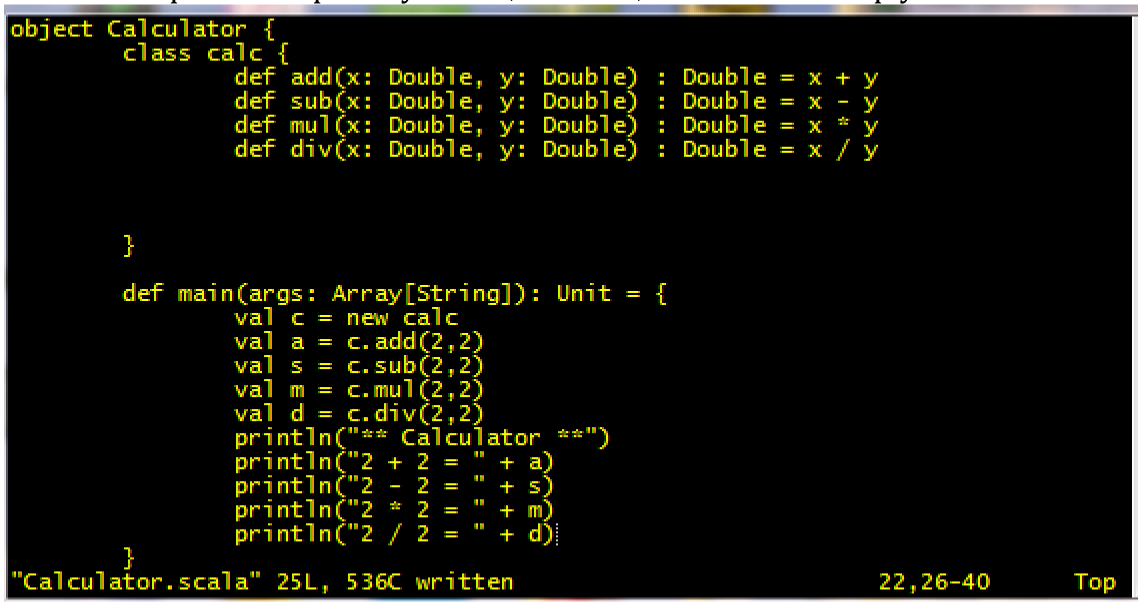
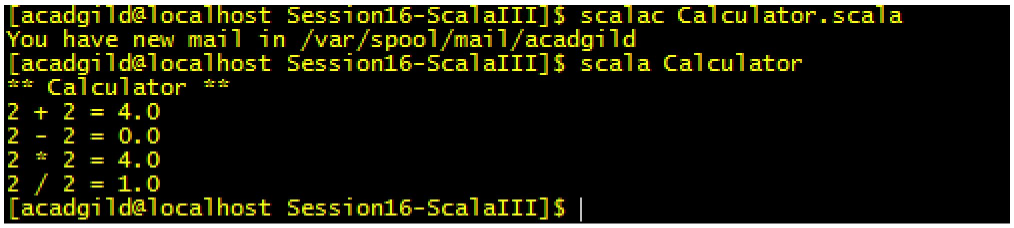
Session 16: Assignment 1

Task 1

Create a calculator to work with rational numbers.
Requirements:
➢ It should provide capability to add, subtract, divide and multiply rational Numbers

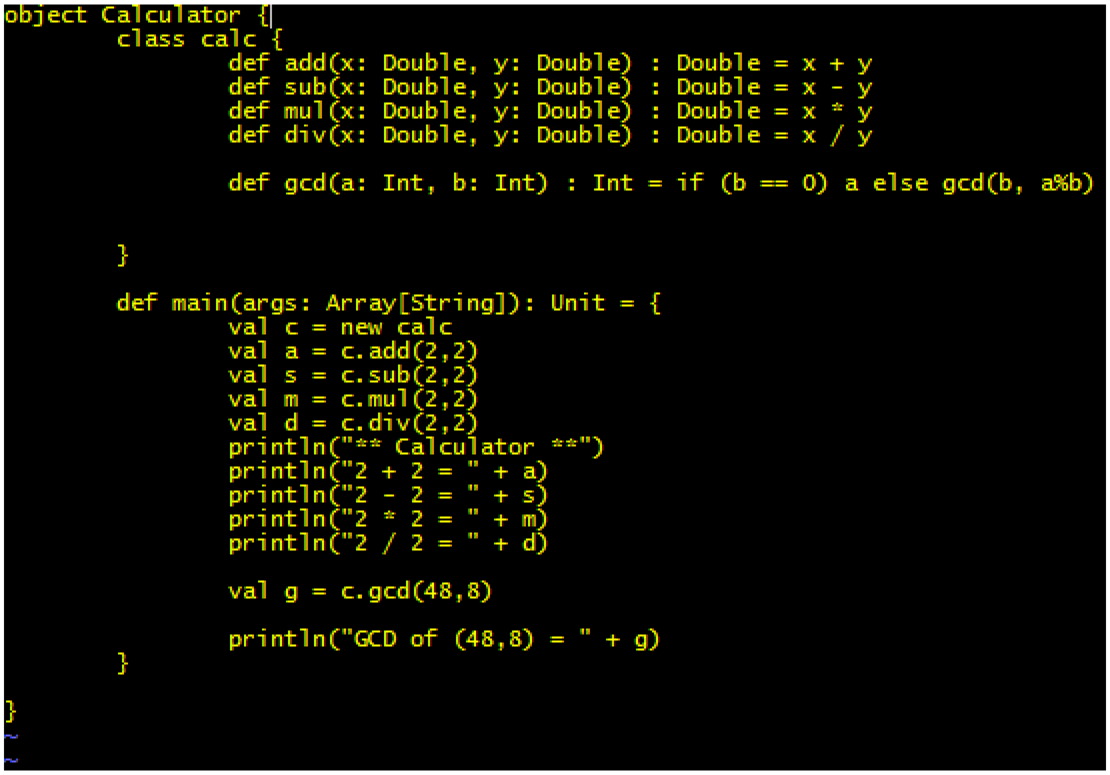


```
object Calculator {
        class calc {
                def add(x: Double, y: Double) : Double = x + y
                def sub(x: Double, y: Double) : Double = x - y
                def mul(x: Double, y: Double) : Double = x * y
                def div(x: Double, y: Double) : Double = x / y



        }

        def main(args: Array[String]): Unit = {
                val c = new calc
                val a = c.add(2,2)
                val s = c.sub(2,2)
                val m = c.mul(2,2)
                val d = c.div(2,2)
                println("** Calculator **")
                println("2 + 2 = " + a)
                println("2 - 2 = " + s)
                println("2 * 2 = " + m)
                println("2 / 2 = " + d)
        }
"Calculator.scala" 25L, 536C written                          22,26-40      Top
```

