**[ 2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]**

Practical:1  
  
  
  
  
 **AIM:KOTLIN PROGRAMS**

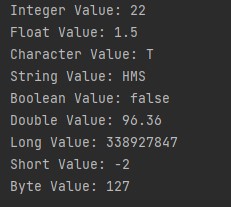
Enrollment number: 20012011150



**Department of Computer Engineering/**

**Information Technology**

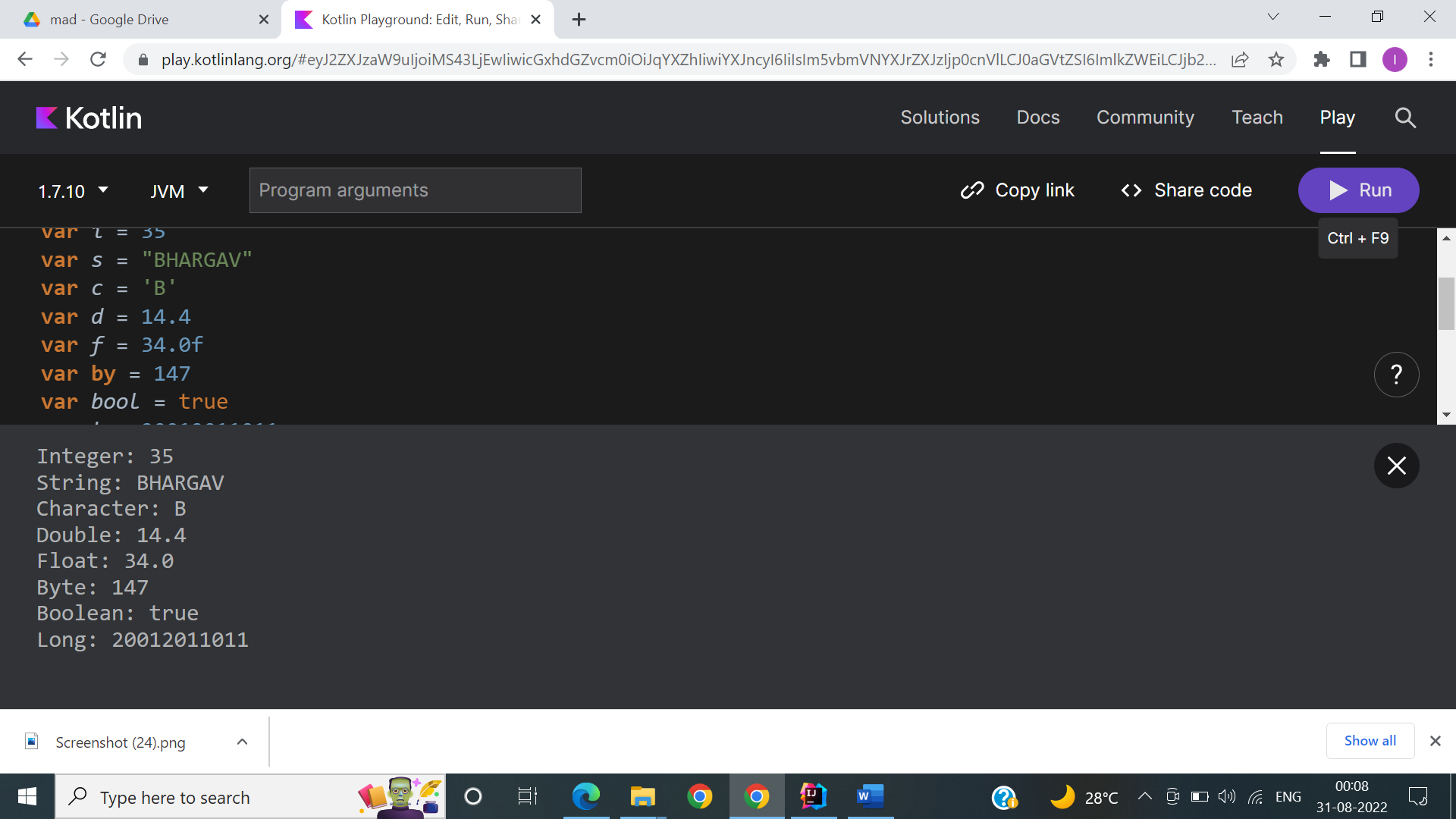
1.Store & Display values in different variable of different type (Integer, Double, Float, Long, Short, Byte, Char, Boolean, String)



Code:

