NGEE ANN SCHOOL OF INFOCOMM TECHNOLOGY

PRG2	**Programming II** Diploma in CSF / IT / FI Year 1 (2020/21) Semester 2	Week **6** 45 minutes
	Practical Test (15%)	

Instructions

Prior to test

- Create a new *Console App (.Net Core)*.
- Name the project *POIApp_YourStudentID*
  (e.g. POIApp_S12345678 if your student ID is S12345678)

Submission

- Map to **Network drive : \\ictspace.ict.np.edu.sg\PRG2PracticalTest**
- Upload the WHOLE *POIApp_YourStudentID* folder into the network drive (***ictspace.ict.np.edu.sg > PRG2PracticalTest > group > studentID***)

Note: It is your RESPONSIBILITY to ensure that the files are submitted correctly.

1. Analyze the class element diagram below.

POI
-latitude: double -longitude: double -name: string -address: string
+POI() +POI(double, double, string, string) +DistanceFrom(POI):double +ToString(): string

Note:

- Each **POI** object represents a point of interest on a map.
- **`DistanceFrom()`** method computes the straight line distance between this object and another **POI** object that is passed in. The formula to compute the straight line distance between two **POI** objects is given by this formula

$$\texttt{distance} = \sqrt{(latitude_1 - latitude_2)^2 + (longitude_1 - longitude_2)^2}$$

where $(latitude_1, longitude_1)$ and $(latitude_2, longitude_2)$ are the coordinates of the 2 **POI** objects.

In C#, use the following method to compute the square root of a `double` variable, `num`:

```
double result = Math.Sqrt( num );
```

Implement the **`POI`** class based on the *Class Element Diagram* above. (20 marks)

2. Add the following to the Program class:

   a. Create a `List` to store **`POI`** objects in the `Main()` method.

```
class Program
{
    static void Main(string[] args)
    {
        // define your list here
```

(5 marks)

   b. Write and call a method to initialize your list of **`POI`** objects:
      i. Read the contents of "`attractions.csv`" (downloaded from MeL) and create the **`POI`** objects. The details in each line correspond to "latitude", "longitude", "name" and "address" attributes respectively.
      ii. Add all the **`POI`** objects to your list.

(20 marks)

   c. Write and call a method to display the details of all the objects in your list as shown below:

```
Name                          Address                                  (Latitude, Longitude)
----                          -------                                   --------, ---------
Universal Studios Singapore   8 Sentosa Gateway Singapore 098269       (1.254028, 103.824203)
Jurong Bird Park              2 Jurong Hill Singapore 628925           (1.318642, 103.706742)
Gardens by the Bay            18 Marina Gardens Dr Singapore 018953    (1.281461, 103.865186)
Esplanade                     80 Mandai Lake Rd Singapore 729826       (1.289911, 103.854894)
Singapore Science Centre      15 Science Centre Rd Singapore 609081    (1.333108, 103.735757)
Singapore Botanic Gardens     1 Cluny Rd Singapore 259569              (1.314431, 103.815667)
```

(12 marks)

d. Write a method to create a new **POI**. You need to prompt the user for the details of the new **POI** object, create it and return it to the main program. You are not required to do any data validation. This method is needed in part (e) below and refer to the user interface shown where values underlined are the user inputs:

```
Create POIObject
================
Latitude: 1.402649
Longitude: 103.788491
Name: Night Safari
Address: 80 Mandai Lake Rd Singapore 729826
```

(20 marks)

e. In the `Main()` method, insert a new **POI** into the current list of **POI** objects by calling the method in part (d) and thereafter adding the object into the list. **Display a suitable message to indicate the new object has been added and show the updated list to the user.**

(8 marks)

f. In the `Program` class, compute and display the nearest and furthest **POI** object from "`Esplanade`", as shown below:

```
The nearest POI from Esplanade is Gardens by the Bay
The furthest POI from Esplanade is Jurong Bird Park
```

(15 marks)

PLAGIARISM WARNING:

If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this practical test. Disciplinary action may also be taken.

Similar action will be taken for student who allows other student(s) to copy his/her work, or posting any solutions or code related to the practical test before the end of the hour for the test.

*** END OF PAPER ***