```
package assignments

object Assign3 {

  def main(ary: Array[String]): Unit = {
    val calci = new calculator

    // Add rational numbers
    println("7/6 + 2/9   = " + calci.mksrting(
     calci.add(new rational(7, 6), new rational(2, 9))))
    println("5/2 - 4/7   = " + calci.mksrting(
     calci.sub(new rational(5, 2), new rational(4, 7))))
    println("3/8 * -9/11 = " + calci.mksrting(
     calci.mul(new rational(3, 8), new rational(-9, 11))))
    println("11/12 / 7/4 = " + calci.mksrting(
     calci.div(new rational(11, 12), new rational(7, 4))))

    //Auxiliary Constructor
    println("11/1 / 7/1  = " + calci.mksrting(
     calci.div(new rational(11), new rational(7))))

    //Whole Numbers
    println("25 + 15     = " + calci.add(25, 15))
    println("40 - 15     = " + calci.sub(40, 15))
    println("34 * 23     = " + calci.mul(34, 23))
    println("40 / 2      = " + calci.div(40, 2))

  }

}

class calculator {
  println("Calculator for Rational numbers and Whole numbers")
  def add(a: rational, b: rational): rational = {
    // = > n1 / d1 + n2 / d2 = (n1 * d2 + n2 * d1) / (d1 * d2)
    new rational(a.num * b.denom + b.num * a.denom,
                 a.denom * b.denom)
  }
  def sub(a: rational, b: rational): rational = {
    // = > n1 / d1 - n2 / d2 = (n1 * d2 - n2 * d1) / (d1 * d2)
    new rational(a.num * b.denom - b.num * a.denom,
                 a.denom * b.denom)
  }
  def mul(a: rational, b: rational): rational = {
    // = > n1 / d1 * n2 / d2 = (n1 * n2 ) / (d1 * d2)
    new rational(a.num * b.num, a.denom * b.denom)
  }
  def div(a: rational, b: rational): rational = {
    // = > n1 / d1 / n2 / d2 = (n1 * d2) / (d1 * n2)
```

```
    new rational(a.num * b.denom, b.num * a.denom)
  }

  //overload for whole numbers
  def add(a: Int, b: Int): Int = {
    a + b
  }
  def sub(a: Int, b: Int): Int = {
    a - b
  }
  def mul(a: Int, b: Int): Int = {
    a * b
  }
  def div(a: Int, b: Int): Double = {
    a / b
  }

  def mksrting(r: rational): String = {
    r.num + "/" + r.denom
  }

}

class rational(a: Int, b: Int) {

  val gcd: Int = gcd(a, b)
  val num = a / gcd
  val denom = b / gcd

  // Auxiliary constructor
  def this(a: Int) = this(a, 1)

  def gcd(a: Int, b: Int): Int = {
    if (a == 0) b else gcd(b % a, a)
  }

}
```

Console Output:

```
Calculator for Rational numbers and Whole numbers
7/6 + 2/9   = 25/18
5/2 - 4/7   = 27/14
3/8 * -9/11 = -27/88
11/12 / 7/4 = 11/21
11/1 / 7/1  = 11/7
25 + 15     = 40
40 - 15     = 25
34 * 23     = 782
40 / 2      = 20.0
```