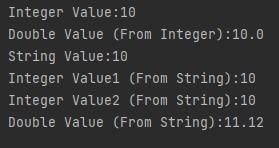
|  |
| --- |
| fun main()  {  println("20012011035\_ISHWAR DESAI")  var i = 35  var s = "Ishwar"  var c = 'I'  var d = 12.4  var f = 32.0f  var by = 137  var bool = true  var l = 20012011035  println("Integer: $i")  println("String: $s")  println("Character: $c")  println("Double: $d")  println("Float: $f")  println("Byte: $by")  println("Boolean: $bool")  println("Long: $l")  } |

Output:



2.Type conversion:

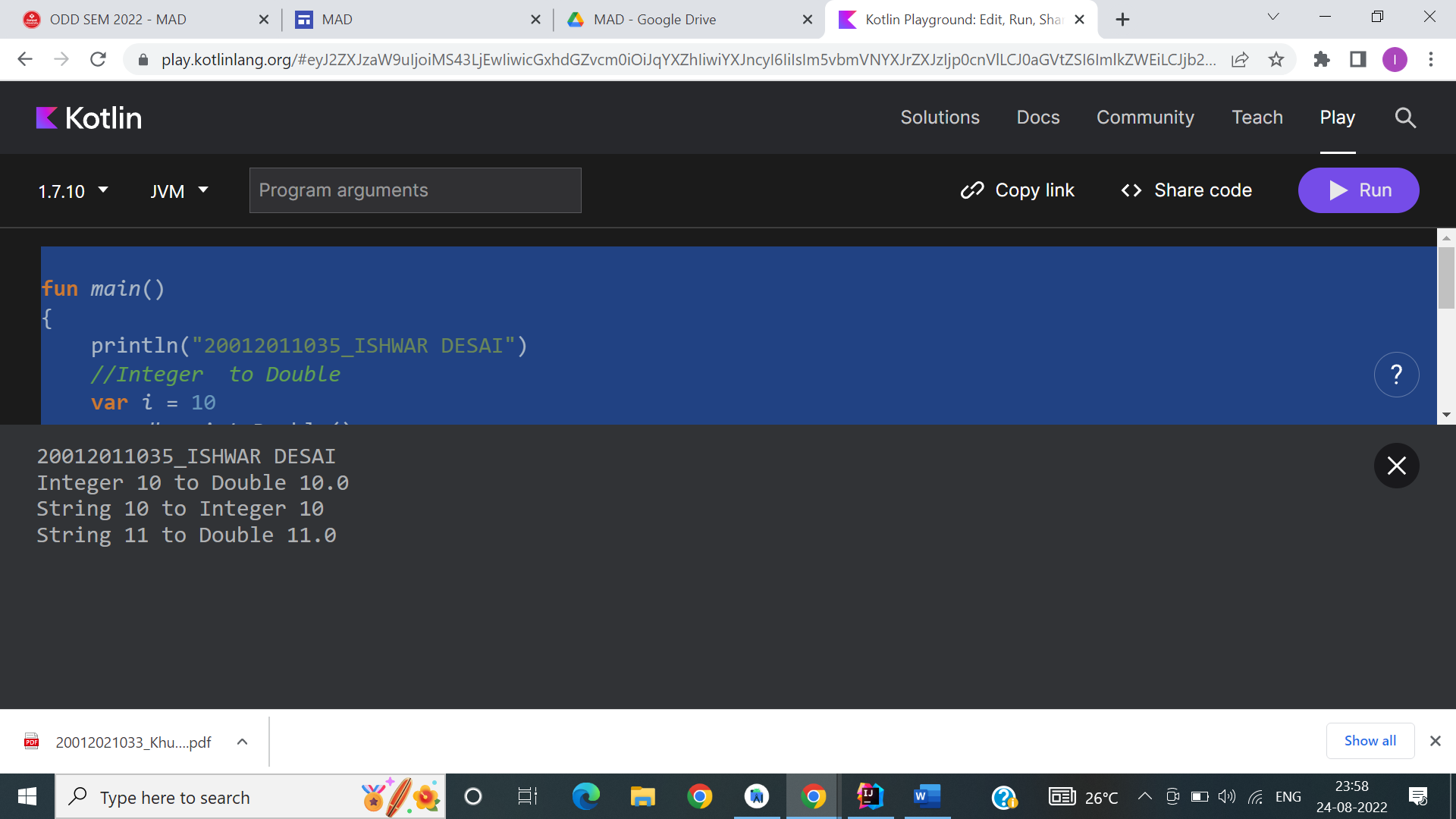
Integer to Double, String to Integer, String to Double.



