[2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]

Practical: 1

AIM-KOTLIN PROGRAMS

Submitted By:
21012011088

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

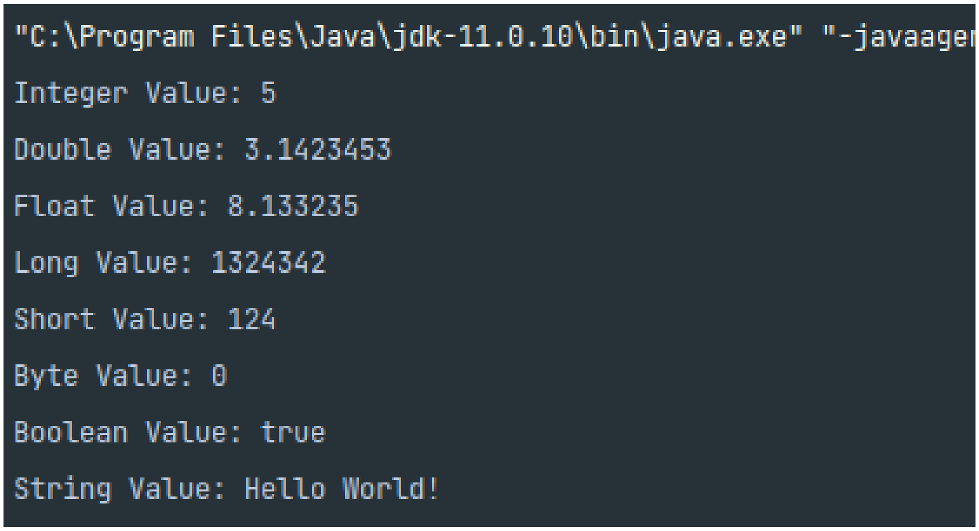
Department of Computer Engineering/Information Technology

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

Answer:

```
fun main(args: Array<String>) {
    val a: Int = 5
    val b: Double = 3.1423453
    val c: Float = 8.133235F
    val d: Long = 1324342
    val e: Short = 124
    val f: Byte = 0
    val g: Boolean = true
    val h: String = "Hello World!"

    println("Integer Value: $a")
    println("Double Value: $b")
    println("Float Value: $c")
    println("Long Value: $d")
    println("Short Value: $e")
    println("Byte Value: $f")
    println("Boolean Value: $g")
    println("String Value: $h")
}
```

Output:



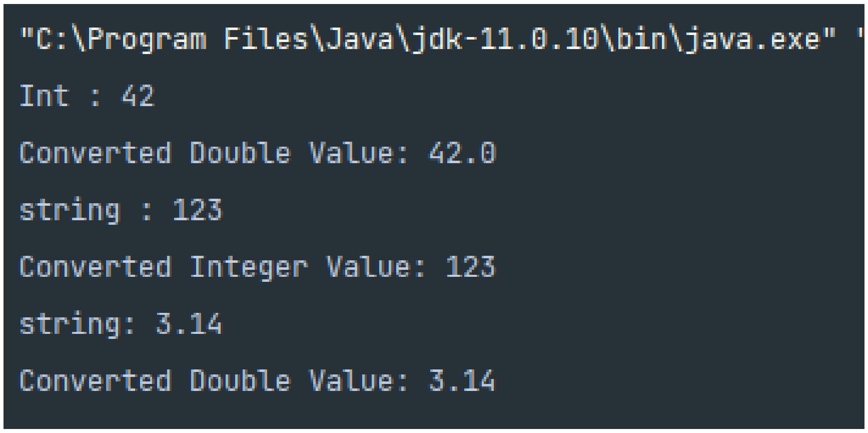
```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe" "-javaagen
Integer Value: 5
Double Value: 3.1423453
Float Value: 8.133235
Long Value: 1324342
Short Value: 124
Byte Value: 0
Boolean Value: true
String Value: Hello World!
```

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

Answer:

```
 fun main(args: Array<String>) {
    val i: Int = 42
    val id: Double = i.toDouble()

    val s: String = "123"
    val si: Int = s.toInt()

    val s2: String = "3.14"
    val s2d: Double = s2.toDouble()

    println("Int : $i")
    println("Converted Double Value: $id")

    println("string : $s")
    println("Converted Integer Value: $si")

    println("string: $s2")
    println("Converted Double Value: $s2d")
}
```

Output:



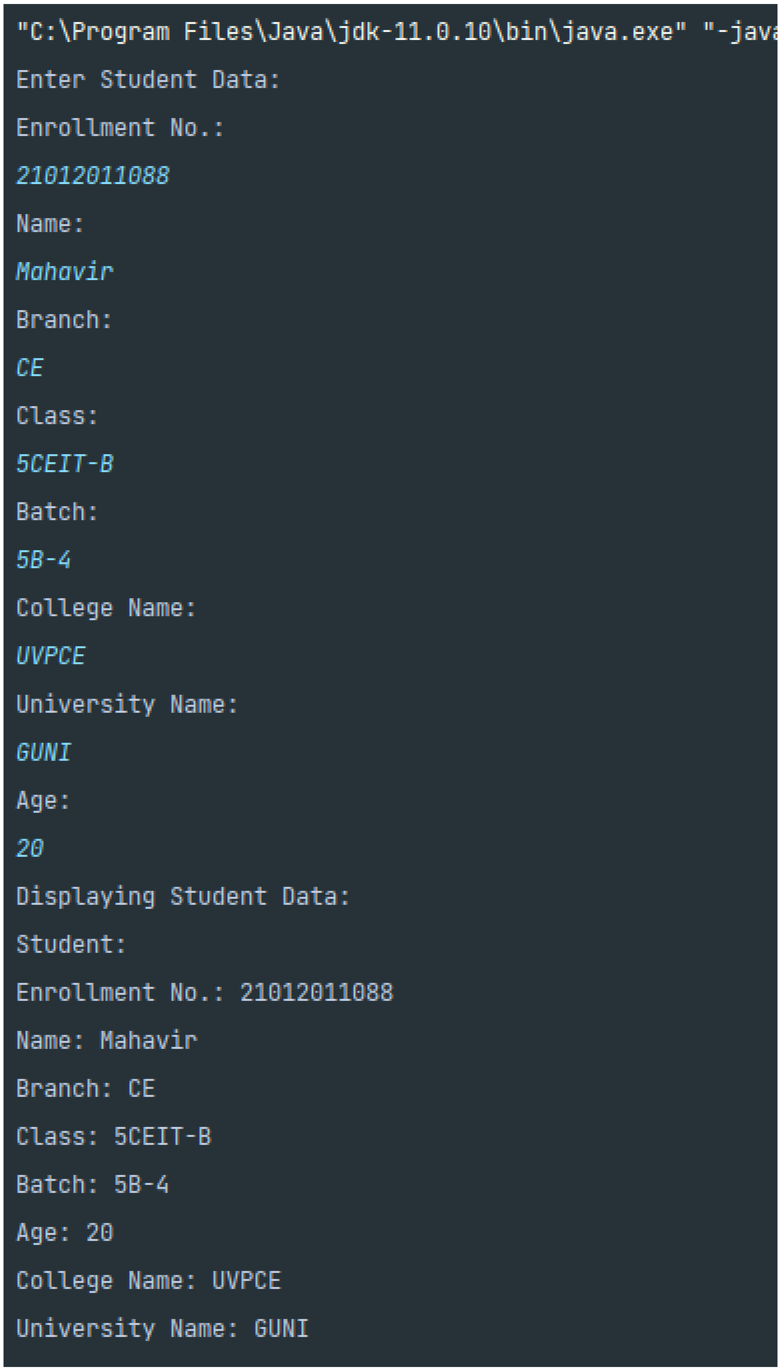
```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe" '
Int : 42
Converted Double Value: 42.0
string : 123
Converted Integer Value: 123
string: 3.14
Converted Double Value: 3.14
```

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

Answer:

```
 fun main() {
    println("Enter Student Data:")
    println("Enrollment No.:")
    val enrollmentNo = readln()

    println("Name:")
    val name = readln()

    println("Branch:")
    val branch = readln()

    println("Class:")
    val studentClass = readln()

    println("Batch:")
    val batch = readln()

    println("College Name:")
    val collegeName = readln()

    println("University Name:")
    val universityName = readln()

    println("Age:")
    val age = readln()

    println("Displaying Student Data:")
    println("Student:")
    println("Enrollment No.: $enrollmentNo")
    println("Name: $name")
    println("Branch: $branch")
    println("Class: $studentClass")
    println("Batch: $batch")
    println("Age: $age")
    println("College Name: $collegeName")
    println("University Name: $universityName")
}
```

Output:



```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe" "-java
Enter Student Data:
Enrollment No.:
21012011088
Name:
Mahavir
Branch:
CE
Class:
5CEIT-B
Batch:
5B-4
College Name:
UVPCE
University Name:
GUNI
Age:
20
Displaying Student Data:
Student:
Enrollment No.: 21012011088
Name: Mahavir
Branch: CE
Class: 5CEIT-B
Batch: 5B-4
Age: 20
College Name: UVPCE
University Name: GUNI
```

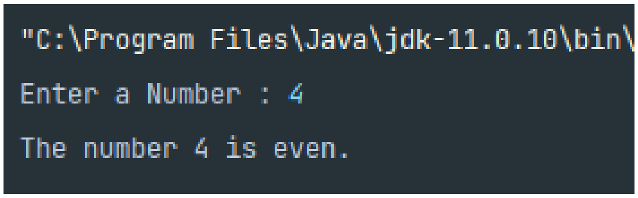
1.4.Find the number is odd or even by using Control Flow inside println() method.

