Vijay’s Assignment – Scala3

Task 1

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)

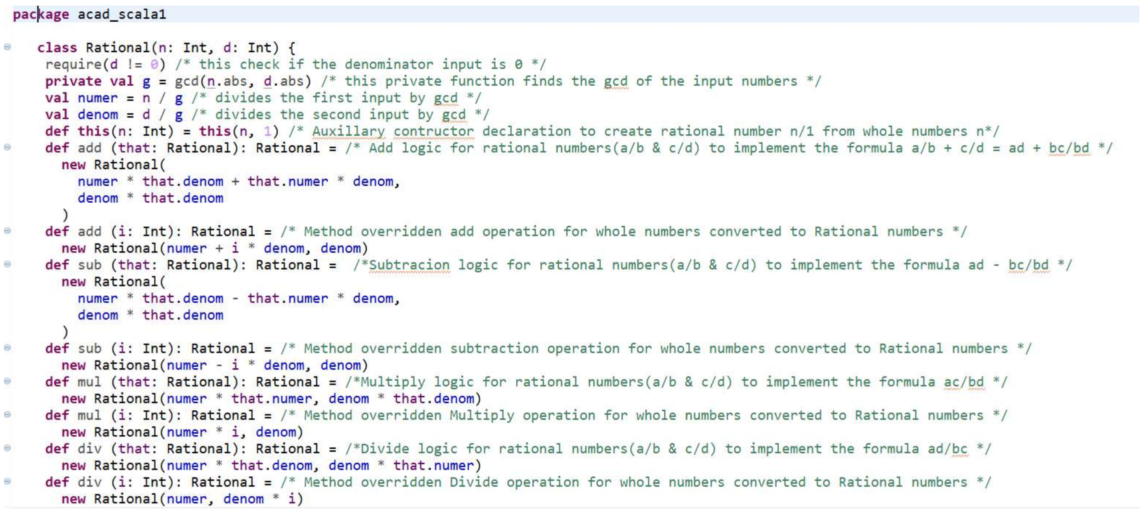
➢ achieve the above using auxiliary constructors

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

rational.

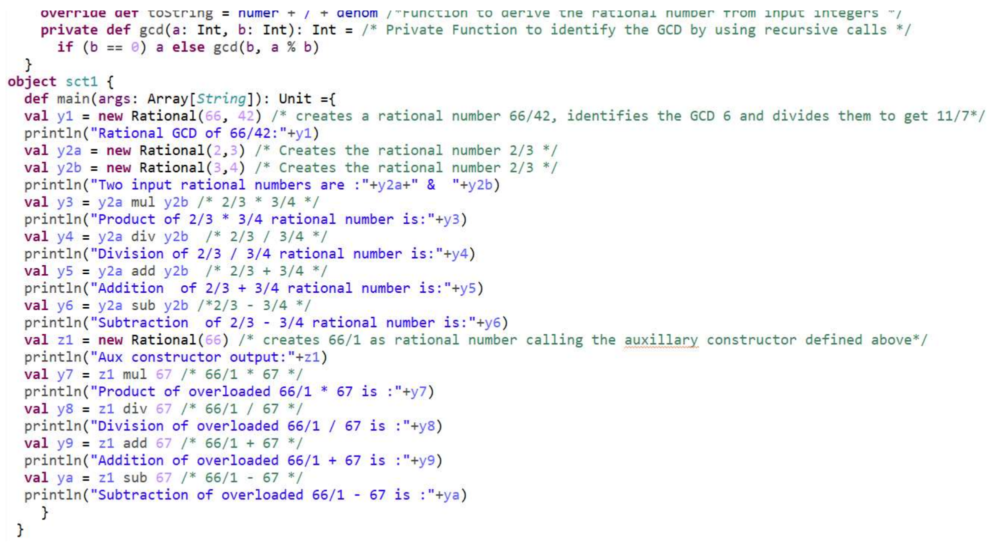
Actual code:

Explanation provided next to each statement as a comment.