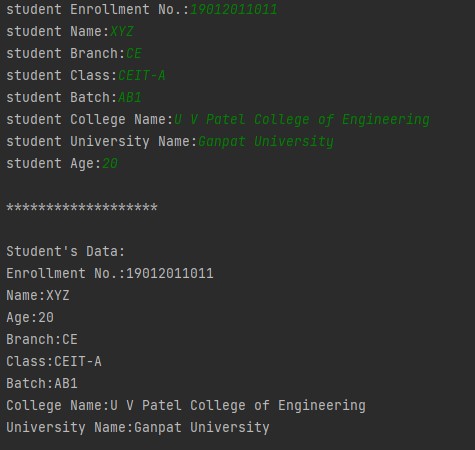
Code:

|  |
| --- |
| fun main()  {    //Integer to Double  var i = 10  var db = i.toDouble()  println("Integer $i to Double $db")  //String to Integer  var s = "10"  var i2 = s.toInt()  println("String $s to Integer $i2")    //String to Double  var s2 = "11"  var d = s2.toDouble()  println("String $s2 to Double $d") } |

Output:



3.Scan student’s information and display all the data.



Code:

fun main()

{

    println("Enter Student's Enroll:")

    var enroll= readLine()

    println("Enter Student's Name:")

    var name= readLine()

    println("Enter Student's Branch:")

    var branch= readLine()

    println("Enter Student's Class:")

    var c= readLine()

    println("Enter Batch:")

    var  batch= readLine()

    println("Enter Student's Collage Name:")

    var  n = readLine()

    println("Enter Student's University Name:")

    var u= readLine()

    println("Enter Student's Age:")

    var age= readLine()

    println(" Student's Enroll:$enroll")

    println(" Student's Name:$name")

    println("Enter Student's Branch:$branch")

    println("Enter Batch:$batch")

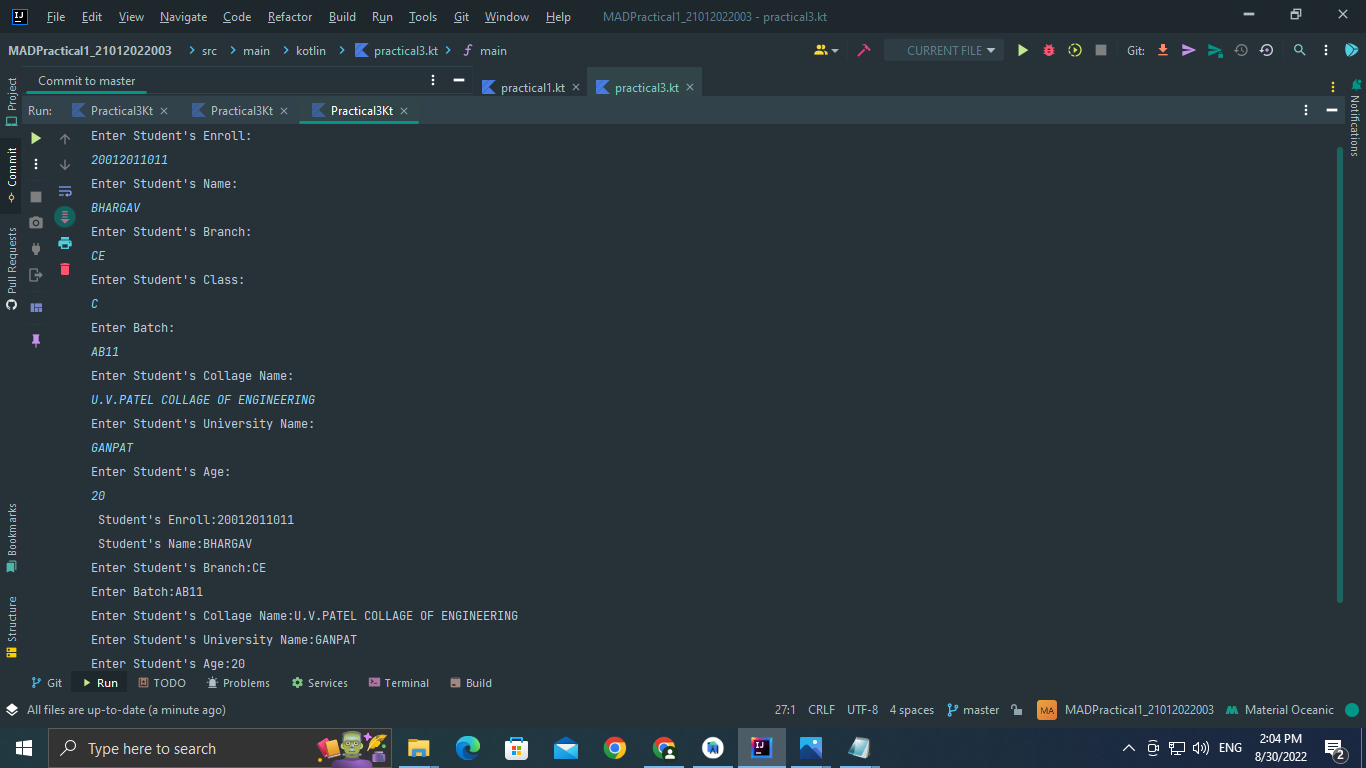
    println("Enter Student's Collage Name:$n")

    println("Enter Student's University Name:$u")

