Mathematical Programming: Course Introduction

March 7th 2023

Instructor: Joon Soo Yoo

Table of Contents

Course Introduction

A little bit of me

- Major
  - BS: Math Major (KU)
  - MS: Information Security (KU)
  - PhD: Information Security (KU) (currently, PhD candidate)
- Interest / Specialization
  - Homomorphic Encryption
  - Private Machine Learning

Contact

- If you have a question, contact me through e-mail:
    a. e-mail: sandiegojs@korea.ac.kr

Course Objective

1. Linear Algebra (before Mid-term)
2. Machine Learning (after Mid-term)

- The main goal of this course is “Programming” in Python.

Course Target: Math/Fintech Major Students

1. Current Enrollment: 50 students
    a. Math Major Students: 27 / FinTech Major Students: 17
    b. Others: 6 (Security / CS / Economy)
2. Target: Math/Fintech Major Students (1st Semester of Sophomore)
    a. Basic Programming
        i. Python Basics / Using Numpy, Pandas libraries and Popular ML libraries

Evaluation

1. Assignment (30%)
    a. Every Week - Programming Assignment
2. Two Quiz (20%)
    a. Programming or Math Problems
    b. First Quiz: Before Mid-term / Second Quiz: After Mid-term
3. Final (50%)

Course Style

1. Before Mid-Term
    a. Implement Linear Algebra (LA) Representative Algorithms
    b. Examples: Gaussian Elimination, Gram-Schmidt
2. After Mid-Term
    a. Introduction to Machine Learning (ML)
    b. Examples: Regression, Clustering

Course Survey

Quick Survey - Course Direction

- Coding Experience
- Familiarity with Linear Algebra

How to Program

Common Mistakes for Math Major

1. Look for reference in BOOK
2. Need to Know All Syntax (Grammar)

How to Program - from Math Major Perspective

1. ChatGPT
2. Googling
3. Youtube
4. Free courses available!
    a. Coursera / edx / Fast Campus

Demonstration!

- ChatGPT / Googling / Youtube

Program Learning

1. Don’t try to memorize!
    a. Just get used to it!
    b. Final term - open Google
2. Solve programming problem or task (Reference Google!)
3. Don’t get too stressed about compiler error

Program Example (Demonstration)

1. Create a program that computes two square matrices of size 2 using numpy library

What to Prepare for Class

Jupyter Notebook (Programming Tool)

1. Jupyter Notebook is Interactive
    a. Easy for Beginners
2. Download Jupyter Notebook using ChatGPT
    a. Ask question: “How can I download jupyter notebook. I am using Windows”
    b. or https://zidarn87.tistory.com/314

Programming Assignment Example

Example of Problem and Report

1. Given an array of integers, write a Python program that finds the sum of all the even numbers in the array.

In [38]:

```python
# define an array of integers
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# initialize a variable to hold the sum of even numbers
sum_of_even = 0

# loop through the array
for num in numbers:
    # check if the number is even
    if num % 2 == 0:
        # add the even number to the sum
        sum_of_even += num

# print the sum of even numbers
print("The sum of even numbers in the array is:", sum_of_even)
```

```
The sum of even numbers in the array is: 30
```

Basic Programming Concepts

1. What is programming?

- Programming is giving a set of instructions to a computer to execute.
- Analogy: Recipe (You) and Cooking (Computer)

2. How many programming languages are there? A lot...

1. Python
2. Java
3. JavaScript
4. C++
5. C#
6. PHP
7. Ruby
8. Swift
9. Objective-C
10. Kotlin
11. Go
12. Rust
13. TypeScript
14. SQL
15. HTML/CSS (markup languages)

1. MATLAB
2. Bash
3. Assembly language
4. Pascal
5. Dart
6. Erlang
7. Groovy
8. Haskell
9. Julia
10. Lisp
11. Prolog
12. Scheme
13. Tcl
14. Shell
15. Visual Basic

1. C
2. Ada
3. COBOL
4. Fortran
5. Smalltalk
6. ActionScript
7. Dart
8. Lua
9. Objective-C
10. Perl
11. PowerShell
12. Ruby on Rails (a framework built on Ruby)
13. Scala
14. TypeScript
15. Kotlin

3. Then, why are there so many?

1) Different Specialization
    a) Python - Data Analysis, AI
    b) HTML/CSS - Create Web and Styling
    c) Javascript - Dynamic Functionality
2) Evolving Technology