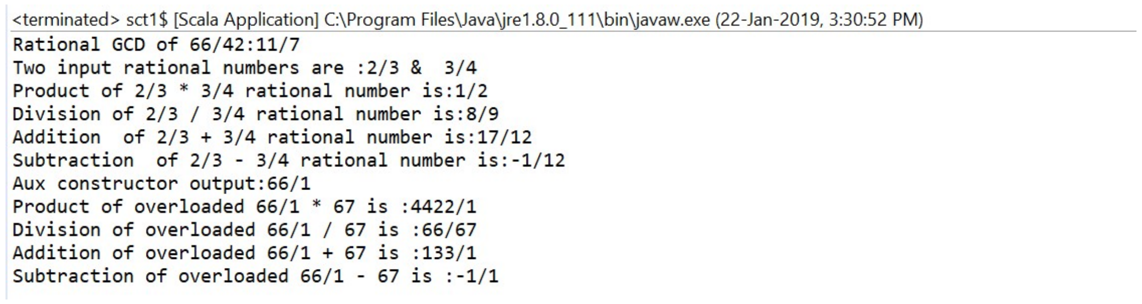
```
package acad_scala1

 class Rational(n: Int, d: Int) {
  require(d != 0) /* this check if the denominator input is 0 */
  private val g = gcd(n.abs, d.abs) /* this private function finds the gcd of the input numbers */
  val numer = n / g /* divides the first input by gcd */
  val denom = d / g /* divides the second input by gcd */
  def this(n: Int) = this(n, 1) /* Auxillary contructor declaration to create rational number n/1 from whole numbers n*/
  def add (that: Rational): Rational = /* Add logic for rational numbers(a/b & c/d) to implement the formula a/b + c/d = ad + bc/bd */
    new Rational(
      numer * that.denom + that.numer * denom,
      denom * that.denom
    )
  def add (i: Int): Rational = /* Method overridden add operation for whole numbers converted to Rational numbers */
    new Rational(numer + i * denom, denom)
  def sub (that: Rational): Rational =  /*Subtracion logic for rational numbers(a/b & c/d) to implement the formula ad - bc/bd */
    new Rational(
      numer * that.denom - that.numer * denom,
      denom * that.denom
    )
  def sub (i: Int): Rational = /* Method overridden subtraction operation for whole numbers converted to Rational numbers */
    new Rational(numer - i * denom, denom)
  def mul (that: Rational): Rational = /*Multiply logic for rational numbers(a/b & c/d) to implement the formula ac/bd */
    new Rational(numer * that.numer, denom * that.denom)
  def mul (i: Int): Rational = /* Method overridden Multiply operation for whole numbers converted to Rational numbers */
    new Rational(numer * i, denom)
  def div (that: Rational): Rational = /*Divide logic for rational numbers(a/b & c/d) to implement the formula ad/bc */
    new Rational(numer * that.denom, denom * that.numer)
  def div (i: Int): Rational = /* Method overridden Divide operation for whole numbers converted to Rational numbers */
    new Rational(numer, denom * i)
```



```scala
    override def toString = numer + / + denom /*Function to derive the rational number from input integers */
    private def gcd(a: Int, b: Int): Int = /* Private Function to identify the GCD by using recursive calls */
      if (b == 0) a else gcd(b, a % b)
  }
object sct1 {
  def main(args: Array[String]): Unit ={
  val y1 = new Rational(66, 42) /* creates a rational number 66/42, identifies the GCD 6 and divides them to get 11/7*/
  println("Rational GCD of 66/42:"+y1)
  val y2a = new Rational(2,3) /* Creates the rational number 2/3 */
  val y2b = new Rational(3,4) /* Creates the rational number 2/3 */
  println("Two input rational numbers are :"+y2a+" &  "+y2b)
  val y3 = y2a mul y2b /* 2/3 * 3/4 */
  println("Product of 2/3 * 3/4 rational number is:"+y3)
  val y4 = y2a div y2b  /* 2/3 / 3/4 */
  println("Division of 2/3 / 3/4 rational number is:"+y4)
  val y5 = y2a add y2b  /* 2/3 + 3/4 */
  println("Addition  of 2/3 + 3/4 rational number is:"+y5)
  val y6 = y2a sub y2b /*2/3 - 3/4 */
  println("Subtraction  of 2/3 - 3/4 rational number is:"+y6)
  val z1 = new Rational(66) /* creates 66/1 as rational number calling the auxillary constructor defined above*/
  println("Aux constructor output:"+z1)
  val y7 = z1 mul 67 /* 66/1 * 67 */
  println("Product of overloaded 66/1 * 67 is :"+y7)
  val y8 = z1 div 67 /* 66/1 / 67 */
  println("Division of overloaded 66/1 / 67 is :"+y8)
  val y9 = z1 add 67 /* 66/1 + 67 */
  println("Addition of overloaded 66/1 + 67 is :"+y9)
  val ya = z1 sub 67 /* 66/1 - 67 */
  println("Subtraction of overloaded 66/1 - 67 is :"+ya)
    }
 }
```

Output



```
<terminated> sct1$ [Scala Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (22-Jan-2019, 3:30:52 PM)
Rational GCD of 66/42:11/7
Two input rational numbers are :2/3 &  3/4
Product of 2/3 * 3/4 rational number is:1/2
Division of 2/3 / 3/4 rational number is:8/9
Addition  of 2/3 + 3/4 rational number is:17/12
Subtraction  of 2/3 - 3/4 rational number is:-1/12
Aux constructor output:66/1
Product of overloaded 66/1 * 67 is :4422/1
Division of overloaded 66/1 / 67 is :66/67
Addition of overloaded 66/1 + 67 is :133/1
Subtraction of overloaded 66/1 - 67 is :-1/1
```