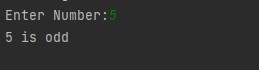
    println("Enter Student's Age:$age")

}

Output:



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

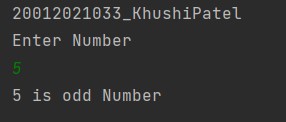


Code:

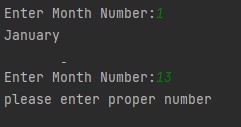
fun main() {

|  |
| --- |
| *println*("Enter Number ") var i = *readLine*()!!.*toInt*() if( i % 2 ==0 )  *println*("$i is even Number") else  *println*("$i is odd Number")    } |

Output:



5.Display month name using When

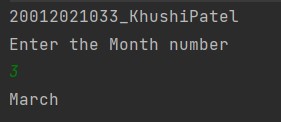


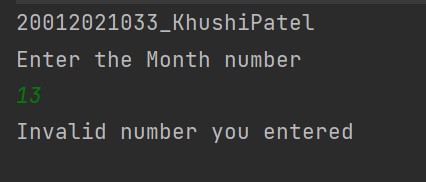
Code:

|  |
| --- |
| fun main()  {  *println*("Enter the Month number") var month = *readLine*()  when (month)  {  "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*("Invalid number you entered") |

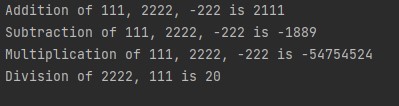
} }

Output:





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



Code:

|  |
| --- |
| import java.util.Scanner fun main(args: Array<String>)  {  val reader = Scanner(System.*`in`*) *println*("Enter the Number 1") val num1:Int = reader.nextInt() val read = Scanner(System.*`in`*) *println*("Enter the Number 2") val num2:Int = read.nextInt()    val add = *addition*(num1,num2) val sub = *subtraction*(num1,num2) val mul = *multiplication*(num1,num2) val div = *division*(num1,num2) *println*("Addition is $add") *println*("Subtraction is $sub") *println*("Multiplication is $mul") *println*("Division is $div")    }  fun addition(num1: Int,num2: Int): Int { |

return num1 + num2

}

fun subtraction(num1: Int,num2: Int): Int

{

return num1 - num2

}

fun multiplication(num1: Int,num2: Int): Int

{

return num1 \* num2

}

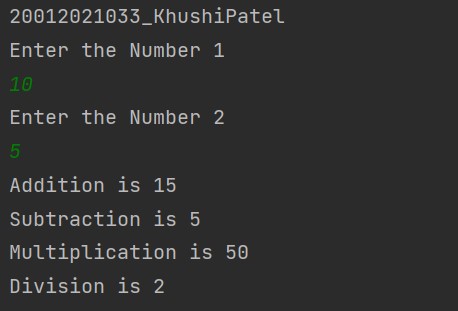
fun division(num1: Int,num2: Int): Int

{

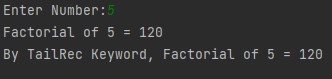
return num1 / num2

}

Output:



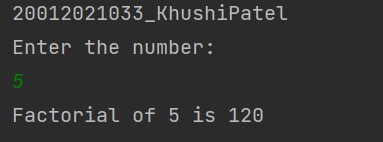
7.Find the factorial of number by recursion. Explain "tailrec" keyword.



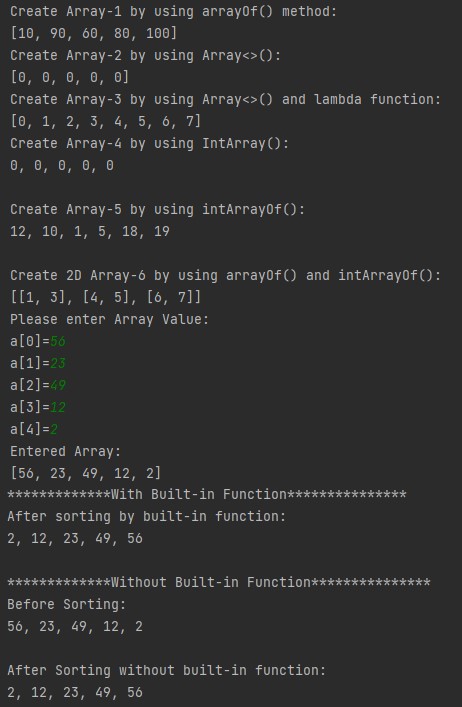
Code:

|  |
| --- |
| tailrec fun fact(n:Int):Int  {  if(n==1) {  return 1  }  return n\**fact*(n-1)  }  fun main()  {  *println*("Enter the number: ") var num:Int = *readLine*()!!.*toInt*()  *println*("Factorial of $num is ${*fact*(num)}") } |

