# Assignment 1

**Forum.java**

//Patrick Noonan(20355426)

**import** java.time.LocalDateTime;

**import** java.util.Scanner;

**import** java.util.UUID;

**public** **class** Forum {

LocalDateTime localDateTime;

**private** String uniqueID, forumID;

**private** String title, post;

**public** Forum() //set up three different constructors one which enters timestamp and one which automatically generates timestamp.

{

}

**public** Forum(String title, String post) //automatically generates

{

**this**.title = title; //takes in title reading

**this**.post = post; //takes in the post

localDateTime = LocalDateTime.*now*();

forumID = createUniqueID(); //creates posts unique ID

}

**public** Forum(String title, String post, LocalDateTime timestamp) **throws** TimeStampException //creates its own timestamp with user entry

{

**this**.title = title; //takes in title

**this**.post = post; //takes in post

**if**(correctTimeStamp(timestamp)) //if statement to check if timestamp is sometime in the future

{

timestamp = localDateTime;

}

**else** //if it is not in the future an exception will be thrown to tell user of invalid timestamp

{

**throw** **new** TimeStampException("Can't enter a time sooner then the present moment");

}

forumID = createUniqueID(); //creates posts unique ID

}

**public** **boolean** correctTimeStamp(LocalDateTime timestamp) //method which checks if timestamp is in the past or present

{

**if**(timestamp.isBefore(LocalDateTime.*now*())) //if timestamo entered is before present time return false else return true

{

**return** **false**;

}

**else** **return** **true**;

}

**public** String createUniqueID()

{

uniqueID = UUID.*randomUUID*().toString();

**return** uniqueID;

}

**public** String getForumID()

{

**return** forumID;

}

**public** String getTitle()

{

**return** title;

}

**public** LocalDateTime queryTimestamp()

{

**return** localDateTime;

}

**public** String queryTitle()

{

**return** title;

}

**public** String queryText()

{

**return** post;

}

**public** **void** editTitle(String newTitle)

{

**this**.title = newTitle;

}

**public** **void** editPost(String newPost)

{

**this**.post = newPost;

}

**public** String toString()

{

String forumPost = "";

forumPost += "TimeStamp: " +localDateTime;

forumPost += "Title: " +title;

forumPost += "Post: " +post;

**return** forumPost;

}

}

**ForumTest.java**

//Patrick Noonan(20355426)

import static org.junit.Assert.assertEquals;

import static org.junit.Assert.assertFalse;

import static org.junit.Assert.assertNotSame;

import static org.junit.Assert.assertThrows;

import static org.junit.Assert.assertTrue;

import static org.junit.Assert.fail;

import java.time.LocalDateTime;

import org.junit.jupiter.api.Test;

public class ForumTests {

@Test

void testAutoGeneratedTimestampConstructor() throws TimeStampException

{

Forum forum = new Forum("Mileage", "The colorado buffaloes clock mileage in the off season"); //creating a forum object using the auto generated timestamp constructor

assertFalse("The title is not equal to kilometrage",forum.getTitle().equals("kilometrage"));//seeing how the assertFalse assertion works

assertTrue(forum.getTitle().equals("Mileage")); //test if the title is equal to milage

assertEquals( forum.queryText(), "The colorado buffaloes clock mileage in the off season");//test is the post variable is being applied correctly and queryText() method is working correctly

assertNotSame( forum.queryTitle(), forum.queryText());

//tests if the timestamp is correct, I attempted to go between two certain timestamps as the .now() of the time of the code running and .now() of the test running maybe different

assertTrue(forum.queryTimestamp().isBefore(LocalDateTime.now()) || forum.queryTimestamp().isAfter(LocalDateTime.now().minusNanos(1000)));

}

@Test

void testConstructorWithThreeEntries() throws TimeStampException

{

LocalDateTime timestamp = LocalDateTime.now().plusDays(5);

Forum forums = new Forum("Driving licence", "You have to do 12 driving lessons to get your driving licenece", timestamp);

assertTrue(forums.getTitle().equals("Driving licence")); //test if the title is correctly assigned and method returns correct title

assertEquals(forums.queryText(), "You have to do 12 driving lessons to get your driving licenece" );//tests if the post is being assigned to correct variable and the method is returning correct variable

}

@Test

void testIfInvalidTimeIsReturned()

{

LocalDateTime timestamp = LocalDateTime.now().minusDays(5);

assertThrows(TimeStampException.class, () ->

{

Forum forums = new Forum("Driving licence", "You have to do 12 driving lessons to get your driving licenece", timestamp);

});

}

@Test

void testIfTheForumIDs()

{

Forum forum = new Forum("Noise Complaint", "The neighbours were too noisy last night");

Forum forummm = new Forum("Exam Results", "Exam results will be availible in the next two weeks");

assertFalse(forum.getForumID().equals(forummm.getForumID()) );

}

}

**TimeStampException.java**

//Patrick Noonan(20355426)

**public** **class** TimeStampException **extends** Exception{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** TimeStampException(String message)

{

**super**(message);

}

}

**Testing**

**Graphical user interface, text, application

Description automatically generated**

Not sure if you want screenshots of the test failing as I progressed through the assignment or not. I did not take these screenshots so I think this is all you want. Was sure what screenshots you are looking for from the brief.