Answer:

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

```
}
```

Output:



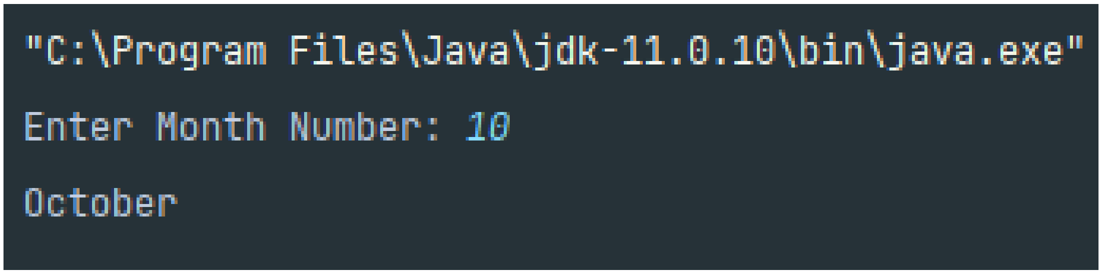
```
"C:\Program Files\Java\jdk-11.0.10\bin\
Enter a Number : 4
The number 4 is even.
```

1.5.Display month name using When.

Answer:

```
fun main() {
    print("Enter Month Number: ")
    val monthNumber = readln().toInt();

    when (monthNumber) {
        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("Please enter a proper number (1 to 12).")
    }
}
```

Output:



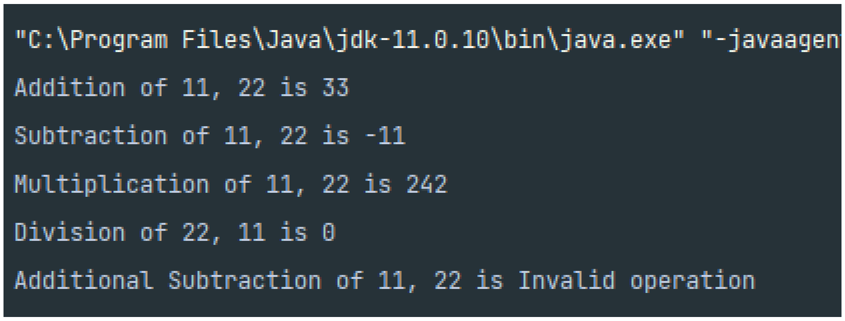
```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe"
Enter Month Number: 10
October
```

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

Answer:

```
 fun main() {
    val number1 = 11
    val number2 = 22

    performOperation("Addition", number1, number2)
    performOperation("Subtraction", number1, number2)
    performOperation("Multiplication", number1, number2)
    performOperation("Division", number2, number1)
    performOperation("Additional Subtraction", number1, number2)
}

fun performOperation(operation: String, num1: Int, num2: Int) {
    val result = when (operation) {
        "Addition" -> num1 + num2
        "Subtraction" -> num1 - num2
        "Multiplication" -> num1 * num2
        "Division" -> num2 / num1
        else -> null
    }

    println("$operation of $num1, $num2 is ${result ?: "Invalid
operation"}")
}
```

Output:



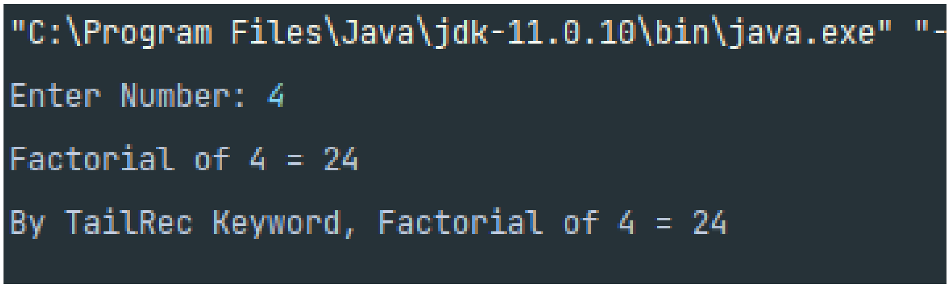
```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe" "-javaagen
Addition of 11, 22 is 33
Subtraction of 11, 22 is -11
Multiplication of 11, 22 is 242
Division of 22, 11 is 0
Additional Subtraction of 11, 22 is Invalid operation
```

