

Practical: 1

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

Submitted By: Dharmendrasinh Rathod
Enrollment number: 22012011043



Department of Computer Engineering/Information Technology

Practical - 1

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.

Input:

```
fun main(){
    var a:Int
    a=24
    print("Integer value: $a")
    var b:Char='P'
    print("\nCharacter value: $b")
    var c:Float=24.7f
    print("\nFloat value: $c")
    var d:Double=24.7219
    print("\nDouble value: $d")
    var e:String="MAD"
    print("\nString value: $e")
    var f:Boolean=true
    print("\nBoolean value: $f")
    var g:Short=-2
    print("\nShort value: $g")
    var h:Long=240721092212
    print("\nLong value: $h")
    var i:Byte=-124
    print("\nByte value: $i")
}
```

Output:

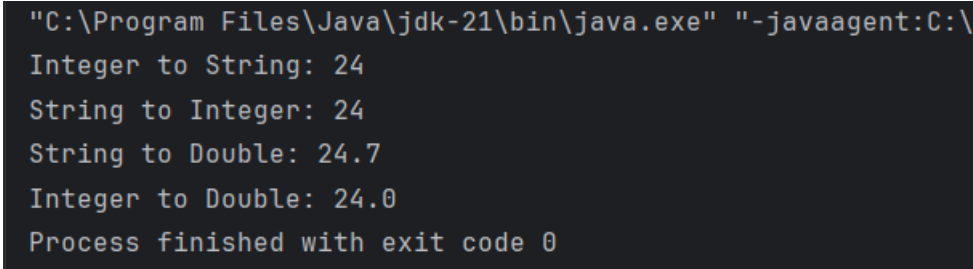
```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBra
Integer value: 24
Character value: P
Float value: 24.7
Double value: 24.7219
String value: MAD
Boolean value: true
Short value: -2
Long value: 240721092212
Byte value: -124
Process finished with exit code 0
```

1.2 Type Conversion: Perform type conversions such as Integer to Double, String to Integer, and String to Double.

Input :

```
fun main(){
    var a:Int=24
    var b:String="24"
    var c:String="24.7"
    print("Integer to String: "+a.toString())
    print("\nString to Integer: "+b.toInt())
    print("\nString to Double: "+c.toDouble())
    print("\nInteger to Double: "+a.toDouble())
}
```

Output:



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\...
Integer to String: 24
String to Integer: 24
String to Double: 24.7
Integer to Double: 24.0
Process finished with exit code 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.

Input :

```
fun main(){
    print("Student Enrollment no.: ")
    val en:String?=readLine()
    print("Student name: ")
    val name:String?=readLine()
    print("Student branch: ")
    val branch:String?=readLine()
    print("Student class: ")
    val cl:String?=readLine()
    print("Student batch: ")
    val batch:String?=readLine()
    print("Student college name: ")
    val cn:String?=readLine()
    print("Student university name: ")
    val un:String?=readLine()
    print("Student age: ")
    val age:String?=readLine()
    print("\n*****\n")
    print("\nStudent data: \n")
}
```

```
print("Enrollment no: $en\n")
print("Name: $name\n")
print("Branch: $branch\n")
print("Class: $cl\n")
print("Batch: $batch\n")
print("College name: $cn\n")
print("University name: $un\n")
}
```

Output:

```
Student Enrollment no.: 22012011042
Student name: RPM
Student branch: CE
Student class: SCE-B-4
Student batch: B-4
Student college name: U V PATEL COLLEGE OF ENGINEERING
Student university name: GANPAT UNIVERSITY
Student age: 20

*****

Student data:
Enrollment no: 22012011042
Name: RPM
Branch: CE
Class: SCE-B-4
Batch: B-4
College name: U V PATEL COLLEGE OF ENGINEERING
University name: GANPAT UNIVERSITY
```

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

Input:

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

```
}
```

Output:

```
Enter any number: 724
724 is an even number.
```

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

Input:

```
fun main() {
    print("Enter a month number (1-12): ")
    val num = readln().toInt()
    val name = when (num) {
        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 number"
    }
    println("Month: $name")
}
```

Output:

```
Enter a month number (1-12): 7
Month: July

Process finished with exit code 0
```