Common Structure of Programming Language

1) Variable Declaration
2) Basic Syntax
3) Data Type and Structures
4) Control Flow Structures
5) Functional Programming

Variables / Data Types

In [5]:
```
# Declare some variables
name = "Alice"
age = 25
is_student = True
grades = [90, 85, 95, 80]
```

In [9]:
```
# Print out the variable values
print("Name:", name)
print("Age:", age)
print("Is Student:", is_student)
print("Grades:", grades)
```

```
Name: Alice
Age: 25
Is Student: True
Grades: [90, 85, 95, 80]
```

Syntax

In [10]:
```
# This code contains a syntax error
x = 5
y = 10
if x > y
    print("x is greater than y")
else:
    print("x is less than or equal to y")
```

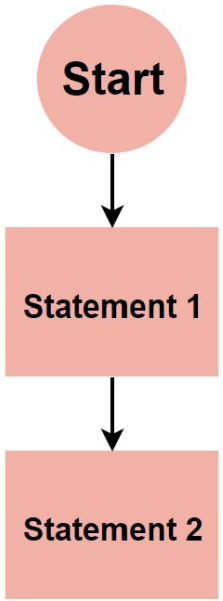
```
  File "<ipython-input-10-886adb7e3710>", line 4
    if x > y
           ^
SyntaxError: invalid syntax
```

In [11]:
```
# This code has been fixed
x = 5
y = 10
if x > y:
    print("x is greater than y")
else:
    print("x is less than or equal to y")
```

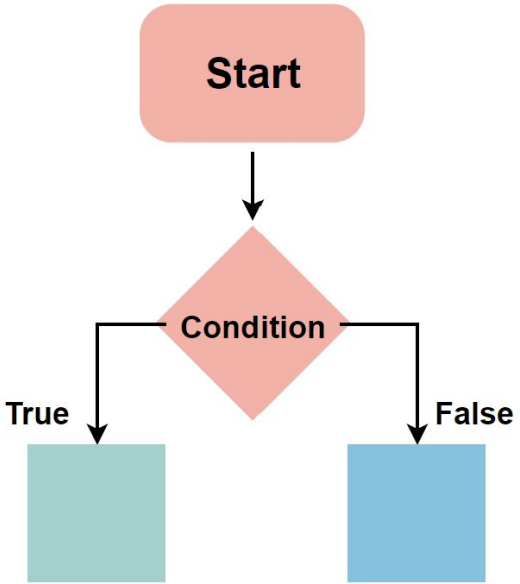
```
x is less than or equal to y
```

Control Flow Structure

1) Sequential



2) Conditional



Control Flow Structure (Example)

1) Sequential

3.1. Sequential

In [12]:
```
# Sequential programming example
x = 5
y = 10
z = x + y
print("The sum of x and y is:", z)
```

```
The sum of x and y is: 15
```

- Sequential programming refers to a programming style where statements are executed one after the other in the order that they are written.
- In this example, each line of code is executed in sequence, starting with the declaration of x, followed by y, then z, and finally the print() statement.

Control Flow Structure (Example)

2) Conditional

3.2. Conditional

In [13]:
```python
# If-else statement example
x = -2
if x > 0:
    print("x is positive")
else:
    print("x is non-positive")
```

```
x is non-positive
```

Loop

In [15]:
```
# For loop example
fruits = ["apple", "banana", "cherry", "melon", "kiwi"]
for fruit in fruits:
    print(fruit)
```

```
apple
banana
cherry
melon
kiwi
```

- The above code is same as the following:

In [16]:
```
# Same Code
print("apple")
print("banana")
print("cherry")
print("melon")
print("kiwi")
```

```
apple
banana
cherry
melon
kiwi
```

Function (Example)

```
Hello, Joon!
Nice to meet you, how are you doing?
Hello, Soo!
Nice to meet you, how are you doing?
Hello, Yoo!
Nice to meet you, how are you doing?
```

Function (Example)

In [25]:
```
# Function Declaration
def greet(name):
    print("Hello, " + name + "!")
    print("Nice to meet you, how are you doing?")
```

In [24]:
```
# Function call
greet("Joon")
greet("Soo")
greet("Yoo")
```

```
Hello, Joon!
Nice to meet you, how are you doing?
Hello, Soo!
Nice to meet you, how are you doing?
Hello, Yoo!
Nice to meet you, how are you doing?
```

EOP