matrix" class for customized output.

Answer:

```
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**************")
    print("Matrix:1 ")
    print(secondMatrix1)
    print("Matrix:2 ")
    print(secondMatrix)
    val thirdMatrix = secondMatrix1 + secondMatrix
    println("Addition: $thirdMatrix")


println("**************Subtraction**************")
    print("Matrix:1 ")
    print(secondMatrix1)
    print("Matrix:2 ")
    print(secondMatrix)
    val subtractMatrix = secondMatrix1 -
secondMatrix
    println("Subtraction: $subtractMatrix")


println("**************Multiplication*************
*")
    print("Matrix:1 ")
    print(firstMatrix)
    print("Matrix:2 ")
    print(secondMatrix)
    val multiplication = firstMatrix * secondMatrix
    println("Multiplication: $multiplication")
}
class Matrix(
    private val data: Array<IntArray>,
    private val noOfRow: Int,
    private val noOfCol: Int
) {

    operator fun plus(other: Matrix): Matrix {
        val result = Array(noOfRow)
{ IntArray(noOfCol) }
        for (i in 0 until noOfRow) {
            for (j in 0 until noOfCol) {
                result[i][j] = this.data[i][j] +
other.data[i][j]
            }
        }
        return Matrix(result, noOfRow, noOfCol)
    }

    operator fun minus(other: Matrix): Matrix {
        val result = Array(noOfRow)
{ IntArray(noOfCol) }
        for (i in 0 until noOfRow) {
            for (j in 0 until noOfCol) {
                result[i][j] = this.data[i][j] -
other.data[i][j]
            }
        }
        return Matrix(result, noOfRow, noOfCol)
    }
    operator fun times(other: Matrix): Matrix {
        val result = Array(noOfRow)
{ IntArray(other.noOfCol) }
        for (i in 0 until noOfRow) {
            for (j in 0 until other.noOfCol) {
                for (k in 0 until noOfCol) {
                    result[i][j] += this.data[i][k]
* other.data[k][j]
                }
            }
        }
        return Matrix(result, noOfRow, other.noOfCol)
    }

    override fun toString(): String {
        val builder = StringBuilder()
        builder.append("($noOfRow x $noOfCol
Matrix):\n")
        for (i in 0 until noOfRow) {
            for (j in 0 until noOfCol) {
                builder.append("${data[i][j]} ")
            }
            builder.append("\n")
        }
        return builder.toString()
    }
}
```

Output:



```
***************Addition***************
Matrix:1 (3 x 2 Matrix):
6 3
9 0
5 4
Matrix:2 (3 x 2 Matrix):
2 3
-9 0
0 4
Addition: (3 x 2 Matrix):
8 6
0 0
5 8

***************Subtraction***************
Matrix:1 (3 x 2 Matrix):
6 3
9 0
5 4
Matrix:2 (3 x 2 Matrix):
2 3
-9 0
0 4
Subtraction: (3 x 2 Matrix):
4 0
18 0
5 0
```