THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

DEPARTMENT OF LAND SURVEYING AND GEO-INFORMATICS
土地測量及地理資訊學系

2021-2022 Semester 2

LSGI3315: GIS Engineering

Subject Lecturer: Dr. Xu Yang

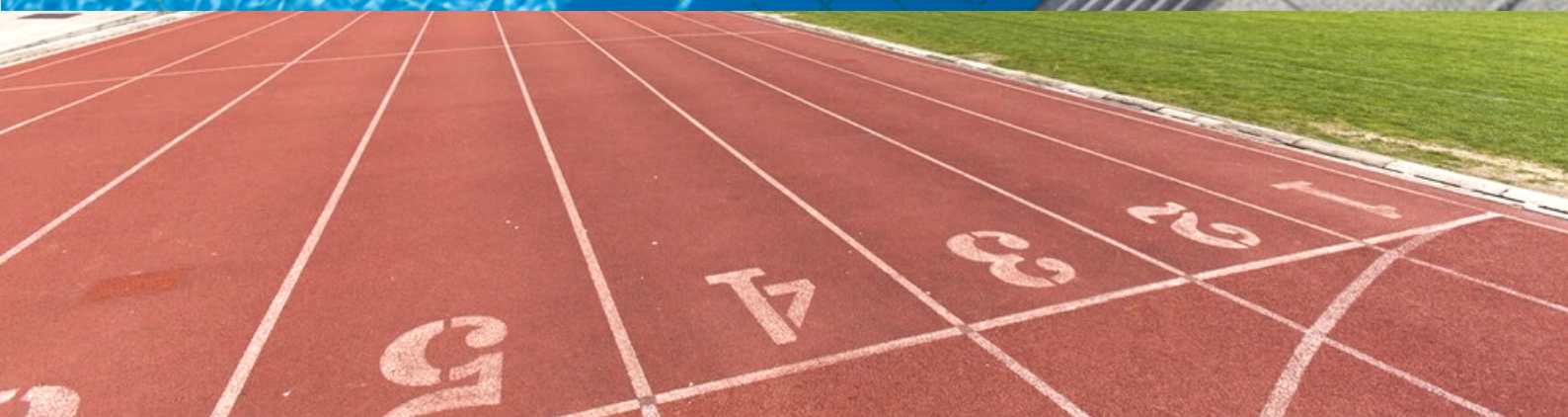
Tutor: Ms. Xinyue Chen and Ms. Mengqi Sun

Group 6 Project (Individual Section)

Task 2: Individual Components (15% Compulsory + 5% Bonus)

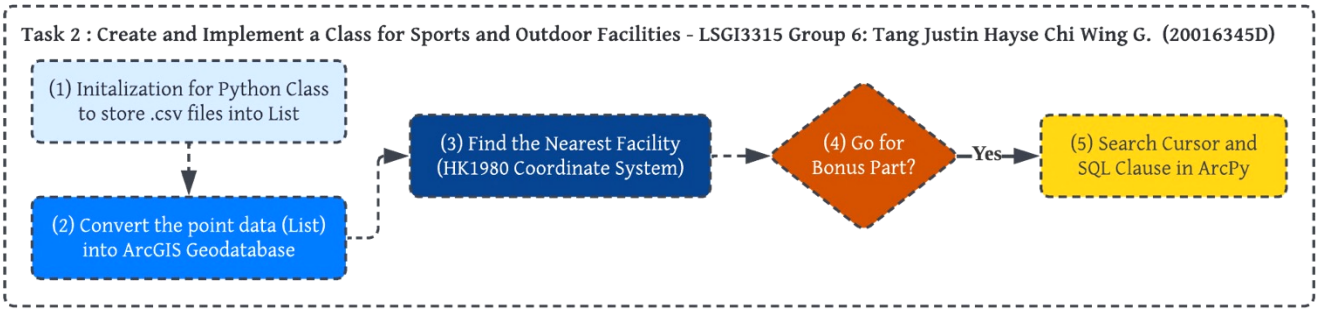
Create and Implement a Python Class for Sport and Outdoor Facilities in Hong Kong

Student: Tang Justin Hayse Chi Wing G. 20016345D



Introduction and Workflow of Task 2 (Only TWO-PAGE is Allowed for the Instruction)

After finishing data cleaning, Task 2 is required to implement a Python class for sports and outdoor facilities individually via ArcPy syntax. To complete missions, a Python script is written and the coding workflow is drawn as follows.