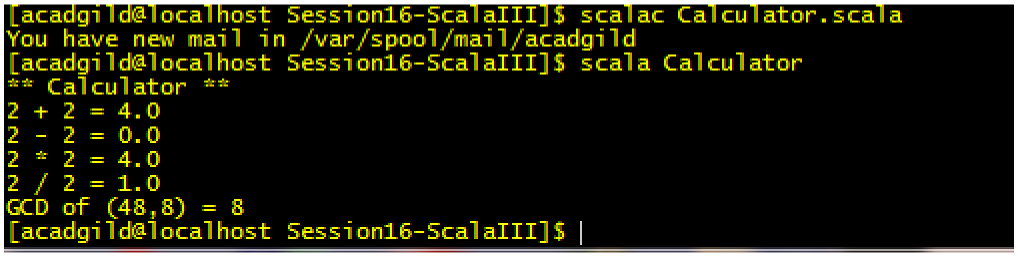


```
[acadgild@localhost Session16-ScalaIII]$ scalac Calculator.scala
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Session16-ScalaIII]$ scala Calculator
** Calculator **
2 + 2 = 4.0
2 - 2 = 0.0
2 * 2 = 4.0
2 / 2 = 1.0
[acadgild@localhost Session16-ScalaIII]$ |
```

➢ Create a method to compute GCD (this will come in handy during operations on rational)



```
object Calculator {
        class calc {
                def add(x: Double, y: Double) : Double = x + y
                def sub(x: Double, y: Double) : Double = x - y
                def mul(x: Double, y: Double) : Double = x * y
                def div(x: Double, y: Double) : Double = x / y

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


        }

        def main(args: Array[String]): Unit = {
                val c = new calc
                val a = c.add(2,2)
                val s = c.sub(2,2)
                val m = c.mul(2,2)
                val d = c.div(2,2)
                println("** Calculator **")
                println("2 + 2 = " + a)
                println("2 - 2 = " + s)
                println("2 * 2 = " + m)
                println("2 / 2 = " + d)

                val g = c.gcd(48,8)

                println("GCD of (48,8) = " + g)
        }

}
~
~
```

