Question A:

Requirements:

\* In C# create a single class that when subclassed allows this sample test code to run using only the file system for storage, no pre-built database allowed; use files.

\* Create all classes required to compile and pass the test case; you may not modify the test. The test is not wrong, it's not a trick.

\* The Id, Save, Delete, and Find methods must be in the base class only; subclasses must not implement these methods themselves.

\* Use nUnit 2.6 as newer versions lack some of these asserts.

\* Put your solution on Github and send me the clone endpoint. Create a branch called “Question\_A”.

Choose whatever types you feel appropriate, the purpose of the exercise is to see some of your code and discuss with you the solution you came up with and the choices you made.

[TestCase]

public void ProgrammerTest() {

var address = new Address("56 Main St", "Mesa", "AZ", "38574");

var customer = new Customer("John", "Doe", address);

var company = new Company("Google", address);

Assert.IsNullOrEmpty(customer.Id);

customer.Save();

Assert.IsNotNullOrEmpty(customer.Id);

Assert.IsNullOrEmpty(company.Id);

company.Save();

Assert.IsNotNullOrEmpty(company.Id);

Customer savedCustomer = Customer.Find(customer.Id);

Assert.IsNotNull(savedCustomer);

Assert.AreSame(customer.Address, address);

Assert.AreEqual(savedCustomer.Address, address);

Assert.AreEqual(customer.Id, savedCustomer.Id);

Assert.AreEqual(customer.FirstName, savedCustomer.FirstName);

Assert.AreEqual(customer.LastName, savedCustomer.LastName);

Assert.AreEqual(customer, savedCustomer);

Assert.AreNotSame(customer, savedCustomer);

Company savedCompany = Company.Find(company.Id);

Assert.IsNotNull(savedCompany);

Assert.AreSame(company.Address, address);

Assert.AreEqual(savedCompany.Address, address);

Assert.AreEqual(company.Id, savedCompany.Id);

Assert.AreEqual(company.Name, savedCompany.Name);

Assert.AreEqual(company, savedCompany);

Assert.AreNotSame(company, savedCompany);

customer.Delete();

Assert.IsNullOrEmpty(customer.Id);

Assert.IsNull(Customer.Find(customer.Id));

company.Delete();

Assert.IsNullOrEmpty(company.Id);

Assert.IsNull(Company.Find(company.Id));

}

Question B:

1. What improvements can be made?

2. What SOLID principle(s) can help here?

3. What is the best architectural decision?

(submit answers in text document)

\* Put the document in the same repo on a branch called “Question\_B”.

class A

{

public A()

{

}

private void SolveBinomial()

{

// Do work ...

// ...

// Send email ...

string[] args = new string[0];

SmtpClient client = new SmtpClient(args[0]);

MailAddress from = new MailAddress("jane@contoso.com", "Jane Clayton", System.Text.Encoding.UTF8);

MailAddress to = new MailAddress("ben@contoso.com");

MailMessage message = new MailMessage(from, to);

message.Body = "Hi Ben, this is Jane, how's it going?";

client.SendCompleted += new SendCompletedEventHandler(binomialEventHandler);

string userState = "test message1";

client.SendAsync(message, userState);

message.Dispose();

}

}

class B

{

public B()

{

}

private void SolveTrinomial()

{

// Do work ...

// ...

// Send email ...

string[] args = new string[0];

SmtpClient client = new SmtpClient(args[0]);

MailAddress from = new MailAddress("bob@contoso.com", "Bob Doe", System.Text.Encoding.UTF8);

MailAddress to = new MailAddress("sarah@contoso.com");

MailMessage message = new MailMessage(from, to);

message.Body = "Hi Sarah, here are the calculations...";

client.SendCompleted += new SendCompletedEventHandler(trinomialEventHandler);

string userState = "test message1";

client.SendAsync(message, userState);

message.Dispose();

}

}

Question C:

Requirements:

\* I want to track user logins. I want to ONLY track the CustomerId, UserName, SaleId, and SaleAmount.

Develop a normalized Database Entity Relationship Diagram. Just draw this on a sheet of paper and take a picture on your camera phone (see attachment/link).

Notes

! There should be no more than 3 tables!

! There is a 1-n between User – Login.

\* Put the document in the same repo on a branch called “Question\_C”.