Instruction of How to Use the Python Code

Core (1) and (2): Initialization for Python Class to store .csv files into List; Convert point data into ArcGIS Geodatabase

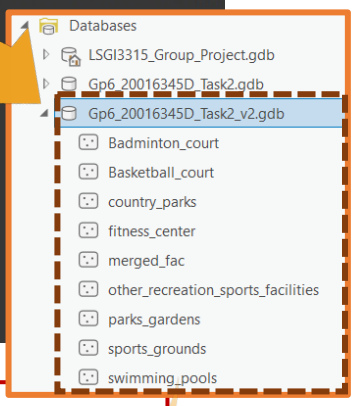
In this section, user can simply type the names of .csv files in the initialization function. The program subsets the data frame and saves the data in Python list. Afterwards, by using `arcpy.CreateFeatureclass_management` and `arcpy.da.InsertCursor`, the point data (in the list) will be inputted into the feature class in ArcGIS geodatabase successfully.

```
18  class facility:  # Create a Class named "facility"
19      def __init__(self):  # Initialization: the Class "facility"
20          # List out all csv files of Sports and Outdoor Facilities and define
21          csv = ['Badminton_court.csv', 'Basketball_court.csv', 'country_parks.csv', 'fitness_center.csv',
                   'other_recreation_sports_facilities.csv', 'parks_gardens.csv', 'sports_grounds.csv', 'swimming_pools.csv']
```

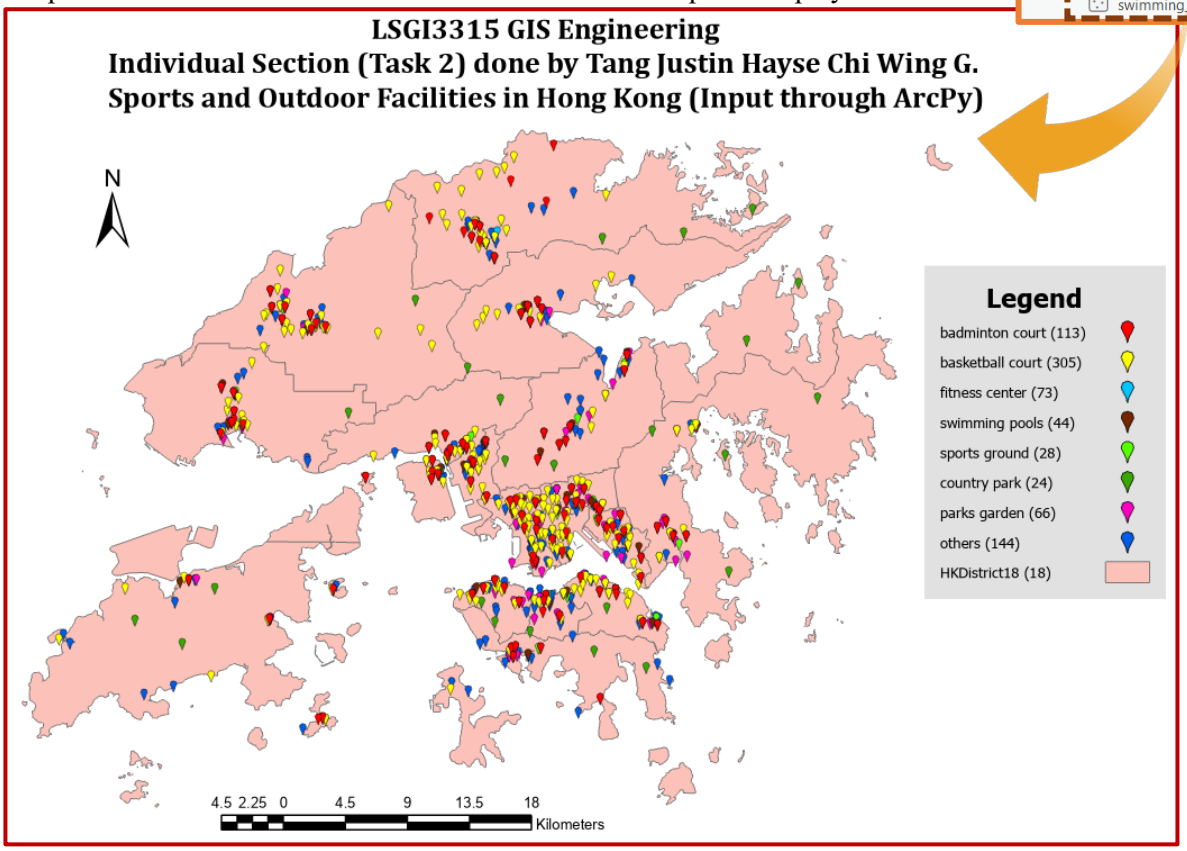
```
LSGI3315_20016345D_Gp6_Task2
C:\Users\justi\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone-1\python.exe C:/Users/justi/Downloads/Group6_Raw/LSGI3315_20016345D_Gp6_Task2.py
***** Mission 1 (Required): Importing the .csv files of Sport and Outdoor Facilities into ArcGIS Pro *****
The Rows from [ Badminton_court.csv ] have been stored in the List successfully!
Creating a Feature Class to input point data (Sports and Outdoor Facilities)...
An empty feature class with the required fields is created! Import the point data into Feature Class now...
The Point Data has been inputted into the Feature Class successfully!

The Rows from [ Basketball_court.csv ] have been stored in the List successfully!
Creating a Feature Class to input point data (Sports and Outdoor Facilities)...
An empty feature class with the required fields is created! Import the point data into Feature Class now...
The Point Data has been inputted into the Feature Class successfully!

The Rows from [ country_parks.csv ] have been stored in the List successfully!
Creating a Feature Class to input point data (Sports and Outdoor Facilities)...
An empty feature class with the required fields is created! Import the point data into Feature Class now...
The Point Data has been inputted into the Feature Class successfully!
```

