

ARTIFICIAL INTELLIGENCE

Lab Manual

[Fall/ Spring 20\_\_]

|  |  |
| --- | --- |
| Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Student Id: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
|  |  |

|  |  |
| --- | --- |
| Prepared By: *Dr. Noman Islam* |  |
| Instructor: *Dr. Noman Islam* |  |

**LIST OF EXPERIMENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Date** | **Experiment** |  |
| **1** | \_\_/\_\_/\_\_ | To setup the environment and familiarize with Python |  |
| **2** | \_\_/\_\_/\_\_ | To study and implement algorithms in Python |  |
| **3** | \_\_/\_\_/\_\_ | To study and implement Graph search algorithms in Python |  |
| **4** | \_\_/\_\_/\_\_ | To study and understand numpy library |  |
| **5** | \_\_/\_\_/\_\_ | To study and implement panda library |  |
| **6** | \_\_/\_\_/\_\_ | To study and implement matplotlib library |  |
| **7** | \_\_/\_\_/\_\_ | To setup environment and familiarize with Tensorflow |  |
| **8** | \_\_/\_\_/\_\_ | To study and implement linear regression in Tensorflow |  |
| **9** | \_\_/\_\_/\_\_ | To study and implement classification algorithms in Tensorflow |  |
| **10** | \_\_/\_\_/\_\_ | To study and implement clustering algorithms in Tensorflow |  |
| **11** | \_\_/\_\_/\_\_ | To study and implement Artificial Neural Network (ANN) in Tensorflow |  |
| **12** | \_\_/\_\_/\_\_ | To study and implement Convolutional Neural Network (CNN) in Tensorflow |  |
| **13** | \_\_/\_\_/\_\_ | To study and implement LSTM in Tensorflow |  |
| **14** | \_\_/\_\_/\_\_ | To study and implement Hidden Markov Model in Tensorflow |  |

**Lab 1: To setup the environment and familiarize with Python**

The objective of this lab is to set up the Python environment and get some familiarity with the language.

To set up the environment, follow the steps below:

1. Download and install Anaconda. Anaconda is the leading open data science platform powered by Python
2. Download and install PyCharm. PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python language.

**Lab Tasks:**

1. Write a small program in Python to print your CV.
2. Write a program that takes the month (1…12) as input. Print whether the season is summer, winter, spring or autumn depending upon the input month.
3. To determine whether a year is a leap year, follow these steps:
   1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
   2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
   3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
   4. The year is a leap year (it has 366 days).
   5. The year is not a leap year (it has 365 days).

Write a program to input an year as integer. Using if…else, determines whether the input is a leap year or not.

1. Write a program that takes a line as input and finds the number of letters and digits in the input
2. Write a program that takes a sentence as input. Compute the frequency of each words and prints them.

**Lab 2: To study and implement basic algorithms in Python**

In this lab, we will familiarize ourselves with functions, classes and other advanced constructs of python.

**Lab Tasks:**

1. Write a program to generate a dictionary that contains (i,sqrt(i)), where *i* is an integer between 1 and n. *n* is a number input by the user.
2. Write a simple calculator program using functions add, sub, mul and div. The program should accepts two numbers and an operator and calls the corresponding function to perform the operation.
3. Write a function that generates a list with values that are square of number between 1 and 20.
4. Define a class named Shape with static method printType. Define methods draw() and area(). Now define two class Rectangle and Triangle. Rectangle has two attributes length and width. The Triangle class has attributes a,b and c. Override the two methods of shape class. Demonstrate the functionality of class by creating its objects.
5. Using recursion, write a program to calculate the reverse of a string.

**Lab 3: To study and implement Graph search algorithms in Python**

In this lab, we are going to implement searching algorithms in Python. There are two popular searching algorithms i.e. Depth First Search (Fig. 3a) and Breadth First Search (Fig 3b).

|  |
| --- |
| procedure DFS(G,v):  label v as discovered  for all edges from v to w in G.adjacentEdges(v) do  if vertex w is not labeled as discovered then  recursively call DFS(G,w) |
| **3a:** Pseudo-code for Breadth First Search |
| dfs(vertex v)  {  visit(v);  for each neighbor w of v  if w is unvisited  {  dfs(w);  add edge vw to tree T  }  } |
| **3b:** Pseudo-code for Depth First Search |

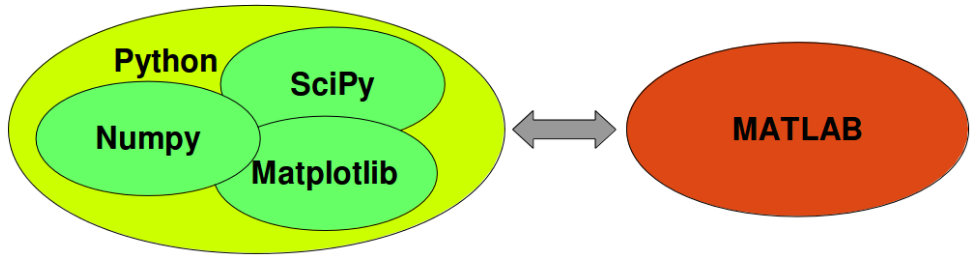
**Fig 3:** Pseudo-code for Graph Searching algorithms

**Lab Task:**

1. Provide the implementation of DFS and BFS algorithms in Python.

**Lab 4: To study and understand numpy library**

In this lab, we are going to explore numpy. NumPy is an acronym for "Numeric Python" or "Numerical Python". It is an open source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines.



Open the Python Notebook and perform the tasks.