Output:



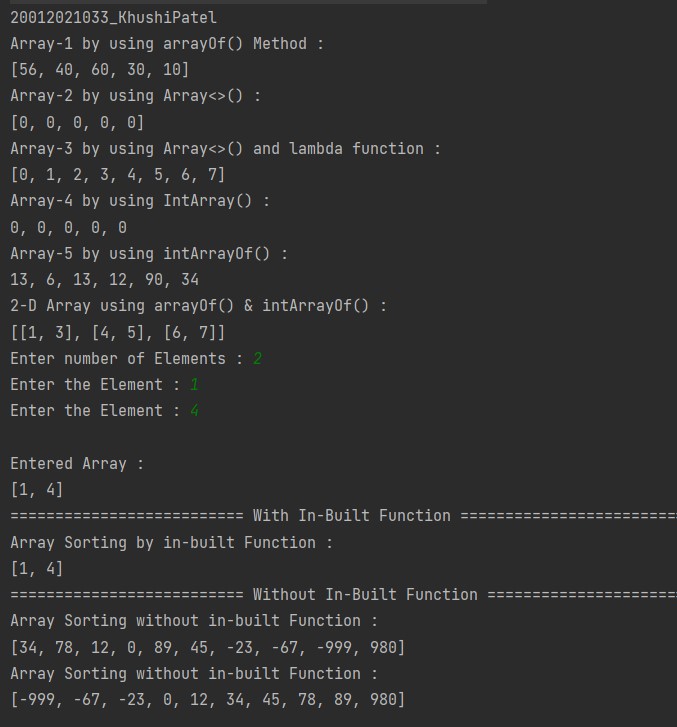
8.Create different types of Array as shown in image. Explore Arrays.deepToString(), contentDeepToString() methods, IntArray variable .joinToString() and use in program to print Array. Explore range, downTo, until etc. for loop and use in this program. Sort Array of Integer data type without using inbuilt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



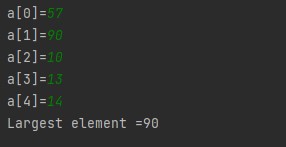
Code:

|  |
| --- |
| fun main() {  *println*("Array-1 by using arrayOf() Method : ") val a1 = *arrayOf*(56, 40, 60, 30, 10) *println*(a1.*contentToString*())  *println*("Array-2 by using Array<>() : ") val a2 = Array(5) **{** 0 **}** *println*(a2.*contentToString*())    *println*("Array-3 by using Array<>() and lambda function : ") |
| val a3 = Array(8) **{** i **->** i **}** *println*(a3.*contentToString*())    *println*("Array-4 by using IntArray() : ") val a4 = IntArray(5)  *println*(a4.*joinToString*(", "))  *println*("Array-5 by using intArrayOf() : ") val a5 = *intArrayOf*(13, 6, 13, 12, 90, 34) *println*(a5.*joinToString*(", "))    *println*("2-D Array using arrayOf() & intArrayOf() : ")  val a6 = *arrayOf*(*intArrayOf*(1, 3), *intArrayOf*(4, 5), *intArrayOf*(6, 7)) *println*(a6.*contentDeepToString*())  *print*("Enter number of Elements : ") val size: Int = *readLine*()!!.*toInt*() val a7 = IntArray(size) **{** 0 **}**    for (i in 0 *until* size) { *print*("Enter the Element : ") a7[i] = *readLine*()!!.*toInt*()  } *println*("\nEntered Array : ") *println*(a7.*contentToString*())    *println*("========================== With In-Built Function  ==========================")  *println*("Array Sorting by in-built Function : ") a7.*sort*()  *println*(a7.*contentToString*())  val a8 = *intArrayOf*(34, 78, 12, 0, 89, 45, -23, -67, -999, 980) *println*("========================== Without In-Built Function  ==========================")  *println*("Array Sorting without in-built Function : ") *println*(a8.*contentToString*())    var temp: Int for (i in a8.*indices*) { for (j in a8.*indices*) { if (a8[j] > a8[i]) { temp = a8[j] a8[j] = a8[i] a8[i] = temp  }  } }  *println*("Array Sorting without in-built Function : ") *println*(a8.*contentToString*()) } |

Output:



9.Find the max number from ArrayList.