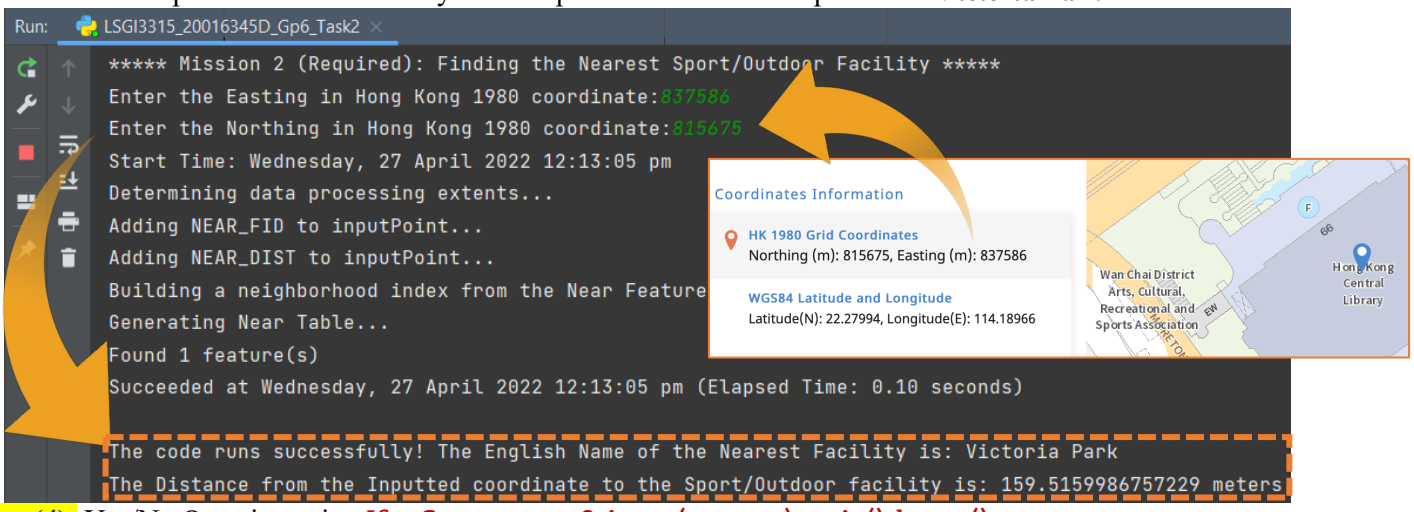


The inputted point data can be visualized in ArcGIS Pro and the output is displayed as follows.



