**Calculator.scala:**

object BasicCalculator {

def calculate(a: Double, b: Double, operator: Char): Double = {

operator match {

case '+' => a + b

case '-' => a - b

case '\*' => a \* b

case '/' =>

if (b != 0) a / b

else throw new ArithmeticException("Cannot divide by zero.")

case \_ => throw new IllegalArgumentException("Invalid operator.")

}

}

}

val num1 = 10.0

val num2 = 5.0

val op = '+'

val result = BasicCalculator.calculate(num1, num2, op)

println(s"Result: $result")

**Positive\_Negative\_number.scala:**

val number = 5 // replace this manually each time

val result = if (number > 0) "Positive"

else if (number < 0) "Negative"

else "Zero"

println(s"$number is $result")

**Prime\_Number\_Checker.scala:**

// Function to check if a number is prime

def isPrime(n: Int): Boolean = {

if (n <= 1) false

else if (n == 2) true

else !(2 to math.sqrt(n).toInt).exists(n % \_ == 0)

}

// Sample list of numbers

val numbers = Seq(2, 3, 4, 5, 10, 13, 17, 20, 23, 25)

// Parallelize as RDD

val rdd = spark.sparkContext.parallelize(numbers)

// Apply isPrime and print results

val result = rdd.map(n => (n, isPrime(n)))

result.collect().foreach {

case (num, true) => println(s"$num is a Prime number")

case (num, false) => println(s"$num is NOT a Prime number")

}