Code:

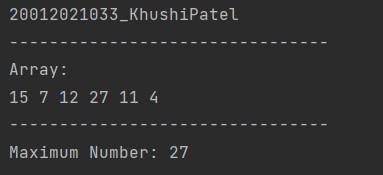
fun main(args:Array<String>){

*println*("--------------------------------")

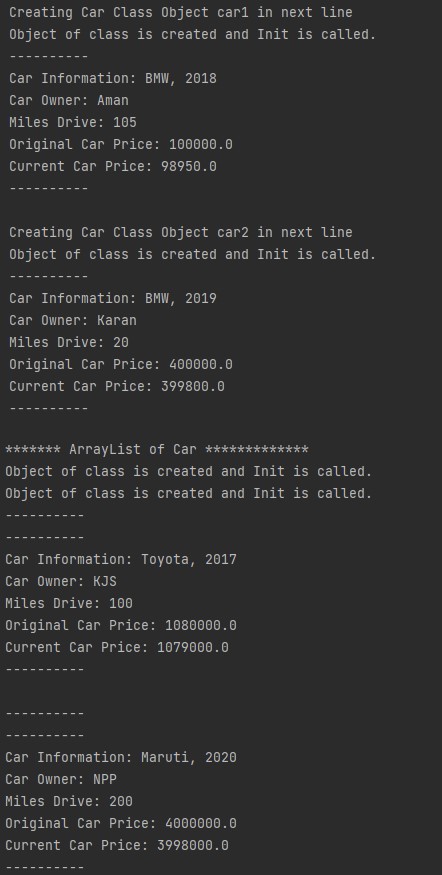
var ary:IntArray = *intArrayOf*(15, 7, 12, 27, 11, 4) var max\_num = ary[0]

|  |
| --- |
| for (i in ary){ if (i > max\_num){ max\_num = i  } }  *println*("Array:") for(i in ary){ *print*(""+ i +" ")  }  *println*("\n--------------------------------") *println*("Maximum Number: "+max\_num) } |

Output:



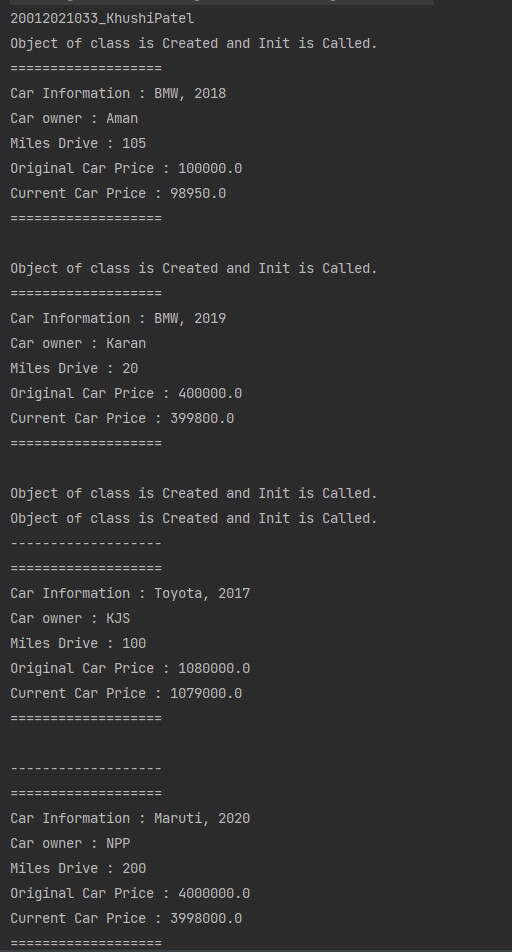
1. Write Different types of Class & Constructor. Create a class Car and set various members like type, model, price, owner. add the function getCarPrice in it. Create an object of Car class and access property of it. (getCarInformation(), getOriginalCarPrice(), getCurrentCarPrice(), displayCarInfo()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



Code:

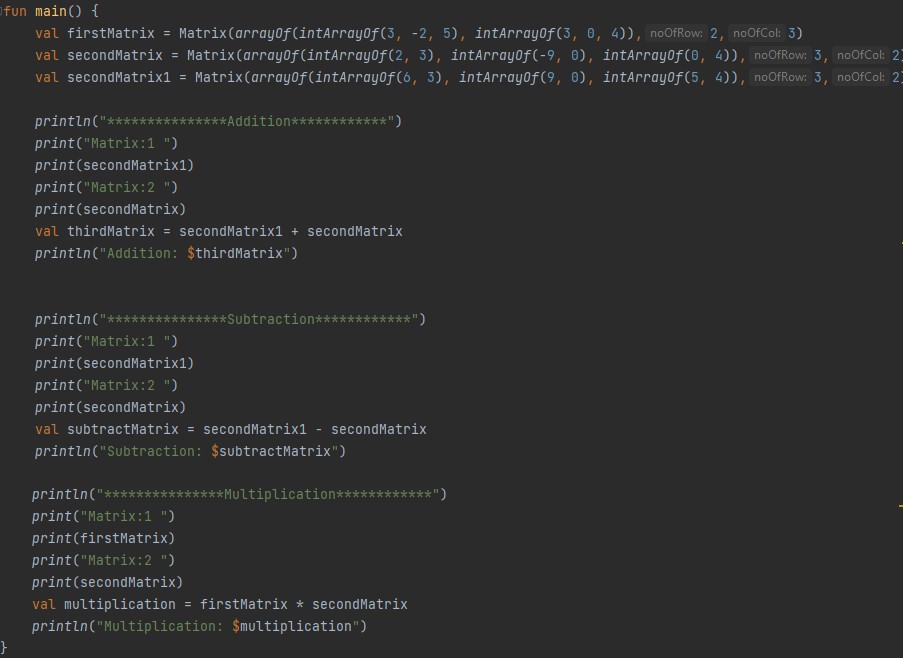
|  |
| --- |
| fun main() {  val car1 = Car("BMW, 2018", "Aman", 105, 100000.0, 98950.0) car1.getCarFullDetails()    val car2 = Car("BMW, 2019", "Karan", 20, 400000.0, 399800.0) car2.getCarFullDetails()    val Cars = ArrayList<Car> (2)  val car3 = Car("Toyota, 2017","KJS",100,1080000.0,1079000.0) val car4 = Car("Maruti, 2020","NPP",200,4000000.0,3998000.0)  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")  }  } |

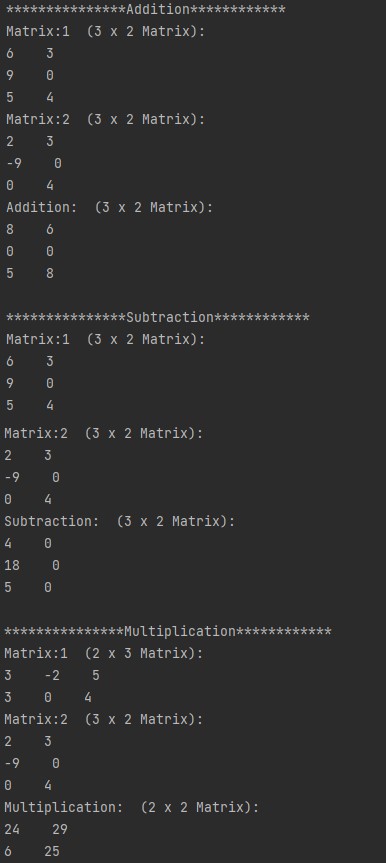
Output:



1. Write about Operator Overloading. Perform Matrix Addition, Subtraction & Multiplication using Class & operator overloading.

Overload toString() function in Matrix class.

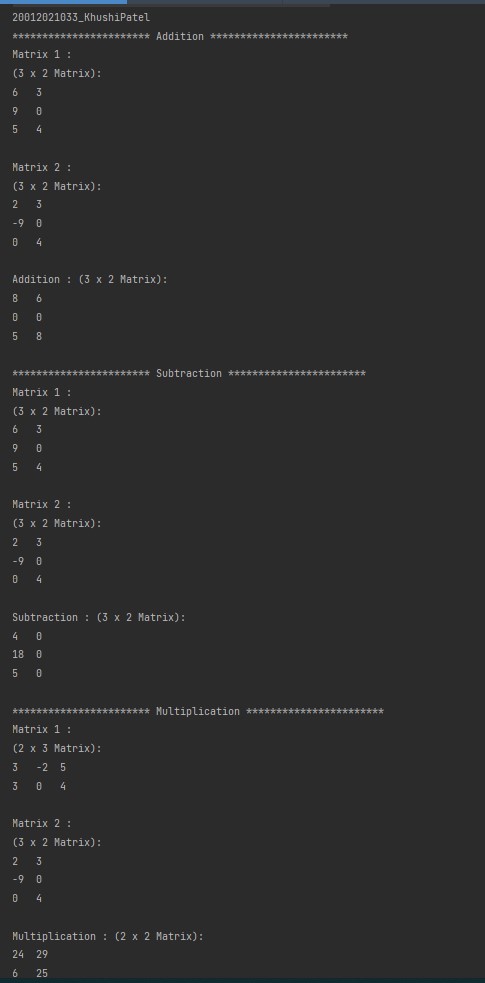




Code:

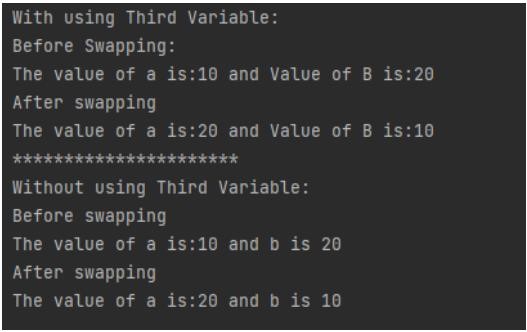
|  |
| --- |
| class Matrix(private val matrix: Array<IntArray>, private val rows: Int, private val cols: Int) {    override fun toString(): String { var res = "($rows x $cols Matrix): \n" for (i in matrix) { for (j in i) { res += "$j\t"  }  res += "\n"  }  return res  }  operator fun plus(obj: Matrix): Matrix {  val sum = Array(this.rows) **{** IntArray(this.cols) **}**    for (i in 0 *until* this.rows) { for (j in 0 *until* this.cols) {  sum[i][j] = this.matrix[i][j] + obj.matrix[i][j]  } }  return Matrix(sum, this.rows, this.cols)  }  operator fun minus(obj: Matrix): Matrix {  val sub = Array(this.rows) **{** IntArray(this.cols) **}**    for (i in 0 *until* this.rows) { for (j in 0 *until* this.cols) {  sub[i][j] = this.matrix[i][j] - obj.matrix[i][j]  } }  return Matrix(sub, this.rows, this.cols)  }  operator fun times(obj: Matrix): Matrix {  val mul = Array(this.rows) **{** IntArray(obj.cols) **}**    for (i in 0 *until* this.rows) { for (j in 0 *until* obj.cols) { mul[i][j] = 0  for (k in 0..obj.cols) {  mul[i][j] += this.matrix[i][k] \* obj.matrix[k][j]  }  } }  return Matrix(mul, this.rows, obj.cols)  }  }  fun main() {  val firstMatrix = Matrix(*arrayOf*(*intArrayOf*(3, -2, 5), *intArrayOf*(3, 0,  4)), 2, 3)  val secondMatrix1 = Matrix(*arrayOf*(*intArrayOf*(2, 3), *intArrayOf*(-9, 0), *intArrayOf*(0, 4)), 3, 2)  val secondMatrix2 = Matrix(*arrayOf*(*intArrayOf*(6, 3), *intArrayOf*(9, 0), *intArrayOf*(5, 4)), 3, 2)  *println*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Addition \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*") *println*("Matrix 1 : ") |
| *println*(secondMatrix2.toString()) *println*("Matrix 2 : ")  *println*(secondMatrix1.toString())  val addMatrix = secondMatrix2 + secondMatrix1 *println*("Addition : $addMatrix")    *println*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Subtraction \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*") *println*("Matrix 1 : ")  *println*(secondMatrix2.toString()) *println*("Matrix 2 : ")  *println*(secondMatrix1.toString())  val subMatrix = secondMatrix2 - secondMatrix1 *println*("Subtraction : $subMatrix")    *println*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Multiplication  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*") *println*("Matrix 1 : ") *println*(firstMatrix.toString()) *println*("Matrix 2 : ")  *println*(secondMatrix1.toString()) val mulMatrix = firstMatrix \* secondMatrix1 *println*("Multiplication : $mulMatrix") } |

Output:



Exercises:

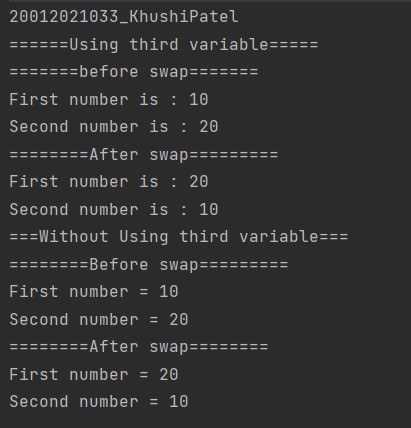
* 1. Swap Value of two variables without using third variable and with using third variable.



Code:

|  |
| --- |
| fun main()  {  *println*("======Using third variable=====") var first = 10 var second = 20  *println*("=======before swap=======") *println*("First number is : $first") *println*("Second number is : $second") val temporary = first first = second second = temporary  *println*("========After swap=========") *println*("First number is : $first") *println*("Second number is : $second")    *println*("===Without Using third variable===") var one = 10 var two = 20  *println*("========Before swap=========") *println*("First number = $one") *println*("Second number = $two")    one = one - two two = one + two one = two - one    *println*("========After swap========") *println*("First number = $one") *println*("Second number = $two") } |

Output:



* 1. 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.

* 1. 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.