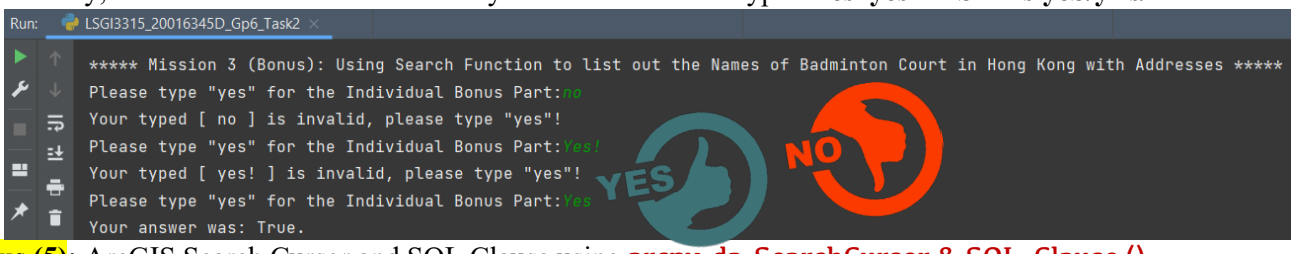
Core (3): Find the Nearest Sports/Outdoor Facility using arcpy.analysis.Near( ) & arcpy.GetMessages()

This function is to take a pair of coordinate in HK1980 grid coordinate system, to return the English name of the nearest facility. The distance between inputted coordinate and the nearest sports/outdoor facility is also generated. User can simply type the northing and easting in HK1980 coordinate to find the closest sports/outdoor facility. For example, after studying at ***The Hong Kong Central Library***, an user wants to exercise to relieve stress, then he/she can use the code to find the closest sports or outdoor facility. The output shows the nearest place is in ***Victoria Park*** and it is reasonable.



Bonus (4): Yes/No Question using If…Statements & input(prompt).strip().lower()

This additional function is to ask the user whether would like to try the Bonus section. If the user inputs "yes", the user will go to the Bonus Section. Otherwise, there will have a loop until the user type correctly. To make the function more user-friendly, the code removes case sensitivity and enables user to type "**Yes/ yes/YES/YEs/yeS/yEs/**" in the window.



Bonus (5): ArcGIS Search Cursor and SQL Clause using arcpy.da.SearchCursor & SQL_Clause ()

SearchCursor is a read-only access to return the information from a feature class. This function is important especially when the user encounter an enormous spatial data. In order to discover the specific sports/ outdoor facilities, it can adopt SQL clause to help us search the data effectively. For instance, an end-user would like to know the badminton courts with the address in Hong Kong, it can use this additional function to find out the corresponding information speedily.

```
136 ▶ if __name__ == "__main__":
137       print("***** Mission 1 (Required): Importing the .csv files of Sport and Outdoor Facilities into ArcGIS Pro *****")
138       f = facility()
139       print("\n")
140       print("***** Mission 2 (Required): Finding the Nearest Sport/Outdoor Facility *****")
141       f.Nearest_Facility()
142       print("\n")
143       print("***** Mission 3 (Bonus): Using Search Function to list out the Names of Badminton Court in Hong Kong with Addresses *****")
144       f.run_yesno()
145       Search_facility = 'Badminton_court.csv'
146       Search_facility_address = ['Facility Name', 'District', 'Address']
147       f.ArcPy_Search_Cursor(Search_facility, Search_facility_address)
148       print("This is the END of LSGI3315 Group 6's Individual Part: Tang Justin Hayse Chi Wing G. (20016345D)")
149       print("Thank you for running the Python Script!")
```

```
Run: LSGI3315_20016345D_Gp6_Task2
***** Search Result: Name, District and Address:
Aberdeen Sports Centre
SOUTHERN
6/F Aberdeen Municipal Services Building, 203 Aberdeen Main Road, Aberdeen
**********
***** Search Result: Name, District and Address:
Ap Lei Chau Sports Centre
SOUTHERN
8 Hung Shing Street, Apleichau
**********
```

The output shows the badminton courts with address in the Southern District in Hong Kong.