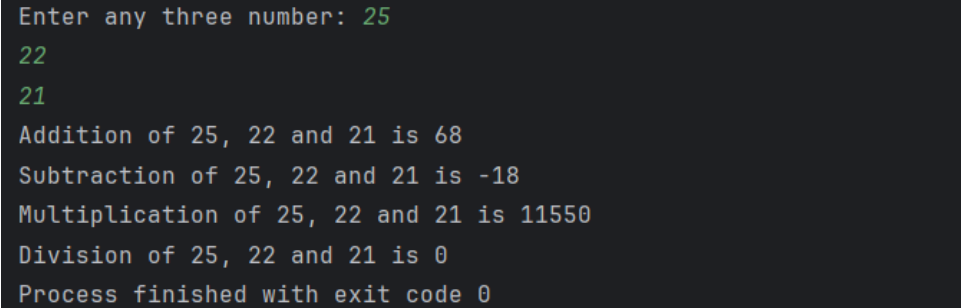
1.6 User-Defined Function: Create a user-defined function to perform arithmetic operations (addition, subtraction, multiplication, division) on two numbers.

Input:

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

```
    return a*b*c
}
fun div(a:Int, b:Int, c:Int):Int{
    return a/b/c
}
fun main(){
    print("Enter any three number: ")
    var a=readln().toInt()
    var b=readln().toInt()
    var c=readln().toInt()
    print("Addition of $a, $b and $c is ${add(a,b,c)}")
    print("\nSubtraction of $a, $b and $c is "+sub(a, b, c))
    print("\nMultiplication of $a, $b and $c is "+mul(a, b, c))
    print("\nDivision of $a, $b and $c is "+div(a, b, c))
}
```

Output:

A screenshot of a terminal window showing the output of the program. The user enters three numbers: 25, 22, and 21. The program then displays the results of addition, subtraction, multiplication, and division for these numbers. The output is as follows:

```
Enter any three number: 25
22
21
Addition of 25, 22 and 21 is 68
Subtraction of 25, 22 and 21 is -18
Multiplication of 25, 22 and 21 is 11550
Division of 25, 22 and 21 is 0
Process finished with exit code 0
```

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

Input:

```
fun main() {
    print("Enter a number: ")
    val num = readln().toInt()
    if (num != null && num >= 0) {
        val result = factorial(num)
        println("The factorial of $num is $result")
    } else {
        println("Please enter a valid non-negative integer.")
    }
}

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

Output:

```
Enter a number: 7
The factorial of 7 is 5040

Process finished with exit code 0
```

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.

Input:

```
fun main(){
    val array1 = arrayOf(1,2,3)
    println("Using arrayOf() method : ${array1.contentToString()}")
    val array2 = arrayOf<Int>(3,4,5)
    println("Using arrayOf<>() method : ${array2.contentToString()}")
    val array3 = Array<Int>(7) { i -> i * 1 }
    println("Using Array<>(){} method : ")
    for (i in 0..array3.size-1)
    {
        print("${array3[i]} ")
    }
    println()
    println("Using IntArray(){} method : ")
    val array4 = IntArray(3) { i -> i * 2 }
    for (i in 0..array4.size-1)
    {
        print("${array4[i]} ")
    }
    println()
    val array5 = intArrayOf(1,2,3)
    println("Using IntArray() method : ${array5.joinToString()}")
    val array6 = arrayOf(
        arrayOf(1,2,3),
        arrayOf(3,4,5),
        arrayOf(5,6,7)
    )
    println("2D Array : ${array6.contentDeepToString()}")
    val array7 = IntArray(5)
    for (i in array7.indices){
        print("Enter value of array7[$i] : ")
        array7[i] = readln().toInt()
    }
}
```

```
println("Your array : ${array7.joinToString()}")
array7.sort()
println("Using builtIn fun. sort : ${array7.joinToString()}")
for (i in array7.indices){
    for (j in array7.indices){
        if (array7[i]<array7[j]){
            val temp = array7[i]
            array7[i] = array7[j]
            array7[j] = temp
        }
    }
}
println("Without Using builtIn fun. sort : ${array7.joinToString()}")
}
```

Output:

```
Using arrayOf() method : [1, 2, 3]
Using arrayOf<>() method : [3, 4, 5]
Using Array<>(){} method :
0 1 2 3 4 5 6
Using IntArray(){} method :
0 2 4
Using IntArray() method : 1, 2, 3
2D Array : [[1, 2, 3], [3, 4, 5], [5, 6, 7]]
Enter value of array7[0] : 4
Enter value of array7[1] : 2
Enter value of array7[2] : 1
Enter value of array7[3] : 5
Enter value of array7[4] : 3
Your array : 4, 2, 1, 5, 3
Using builtIn fun. sort : 1, 2, 3, 4, 5
Without Using builtIn fun. sort : 1, 2, 3, 4, 5
```

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

Input:

```
fun main(){
    print("Enter size of array: ")
    val a= readln().toInt()
    var x=Array(a){0}
    print("Enter $a elements of array: ")
    for(i in 0..a-1){
        x[i]= readln().toInt()
    }
}
```



```
}
var g=0
for(i in 0..a-1){
    if(g<x[i]){
        g=x[i]
    }
}
print("\nGreatest number in array is $g...")
}
```

Output:

```
Enter size of array: 5
Enter 5 elements of array: 25
22
92
21
24

Greatest number in array is 92...
Process finished with exit code 0
```

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.

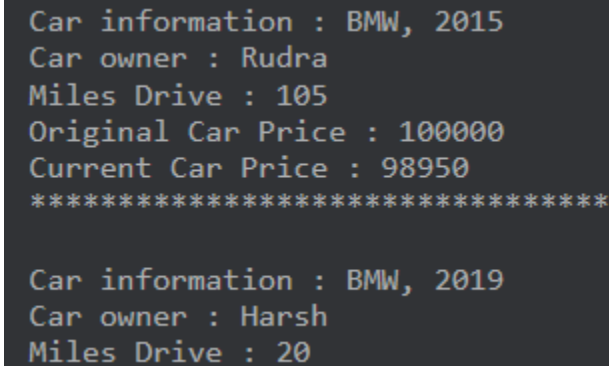
Input:

```
class Car (val info : String, val owner : String, val miles : Int, val oPrice : Int, val cPrice : Int) {
    fun information() {
        println("Car information : $info")
        println("Car owner : $owner")
        println("Miles Drive : $miles")
        println("Original Car Price : $oPrice")
        println("Current Car Price : $cPrice")
        println("*****")
        println()
    }
}

fun main(){
    val c1 = Car("BMW, 2015", "Rudra", 105, 100000, 98950)
    val c2 = Car("BMW, 2019", "Harsh", 20, 400000, 399800)
    val c3 = Car("Toyota", "Raj", 100, 10880000, 1079000)
```

```
val c4 = Car("Maruti","Rami",200,400000,399800)
c1.information()
c2.information()
c3.information()
c4.information()
}
```

Output:



```
Car information : BMW, 2015
Car owner : Rudra
Miles Drive : 105
Original Car Price : 100000
Current Car Price : 98950
*****

Car information : BMW, 2019
Car owner : Harsh
Miles Drive : 20
```

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.

Input:

```
class Matrix(private val matrix: Array<IntArray>, private val rows: Int, private val cols: Int) {
    operator fun plus(other: Matrix): Matrix {
        val result = Array(rows) { IntArray(cols) }
        for (i in 0 until rows) {
            for (j in 0 until cols) {
                result[i][j] = this.matrix[i][j] + other.matrix[i][j]
            }
        }
        return Matrix(result, rows, cols)
    }
    operator fun minus(other: Matrix): Matrix {
        val result = Array(rows) { IntArray(cols) }
        for (i in 0 until rows) {
            for (j in 0 until cols) {
                result[i][j] = this.matrix[i][j] - other.matrix[i][j]
            }
        }
    }
}
```

```
        return Matrix(result, rows, cols)

    }

    operator fun times(other: Matrix): Matrix {
        val result = Array(this.rows) { IntArray(other.cols) }
        for (i in 0 until this.rows) {
            for (j in 0 until other.cols) {
                for (k in 0 until this.cols) {
                    result[i][j] += this.matrix[i][k] * other.matrix[k][j]
                }
            }
        }
        return Matrix(result, this.rows, other.cols)
    }

    override fun toString(): String {
        return matrix.joinToString(separator = "\n") { it.joinToString(separator = " ") }
    }
}

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:\n$secondMatrix1\n")
    print("Matrix 2:\n$secondMatrix\n")
    val thirdMatrix = secondMatrix1 + secondMatrix
    println("Addition:\n$thirdMatrix")

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

    println("*****Multiplication*****")
    print("Matrix 1:\n$firstMatrix\n")
    print("Matrix 2:\n$secondMatrix\n")
    val multiplication = firstMatrix * secondMatrix
    println("Multiplication:\n$multiplication")
}
```

Output:

```
*****Addition*****
Matrix 1:
6 3
9 0
5 4
Matrix 2:
2 3
-9 0
0 4
Addition:
8 6
0 0
5 8
*****Subtraction*****
Matrix 1:
6 3
9 0
5 4
Matrix 2:
2 3
-9 0
0 4
Subtraction:
4 0
18 0
5 0
*****Multiplication*****
Matrix 1:
3 -2 5
3 0 4
Matrix 2:
2 3
-9 0
0 4
Multiplication:
24 29
6 25

Process finished with exit code 0
```