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

Answer:

```
 fun main() {
    print("Enter Number: ")
    val number = readln().toInt();

    val factorial = calculateFactorial(number)
    println("Factorial of $number = $factorial")

    val factorialTailRec = calculateFactorialTailRec(number)
    println("By TailRec Keyword, Factorial of $number = 
$factorialTailRec")
}

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

tailrec fun calculateFactorialTailRec(n: Int, accumulator: Int = 
1): Int {
    return if (n == 0 || n == 1) {
        accumulator
    } else {
        calculateFactorialTailRec(n - 1, accumulator * n)
    }
}
```

Output:



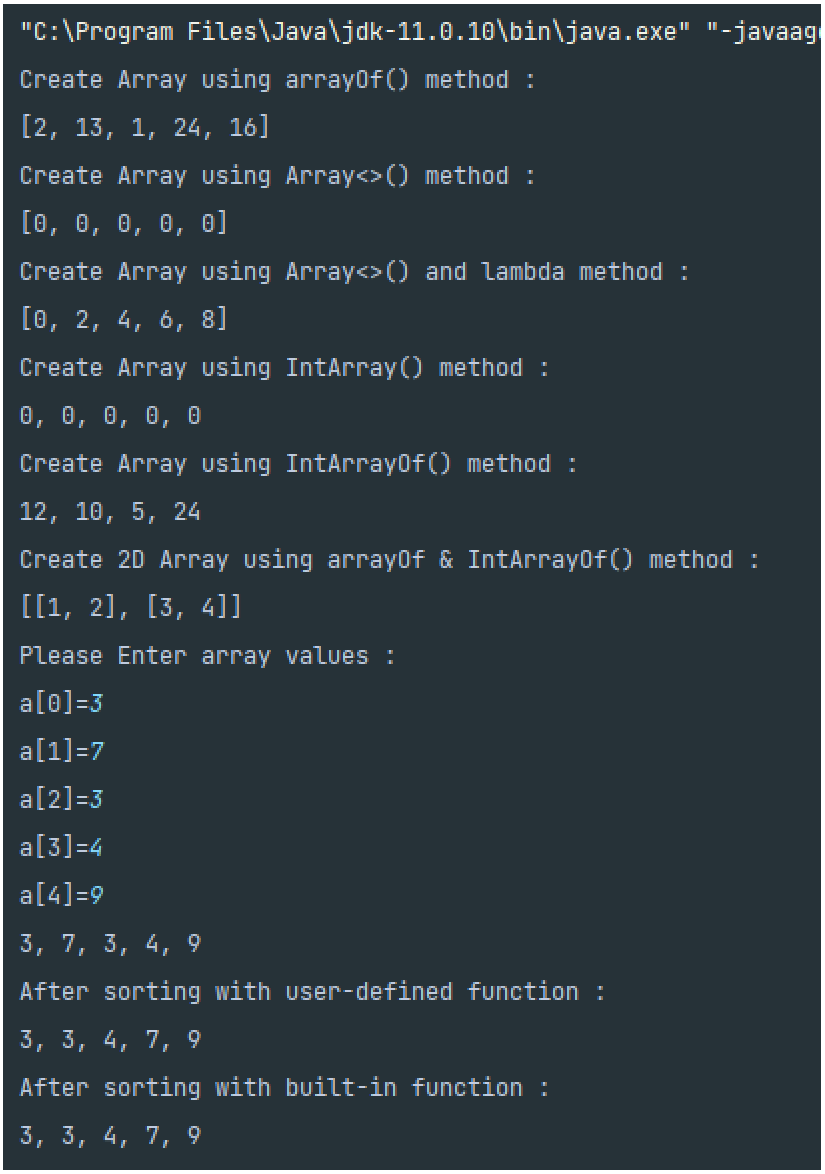
```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe" "-
Enter Number: 4
Factorial of 4 = 24
By TailRec Keyword, Factorial of 4 = 24
```