--- This is the END of LSGI3315 Group 6 Individual Section (Tang Justin Hayse Chi Wing G). Thank you. ---

Appendix: Python Code for the Task 2 Individual Section

Mission 1 (Core): Using __ini__(self) to define instance variables to record the attributes of sports facilities

```
class facility:  # Create a Class named "facility"
    def __init__(self):  # Initialization: the Class "facility"
        # List out all csv files of Sports and Outdoor Facilities and define
        csv = ['Badminton_court.csv', 'Basketball_court.csv', 'country_parks.csv', 'fitness_center.csv',
               'other_recreation_sports_facilities.csv', 'parks_gardens.csv', 'sports_grounds.csv', 'swimming_pools.csv']
        for csv_files in csv:
            # Input all csv files
            self.fileInput = csv_files
            # Read the DataFrame
            data_frame = pd.read_csv(self.fileInput, engine='python')
            # Subset the Dataframe and record the attributes of sports and outdoor facilities
            self.df = data_frame[
                ["GMID", "Dataset", "Facility Name", "Address", "District", "Northing", "Easting", "Latitude", "Longitude", "Last Update "]]
            # Create an empty list to store the values from CSV
            self.input_list = []
            # Create a For-loop to scan the rows
            for index, row in data_frame.iterrows():
                # Converting the Pandas Series into list
                data_list = row.tolist()
                # Appending the list into "data_list"
                self.input_list.append(data_list)
            print("The Rows from [", str(csv_files), "] have been stored in the List successfully!")
            self.CSV_To_FeatureClass()
```

Mission 2 (Core): Convert the information into a feature class, to store data in ArcGIS Geodatabase

```
    def CSV_To_FeatureClass(self):   # This Function is to convert the input files into Geodatabase Feature Class
        print("Creating a Feature Class to input point data (Sports and Outdoor Facilities)... ")
        # Create an Empty Feature Class through ArcPy
        self.fc = arcpy.CreateFeatureclass_management("Gp6_20016345D_Task2_v2.gdb", self.fileInput.split(".")[0],
                                                      "POINT", "", "DISABLED", "DISABLED",
                                                      arcpy.SpatialReference(4326))
        # The column in ArcGIS Geodatabase
        column_list = ["GMID", "Dataset", "Facility Name", "Address", "District", "Last Update "]
        # Create a For-loop to add fields into shapefile
        for field in column_list:
            if "Last Update" in field:
                name = field
                field_type = "DATE"
            else:
                name = field
                field_type = "TEXT"
            arcpy.AddField_management(self.fc, name, field_type, field_length=200)
        print("An empty feature class with the required fields is created! Import the point data into Feature Class now...")
        # Fields list of Feature class to insert the data
        featureclass_field_list = ["SHAPE", "GMID", "Dataset", "Facility_Name", "Address", "District", "Last_Update"]
        # Create an ArcPy Data Access cursor to insert values in the feature class
        cursor = arcpy.da.InsertCursor(self.fc, featureclass_field_list)
        # Insert the Point Data into the Feature class
        # Create a For-Loop via Pandas DataFrame
        for index, row in self.df.iterrows():
            # Fetch the preferred information
            shape = arcpy.Point(float(row["Longitude"]), float(row["Latitude"]))
            GMID = row["GMID"]
            dataset = row["Dataset"]
            facility_name = row["Facility Name"]
            address = row["Address"]
            district = row["District"]
            last_update = row["Last Update "]
            # Feature Class Attribute List to insert values
            List_InsertRow = [shape, GMID, dataset, facility_name, address, district, last_update]
            # Inserting new row using the Data Access Cursor
            cursor.insertRow(List_InsertRow)
        print("The Point Data has been inputted into the Feature Class successfully! \n")
```

