**Assignment 13.1**

**Problem Statement**

**Create a Scala application to find the GCD of two numbers.**

**Solution**

The greatest common divisor (gcd) of two or more integers, which are not all zero, is the largest positive integer that divides each of the integers. For example, the gcd of 8 and 12 is 4.

**The Euclidean algorithm**,

uses a division algorithm such as long division in combination with the observation that the gcd of two numbers also divides their difference.

To compute gcd(48,18),

1.divide 48 by 18 to get a quotient of 2 and a remainder of 12.

2.Then divide 18 by 12 to get a quotient of 1 and a remainder of 6.

3.Then divide 12 by 6 to get a remainder of 0, which means that 6 is the gcd.

Note that we ignored the quotient in each step except to notice when the remainder reached 0, signalling that we had arrived at the answer.

Formally the algorithm can be described as:

gcd(a,0)=a

gcd(a,b)=gcd(b,a mod b)

**Program**

**package** com.assignment.gcd

//Create a Scala application to find the GCD of two numbers.

//Greatest Common Divisor

**object** GCDObject {

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

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

**var** a=12

**var** b=16

print("Greatest Common Divisor of "+ a+" and "+b +" is : \t")

print(gcd(a, b))

}

}