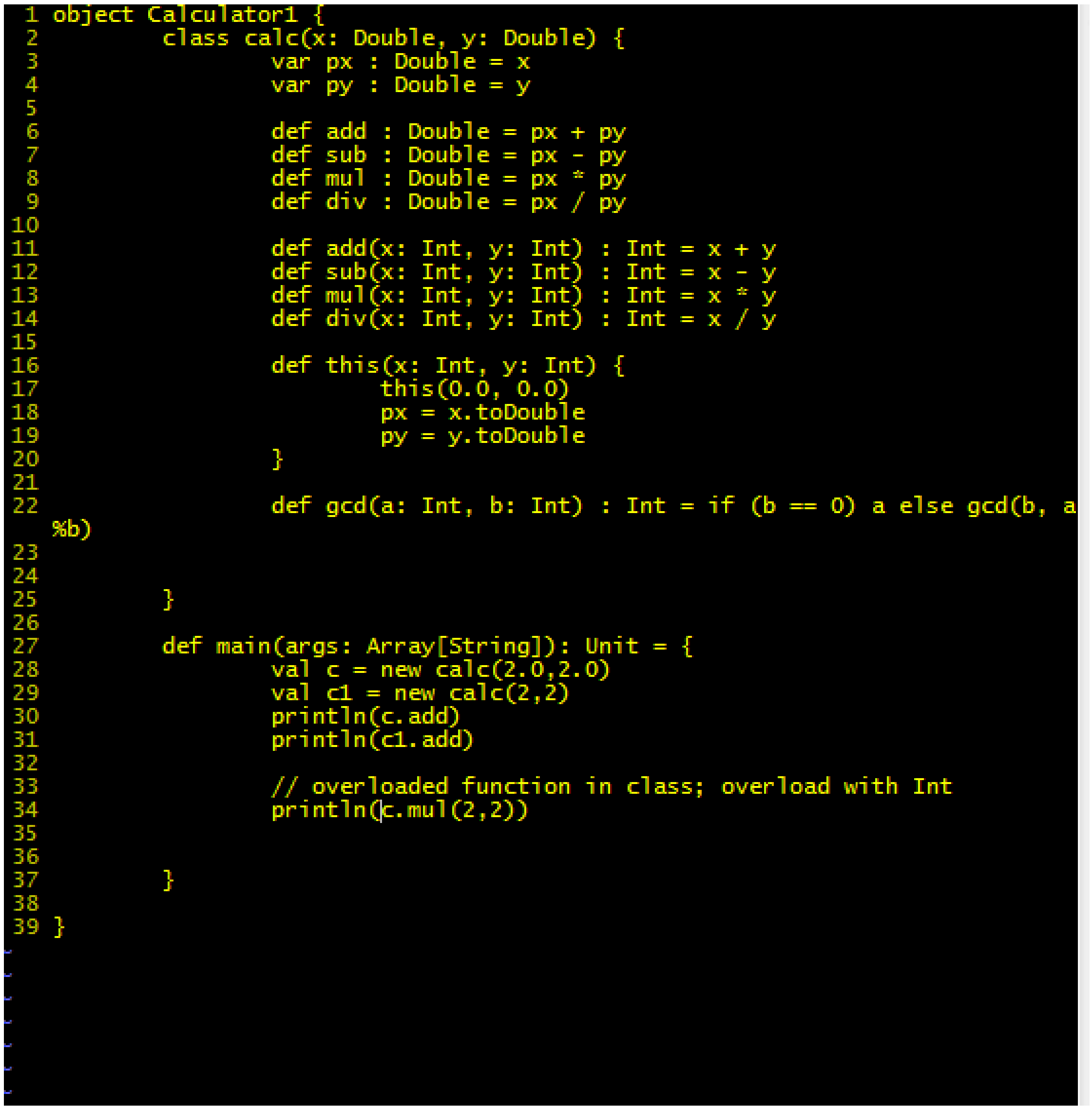


```
[acadgild@localhost Session16-ScalaIII]$ scalac Calculator.scala
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Session16-ScalaIII]$ scala Calculator
** Calculator **
2 + 2 = 4.0
2 - 2 = 0.0
2 * 2 = 4.0
2 / 2 = 1.0
GCD of (48,8) = 8
[acadgild@localhost Session16-ScalaIII]$ |
```

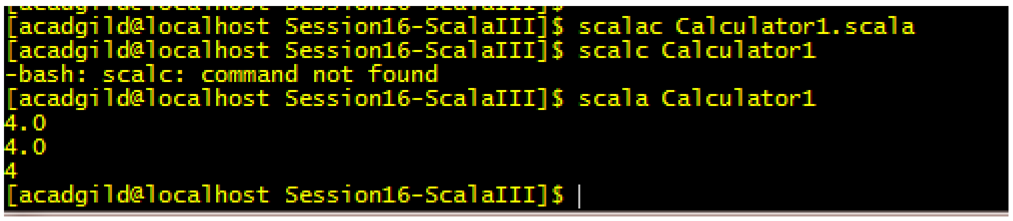
Add option to work with whole numbers which are also rational numbers i.e. (n/1)

➢ achieve the above using auxiliary constructors

➢ enable method overloading to enable each function to work with numbers and rational.



```scala
object Calculator1 {
        class calc(x: Double, y: Double) {
                var px : Double = x
                var py : Double = y

                def add : Double = px + py
                def sub : Double = px - py
                def mul : Double = px * py
                def div : Double = px / py

                def add(x: Int, y: Int) : Int = x + y
                def sub(x: Int, y: Int) : Int = x - y
                def mul(x: Int, y: Int) : Int = x * y
                def div(x: Int, y: Int) : Int = x / y

                def this(x: Int, y: Int) {
                        this(0.0, 0.0)
                        px = x.toDouble
                        py = y.toDouble
                }

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


        }

        def main(args: Array[String]): Unit = {
                val c = new calc(2.0,2.0)
                val c1 = new calc(2,2)
                println(c.add)
                println(c1.add)

                // overloaded function in class; overload with Int
                println(c.mul(2,2))


        }

}
```



```
[acadgild@localhost Session16-ScalaIII]$ scalac Calculator1.scala
[acadgild@localhost Session16-ScalaIII]$ scalc Calculator1
-bash: scalc: command not found
[acadgild@localhost Session16-ScalaIII]$ scala Calculator1
4.0
4.0
4
[acadgild@localhost Session16-ScalaIII]$ |
```