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:

```
 import java.util.Arrays
var lambdaFun = {size:Int -> Array<Int>(size){index -> index*2} }

fun sortArray(arrOrig : IntArray):IntArray{
    var arr = arrOrig;
    val len = arr.size;
    for(i in 0..<len){
        for(j in 0..<len){
            if (arr[j]>arr[i]){
                var temp = arr[i];
                arr[i] = arr[j];
                arr[j]= temp;
            }
        }
    }
    return arr;
}

fun main() {
    println("Create Array using arrayOf() method :")
    var arr1 = arrayOf(2, 13, 1, 24, 16)
    println(Arrays.deepToString(arr1))

    println("Create Array using Array<>() method :")
    var arr2 = Array<Int>(5){0}
    println(arr2.contentDeepToString())

    println("Create Array using Array<>() and lambda method :")
    var arr3 = lambdaFun(5)
    println(arr3.contentDeepToString())

    println("Create Array using IntArray() method :")
    var arr4 = IntArray(5){0}
    println(arr4.joinToString())
    println("Create Array using IntArrayOf() method :")
    var arr5 = intArrayOf(12,10,5,24)
    println(arr5.joinToString())

    println("Create 2D Array using arrayOf & IntArrayOf() method
:")
    var arr6 = arrayOf(intArrayOf(1,2) , intArrayOf(3,4))
    println(arr6.contentDeepToString())

    var arr7 = IntArray(5){0};
    println("Please Enter array values : ")
    for (i in 0..4){
        print("a[$i]=")
        arr7[i] = readln().toInt();
    }
    println(arr7.joinToString())

    println("After sorting with user-defined function : ")
    var sortArr = sortArray(arr7)
    println(sortArr.joinToString())

    println("After sorting with built-in function : ")
    arr7.sort()
    println(arr7.joinToString())
}
```

Output:



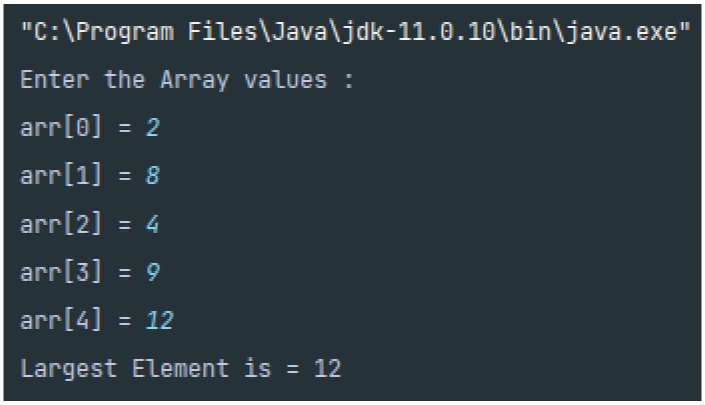
```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe" "-javaag
Create Array using arrayOf() method :
[2, 13, 1, 24, 16]
Create Array using Array<>() method :
[0, 0, 0, 0, 0]
Create Array using Array<>() and lambda method :
[0, 2, 4, 6, 8]
Create Array using IntArray() method :
0, 0, 0, 0, 0
Create Array using IntArrayOf() method :
12, 10, 5, 24
Create 2D Array using arrayOf & IntArrayOf() method :
[[1, 2], [3, 4]]
Please Enter array values :
a[0]=3
a[1]=7
a[2]=3
a[3]=4
a[4]=9
3, 7, 3, 4, 9
After sorting with user-defined function :
3, 3, 4, 7, 9
After sorting with built-in function :
3, 3, 4, 7, 9
```

1.9.Find the max number from ArrayList.

Answer:

```
 fun main(){
    println("Enter the Array values : ")
    var arr = Array<Int>(5){0};
    for(i in 0..<5){
        print("arr[$i] = ")
        arr[i] = readln().toInt();
    }
    arr.sort();
    println("Largest Element is = ${arr[arr.size-1]}")
}
```

Output:



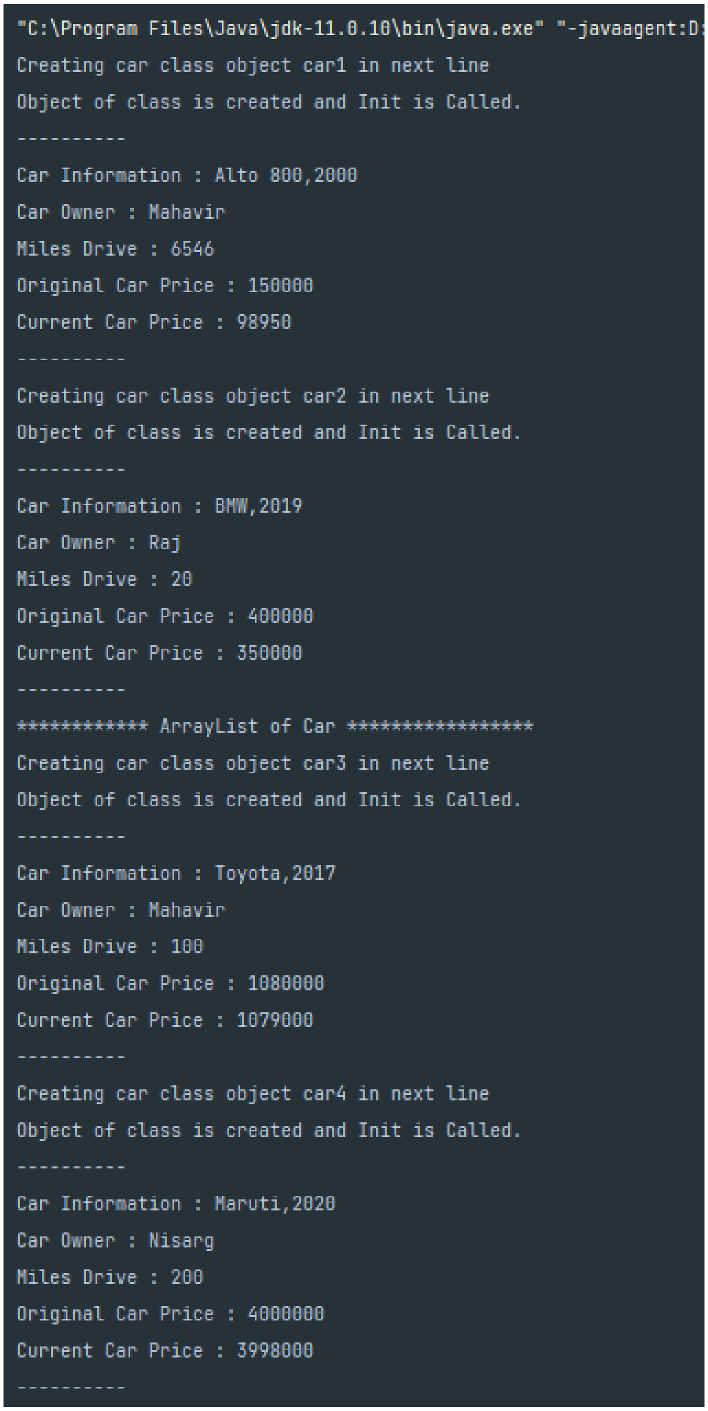
```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe"
Enter the Array values :
arr[0] = 2
arr[1] = 8
arr[2] = 4
arr[3] = 9
arr[4] = 12
Largest Element is = 12
```

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(
    var carobjn: Int,
    var type: String,
    var model: String,
    var originalprice: Long,
    var currentprice: Long,
    var owner: String,
    var miles: Int
) {
    fun getcarprice(): Long {
        return originalprice
    }

    fun getcarinformation(): Array<String> {
        var a1 = arrayOf<String>(type, model, owner)
        return a1
    }

    fun getoriginalprice(): Long {
        return originalprice
    }
    fun getcurrentprice(): Long {
        return currentprice
    }

    fun displycarinfo() {
        println("Creating car class object car$carobjn in next
line")
        println("Object of class is created and Init is Called.")
        println("----------")
        var a = getcarinformation()
        println("Car Information : ${a[0]},${a[1]}")
        println("Car Owner : ${a[2]}")
        println("Miles Drive : $miles")
        println("Original Car Price : $originalprice")
        println("Current Car Price : $currentprice")
        println("----------")
    }



