Q1. Create abstract class account having private member variable

a. Id [ it should be readonly property ,id should be generated by application]

b. Name [write getter setter and give validation length of name can not be >15]

c. Balance[write setter and balance property can be modify only by child class]

d. abstract double withdraw(int amt) declare abstract method.

e. create deposit method which will increase the balance.

Create two child class Saving and current. In saving account user has to maintain minimum balance of 1000. If user withdraw money ensure your application reduce balance after considering the rule of minimum balance. Write validation for withdraw 0 or -ve amount.

In saving account create static method public double payinterest() who’s job is to increase balance as well as return interest amount. Rate of interest is 0.08. This class has instance member Type which will be set to “Saving” for all user.

In Current account negative balance allowed. In this class create static method public double collect Interest() which will deduct 0.09% on overdraft facility ie. If there is -ve balance bank will charged interest, and reduce the balance. This class has instance member Type which will be set to value “Current”

When you load application it should print bank name.

In case of insufficient balance your application should write user define Exception, and put user Id, name and balance in a file. file name should created dynamically ie. File name should be the username and it should append all exception of that user only.

Your application should create only 5 object. If it create 6 th object it should throe exception.

Using parent reference store all child object in List class.

Declare const maxuser=5 , minbalance 1000, static double give\_interest =0.08, static collect\_interest = 0.09

Sort your data by username.

On every withdraw user should get sms and email [use event]

Using linQ display all user group by Type.

Using LinQ solve following query

a. display name of user having highest balance.

b, display all user who’s name is starting from “s”

c. display all user who’s balance is > 40000

d. display all user who’s balance is <20000

e. display all user

f. display only id and name of all user using anonymous object

g. display min , max balance.

Q2. Create a class math which has method factorial who’s job is to return factorial of a number.

Create two thread in a class “myclassthread” in this class create a static object of math class. Create two thread and call factorial method. Ensure only one thread execute method at atime. Each thread will pass different number to calculate factorial of a number.