1. Automation framework:

a. Using Ipro Tech website as the target, create a barebone framework with minimal components to test the Careers page UI. The solution should contains example(s) of:

i. an API method implementation

ii. locating an element on the webpage

iii. Gherkin and step definition

I have not had the opportunity to test APIs yet in my career. This is something I have tried to learn on my own. I have done tutorials with SOAP UI and Postman being the more recent one. I have done a few things in Postman in my current role like perform a GET request and verify the response status as 200. I am trying to learn more about this technology. I wasn’t sure what I would test with the information given above. Maybe with some more information I could have figure this out.

The attached solution contains the Web Elements and step definitions.

2. C# challenge questions:

a. Write a program that converts 1 lower case letter ("a" - "z") to its corresponding upper case letter ("A" - "Z"). For example if the user enters "c" then the program will show "C" on the screen.

1. See the attached code

b. Write a few snippets of code showing utilization of each of the pillars of OOP

1. Encapsulation

1. See Attached snippet

2. Inheritance

1. I didn’t write a snippet but an example of inheritance in a page object model framework would be how I use the page objects classes with the test class. The page objects contain all of the web element objects and methods. I can use all of these in my test cases using inheritance without having to write the methods over and over again.

3. Polymorphism

1. Polymorphism allows the same functions with the same scope to be used as long as the arguments are different

void print(int i) {

Console.WriteLine("Printing int: {0}", i );

}

void print(double f) {

Console.WriteLine("Printing float: {0}" , f);

}

4. Abstract

1. A good Abstract example would be like the extent reports in the framework I currently use. The tester never sees the code as it works behind the test cases and produces a report for each test executed. If the tester wants a screenshot of a particular step for verification, the extent report will take the screenshot and add it to the extent report for the particular test.

3. New tool learning:

a. Write an example of a JMeter project

1. JMeter is another technology I have not had the opportunity to work with yet. I understand the capabilities of it for Load and Strass testing applications. Load testing will test the expected number of users who might be using the service at the same time. Stress testing will take the server beyond its defined maximum load capacity. This is to test the maximum load the web server can handle and what types of errors are produced at high loads. I can see this as being extremely important in a number of applications like banking or any kind of online shopping

using System;

namespace CSharpExamples

{

class CharacterConverter

{

static void Main(string[] args)

{

Console.WriteLine("Converts all characters in a string from lowercase to uppercase and displays the converted string");

Console.WriteLine("Enter a string: ");

//Reads the string entered by the user

string UserStr = Console.ReadLine();

//Counts the length of the string

int StrLength = UserStr.Length;

//Puts the string into a character array to check each character for uppercase or lowercase

char[] StrArray = UserStr.ToCharArray(0, StrLength);

Console.WriteLine("You entered '" + UserStr + "' to be converted");

Console.WriteLine("Here is the conversion: ");

//This for loop will iterate through each character in the character array

for (int i = 0; i < StrLength; i++)

{

char ArrayChar = StrArray[i];

//This if statement will check each character if it is lowercase or uppercase

if (Char.IsLower(ArrayChar)) //If it is lowercase

{

Console.Write(Char.ToUpper(ArrayChar));//convert to uppercase

}

else

{

Console.Write(ArrayChar);//leave it as

}

}

Console.Write("\n\n");

}

}

}

using System;

namespace Example

{

class Encapsulation

{

//object attributes are kept private for encapsulation

//these attributes can only be accessed using public member functions

private double length; // Length of a box

private double width; // width of a box

private double height; // Height of a box

public void setLength(double len)

{

length = len;

}

public void setBreadth(double wid)

{

width = wid;

}

public void setHeight(double hei)

{

height = hei;

}

public double getVolume()

{

return length \* width \* height;

}

}

class Boxtester

{

static void Main(string[] args)

{

Box Box1 = new Box(); // Declare Box1 of type Box

Box Box2 = new Box();

double volume;

// Declare Box2 of type Box

// box 1 specification

Box1.setLength(6.0);

Box1.setBreadth(7.0);

Box1.setHeight(5.0);

// box 2 specification

Box2.setLength(12.0);

Box2.setBreadth(13.0);

Box2.setHeight(10.0);

// volume of box 1

volume = Box1.getVolume();

Console.WriteLine("Volume of Box1 : {0}", volume);

// volume of box 2

volume = Box2.getVolume();

Console.WriteLine("Volume of Box2 : {0}", volume);

Console.ReadKey();

}

}

}