}

fun main() {
    var no: Int = 0;
    no = no + 1
    var obj1 = car(no, "Alto 800", "2000", 150000, 98950,
"Mahavir", 6546)
    obj1.displycarinfo()
    no=no+1
    var obj2= car(no,"BMW","2019",400000,350000,"Raj",20)
    obj2.displycarinfo()
    println("************ ArrayList of Car ****************")
    no=no+1
    var
obj3=car(no,"Toyota","2017",1080000,1079000,"Mahavir",100)
    no=no+1
    var obj4=car(no,"Maruti","2020",4000000,3998000,"Nisarg",200)
    var person=arrayOf<car>(obj3,obj4)
    for(i in person)
    {
        i.displycarinfo()
    }
}
```

Output:



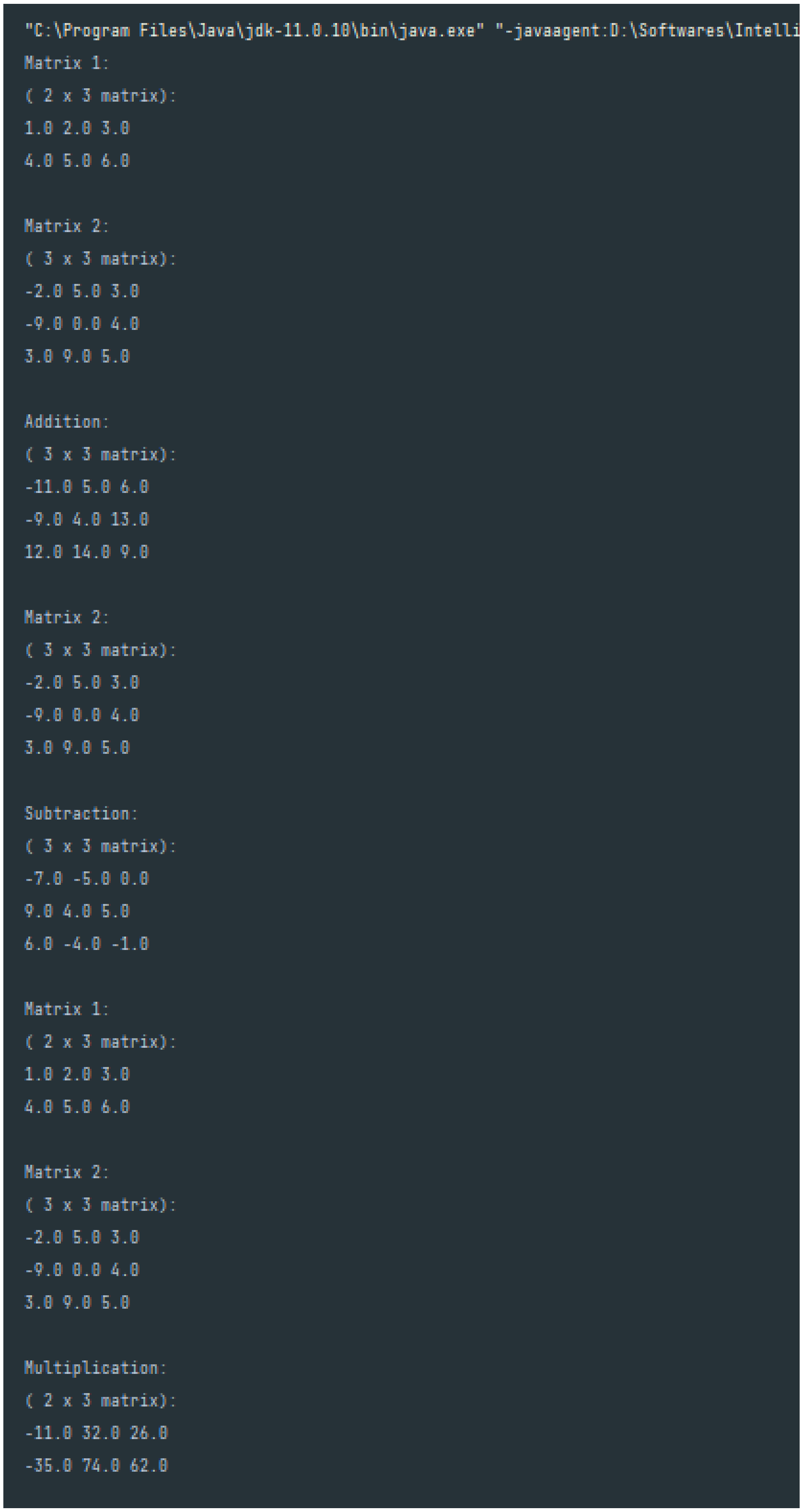
```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe" "-javaagent:D:
Creating car class object car1 in next line
Object of class is created and Init is Called.
----------
Car Information : Alto 800,2000
Car Owner : Mahavir
Miles Drive : 6546
Original Car Price : 150000
Current Car Price : 98950
----------
Creating car class object car2 in next line
Object of class is created and Init is Called.
----------
Car Information : BMW,2019
Car Owner : Raj
Miles Drive : 20
Original Car Price : 400000
Current Car Price : 350000
----------
************* ArrayList of Car *****************
Creating car class object car3 in next line
Object of class is created and Init is Called.
----------
Car Information : Toyota,2017
Car Owner : Mahavir
Miles Drive : 100
Original Car Price : 1080000
Current Car Price : 1079000
----------
Creating car class object car4 in next line
Object of class is created and Init is Called.
----------
Car Information : Maruti,2020
Car Owner : Nisarg
Miles Drive : 200
Original Car Price : 4000000
Current Car Price : 3998000
----------
```

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(private val data: Array<DoubleArray>) {
    val rows: Int = data.size
    val cols: Int = if (rows > 0) data[0].size else 0

    operator fun plus(other: Matrix): Matrix {
        if (rows != other.rows || cols != other.cols)
            throw IllegalArgumentException("Matrix dimensions do
not match for addition")

        val resultData = Array(rows) { DoubleArray(cols) }
        for (i in 0 until rows) {
            for (j in 0 until cols) {
                resultData[i][j] = this.data[i][j] +
other.data[i][j]
            }
        }
        return Matrix(resultData)
    }

    operator fun minus(other: Matrix): Matrix {
        if (rows != other.rows || cols != other.cols)
            throw IllegalArgumentException("Matrix dimensions do
not match for subtraction")

        val resultData = Array(rows) { DoubleArray(cols) }
        for (i in 0 until rows) {
            for (j in 0 until cols) {