Mission 3 (Core): Find the nearest sports/outdoor facility

```
    def Nearest_Facility(self):
        arcpy.env.workspace = r"C:\Users\justi\Downloads\LSGI3315_Individual_Gp6_20016345D\Gp6_20016345D_Task2_v2.gdb"
        csv = ['Badminton_court', 'Basketball_court', 'country_parks', 'fitness_center',
               'other_recreation_sports_facilities', 'parks_gardens', 'sports_grounds', 'swimming_pools']
        Near = []   # Create an empty list to store the point data for this function
        arcpy.management.Merge(csv, 'merged_fac')   # Merge all point data
        easting = input("Enter the Easting in Hong Kong 1980 coordinate:")  # Input Easting
        northing = input("Enter the Northing in Hong Kong 1980 coordinate:")  # Input Northing
        point = arcpy.Point(float(easting), float(northing))  # Define the Data Type of the Input
        ptGeometry = arcpy.PointGeometry(point, arcpy.SpatialReference(2326))  # Define HK1980 Coordinate System (2326)
        arcpy.CopyFeatures_management(ptGeometry, "inputPoint")
        arcpy.analysis.Near("inputPoint", 'merged_fac', '', '', '', '')
        print(arcpy.GetMessages(), '\n')
        field = ["NEAR_FID", "NEAR_DIST"]
        cursor = (arcpy.da.SearchCursor("inputPoint", field))
        for row in cursor:  # Using for-loop to find the nearest facilities
            Near.append(row[0])
            Near.append(row[1])
        del cursor
        ID = int(Near[0]) - 1
        distance = float(Near[1])
        cursor = arcpy.da.SearchCursor("merged_fac", 'Facility_Name')
        for index, row in enumerate(cursor):
            if index == ID:  # Print the English Name of the nearest facility
                print('The code runs successfully! The English Name of the Nearest Facility is: ' + row[0])
        del cursor
        print('The Distance from the Inputted coordinate to the Sport/Outdoor facility is: ' + str(distance) + ' meters')
```

Mission 4 (Bonus): Yes/No Question + ArcPy Search Cursor and SQL Clause

```
    def yesno(self):  # Bonus Part 1: Create a Yes/No Function asking the end-user whether want to try the Bonus Part
        prompt = f'Please type "yes" for the Individual Bonus Part:'
        answer = input(prompt).strip().lower()  # Eliminate the case sensitivity (Allow: YES/Yes/yes/YEs/yeS/yEs/YeS/yES)
        if answer not in ['yes']:
            print(f'Your typed [ {answer} ] is invalid, please type "yes"!')  # print out the wrong answer
            return self.yesno()  # ask the user again if he/she does not type 'yes'
        if answer == 'yes':
            return True  # Go to the Bonus Part
        return False

    def run_yesno(self):
        answer = self.yesno()  # Call the 'yesno' Function
        print(f'Your answer was: {answer}.')  # print out the answer that the end-user typed

    def ArcPy_Search_Cursor(self, fc, address):  # Develop a Search Cursor Function

        with arcpy.da.SearchCursor(fc, address) as user_search_cursor:
            for i in user_search_cursor:
                print("***** Search Result: Name, District and Address:")
                for j in range(len(address)):
                    # display the info in python interface for user
                    fc, address, sql_clause = (None, i, j)  # Set the SQL Clause
                    print((i[j]))  # Print out the result that end-users would like to search
                print("**********************************************************************************************")
```

Main Function of the Python Program

```
if __name__ == "__main__":
    print("***** Mission 1 (Required): Importing the .csv files of Sport and Outdoor Facilities into ArcGIS Pro *****")
    f = facility()
    print("\n")
    print("***** Mission 2 (Required): Finding the Nearest Sport/Outdoor Facility *****")
    f.Nearest_Facility()
    print("\n")
    print("***** Mission 3 (Bonus): Using Search Function to list out the Names of Badminton Court in Hong Kong with Addresses *****")
    f.run_yesno()
    Search_facility = 'Badminton_court.csv'
    Search_facility_address = ['Facility Name', 'District', 'Address']
    f.ArcPy_Search_Cursor(Search_facility, Search_facility_address)
    print("This is the END of LSGI3315 Group 6's Individual Part: Tang Justin Hayse Chi Wing G. (20016345D)")
    print("Thank you for running the Python Script!")
```