Create a calculator to work with rational numbers.

Requirements:

○ It should provide capability to add, subtract, divide and multiply rational

numbers

○ Create a method to compute GCD (this will come in handy during operations on

rational)

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.

**Solution:**

**Calc.scala**

class Calc(n:Int,d:Int)

{

require(d!=0)

private val g=gcd(n.abs,d.abs)

var num=n/g

var den =d/g

private def gcd(x:Int, y:Int) :Int=

{if(x==0) y else if (x<0) gcd(-x,y) else if (y<0) gcd(x,-y) else gcd(y%x,x)}

def this(n: Int ) = this(n,1) //auxiliary construct

def add (r:Calc):Calc =new Calc(num\*r.den+r.num\*den,den\*r.den)

def add (i:Int): Calc =new Calc(num +i\*den,den) //method overloading for add

def substract (r:Calc):Calc =new Calc(num\*r.den-r.num\*den,den\*r.den)

def substract (i:Int): Calc =new Calc(num -i\*den,den) //method overloading for substract

def multiply (r:Calc):Calc =new Calc(num\*r.num, den\*r.den)

def multiply (i:Int): Calc =new Calc(num \*i,den) //method overloading for multiplication

def divide (r:Calc):Calc =new Calc(num\*r.den, den\*r.num)

def divide (i:Int): Calc =new Calc(num,den\*1) //method overloading for division

override def toString:String=num+ "/" +den

}

**CalcObj.scala**

object calcObj

{

def main(args:Array[String]):Unit=

{

val a = new Calc(22,25)

val b = new Calc(19)

val c = new Calc(33,15)

val d = new Calc(13)

val p = a add 5

println(p)

val q = b multiply new Calc(13,25)

println(q)

val r = c substract new Calc(14,1)

println(r)

val s =d divide 51

println(s)

}

}

**OutPut**