                resultData[i][j] = this.data[i][j] -
other.data[i][j]
            }
        }
        return Matrix(resultData)
    }
    operator fun times(other: Matrix): Matrix {
        if (cols != other.rows)
            throw IllegalArgumentException("Matrix dimensions do
not match for multiplication")

        val resultData = Array(rows) { DoubleArray(other.cols) }
        for (i in 0 until rows) {
            for (j in 0 until other.cols) {
                for (k in 0 until cols) {
                    resultData[i][j] += this.data[i][k] *
other.data[k][j]
                }
            }
        }
        return Matrix(resultData)
    }

    override fun toString(): String {
        val sb = StringBuilder()
        sb.append("( $rows x $cols matrix):\n")
        for (i in 0 until rows) {
            for (j in 0 until cols) {
                sb.append(data[i][j])
                sb.append(" ")
            }
            sb.append("\n")
        }
        return sb.toString()
    }
}

fun main() {
    val firstMatrixData = arrayOf(doubleArrayOf(1.0, 2.0, 3.0),
doubleArrayOf(4.0, 5.0, 6.0))
    val secondMatrixData = arrayOf(doubleArrayOf(-2.0, 5.0, 3.0),
doubleArrayOf(-9.0, 0.0, 4.0), doubleArrayOf(3.0, 9.0, 5.0))
    val secondMatrix1Data = arrayOf(doubleArrayOf(-9.0, 0.0,
3.0), doubleArrayOf(0.0, 4.0, 9.0), doubleArrayOf(9.0, 5.0, 4.0))

    val firstMatrix = Matrix(firstMatrixData)
    val secondMatrix = Matrix(secondMatrixData)
    val secondMatrix1 = Matrix(secondMatrix1Data)

    println("Matrix 1:")
    println(firstMatrix)

    println("Matrix 2:")
    println(secondMatrix)

    val thirdMatrix = secondMatrix1 + secondMatrix
    println("Addition:")
    println(thirdMatrix)

    println("Matrix 2:")
    println(secondMatrix)

    val subtractMatrix = secondMatrix1 - secondMatrix
    println("Subtraction:")
    println(subtractMatrix)

    println("Matrix 1:")
    println(firstMatrix)

    println("Matrix 2:")
    println(secondMatrix)

    val multiplication = firstMatrix * secondMatrix
    println("Multiplication:")
    println(multiplication)
}
```

Output:



```
"C:\Program Files\Java\jdk-11.0.10\bin\java.exe" "-javaagent:D:\Softwares\Intelli
Matrix 1:
( 2 x 3 matrix):
1.0 2.0 3.0
4.0 5.0 6.0

Matrix 2:
( 3 x 3 matrix):
-2.0 5.0 3.0
-9.0 0.0 4.0
3.0 9.0 5.0

Addition:
( 3 x 3 matrix):
-11.0 5.0 6.0
-9.0 4.0 13.0
12.0 14.0 9.0

Matrix 2:
( 3 x 3 matrix):
-2.0 5.0 3.0
-9.0 0.0 4.0
3.0 9.0 5.0

Subtraction:
( 3 x 3 matrix):
-7.0 -5.0 0.0
9.0 4.0 5.0
6.0 -4.0 -1.0

Matrix 1:
( 2 x 3 matrix):
1.0 2.0 3.0
4.0 5.0 6.0

Matrix 2:
( 3 x 3 matrix):
-2.0 5.0 3.0
-9.0 0.0 4.0
3.0 9.0 5.0

Multiplication:
( 2 x 3 matrix):
-11.0 32.0 26.0
-35.0 74.0 62.0
```