**Session 5**

**Assignment 5.4**

Student Name: Karthik K

Course: Big Data Hadoop & Spark Training

Task 1 – Create a Scala Class ***“Calc”***

Scala Code

**class** Calc (n:Int, d:Int)



{



*require*(d!=0)



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



**val** *num*= n/*g*

**

**val** *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 constructor



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



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



**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 formultiplication



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



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



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

**

}



The statement, ***“def this(n: Int) = this(n, 1) “*** is an auxiliary constructor, we have created an Object **“CalcObj”** to perform the above functions.

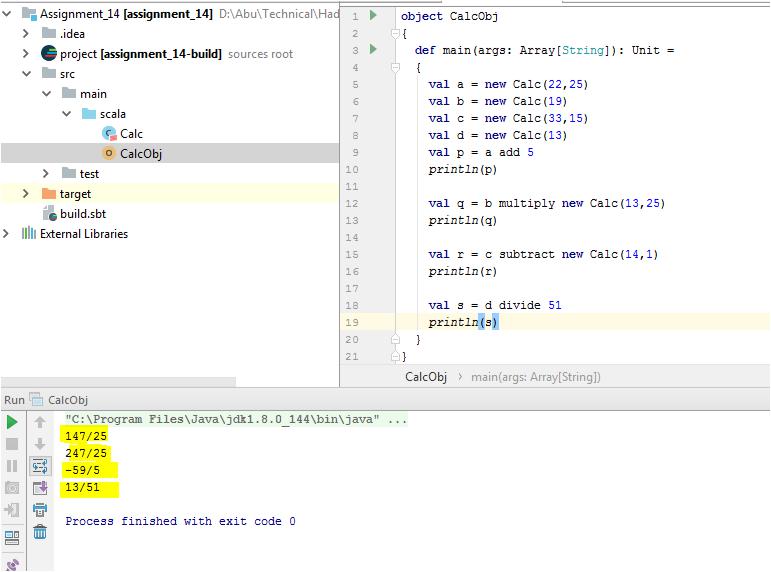
We have Enabled method **overloading** to enable each function (add, sub, multiplication and division) to work with numbers and rational.

We have written the code in such a way that it works with whole numbers as well as with rational numbers (n/1).

# 

Expected Output

1. Example 1,



2. Example 2,

