# CIS-30A Course Project

This project will assess your understanding of Python programming concepts that were covered in the course. Review project options and select an option for the project. **Select an option and submit each part (1, 2, 3) of the project in Canvas**.

## Project Options

Select 1 of the listed options.

### Option 1: Business Python Program

Write a Python program **to place order and set appointment for delivery** **of goods or services from a business of your choice** (restaurant, grocery, mobile pet spa, mobile car detailer, home cleaning, home repair/improvement, mobile car repair, etc.…) .

\*Note: [Tkcalendar](https://pypi.org/project/tkcalendar/1.1.5/) can be used in this option.

* + The program should prompt the user to select products or services and appointment or delivery date and time from the available options in the 1-year span.
  + The program should display the user selection on screen.
  + The program should output the order summary and appointment in a text file.
  + The program should contain the following components:
    - Include comments throughout the program (10 points)
    - Use variables and list to store and access data. You can use tuple or dictionary in place of a list (20 points)
    - Use string object to display and control text output. (10 points)
    - Define 2 or more functions and use function calls to execute tasks in the program. (20 points)
    - Implement loop (for or while or both) (10 points)
    - Include conditional statement (if or if-else or if-elif-else) (10 points)
    - Use a non-built-in module, custom module. (20 points)
    - Contains at least 2 classes and 1 sub-class (20 points)
    - Includes 1 or more objects and 1 or more methods in each class. (20 points)
    - Implement error detection using Python built-in exceptions. (10 points)
    - Implement file operations and file output. (20 points)
    - Integrate UI (optional): Bonus 30 points

## Option 2: Personal Finance Tracking Program

Write a Python program for personal finance tracking. The program should allow the user to create a budget, input monthly or weekly income, expenses and savings.

\*Optional libraries:

* [numpy](http://www.numpy.org/) – NumPy is the fundamental package for scientific computing with Python. It is a first-rate library for numerical programming and is widely used in academia, finance, and industry. NumPy specializes in basic array operations.
* [scipy](https://www.scipy.org/) – SciPy supplements the popular Numeric module, Numpy. It is a Python-based ecosystem of open-source software for mathematics, science, and engineering. It is also used intensively for scientific and financial computation based on Python
* [pandas](http://pandas.pydata.org/) – The pandas library provides high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Pandas focus is on the fundamental data types and their methods, leaving other packages to add more sophisticated statistical functionality
* [quantdsl](https://github.com/johnbywater/quantdsl) – Quand DSL is domain specific language for quantitative analytics in finance and trading. *Quant DSL*is a functional programming language for modeling derivative instruments.
* [statistics](https://docs.python.org/3/library/statistics.html) – This is a built-in Python library for all basic statistical calculations.
  + The program should prompt the user to input periodic (monthly or weekly) budget, income, expenses and saving goals.
  + The program should allow the user to modify and update income, expenses and savings.
  + The program should output the summary of income, expenses, and savings in text file.
  + The program should contain the following technical components:
    - Include comments throughout the program (10 points)
    - Use variables and list to store and access data. You can use tuple or dictionary in place of a list (20 points)
    - Use string object to display and control text output. (10 points)
    - Define 2 or more functions and use function calls to execute tasks in the program. (30 points)
    - Implement loop (for or while or both) (20 points)
    - Include conditional statement (if or if-else or if-elif-else) (10 points)
    - Use a non-built-in module, custom module. (20 points)
    - Contains at least 2 classes and 1 sub-class (30 points)
    - Includes 1 or more objects and 1 or more methods in each class. (20 points)
    - Implement error detection using Python built-in exceptions. (10 points)
    - Implement file operations and file output. (20 points)
    - Integrate UI (optional): Bonus 30 points

### Option 3: Key Logger Program

Keylogging is, as the name suggests, the action of recording the keys entered on the keyboard on a computer. Keyloggers can be used for both legitimate and malicious purposes.

Write a keylogger program that records the user input. You can use the [Pynput library](https://pynput.readthedocs.io/en/latest/index.html) to control and monitor input devices such as mouse and keyboard. Additionally, you can use [Listener API](https://python-can.readthedocs.io/en/master/listeners.html) and [Python-can](https://python-can.readthedocs.io/en/master/index.html) to read data buffer and create log files of data that has been input by the users.

* The program should entice the user to enter information.
* The program should record keyboard input and save input information to a text file.
* The program should output a text file that contains input information from the user.
* The program should contain the following components:
  + Include comments throughout the program (10 points)
  + Use variables and list to store and access data. You can use tuple or dictionary in place of a list (20 points)
  + Use string object to display and control text output. (10 points)
  + Define 2 or more functions and use function calls to execute tasks in the program. (20 points)
  + Implement loop (for or while or both) (10 points)
  + Include conditional statement (if or if-else or if-elif-else) (10 points)
  + Use a non-built-in module, custom module. (20 points)
  + Contains at least 2 classes and 1 sub-class (20 points)
  + Includes 1 or more objects and 1 or more methods in each class. (20 points)
  + Implement error detection using Python built-in exceptions. (10 points)
  + Implement file operations and file output. (20 points)
  + Integrate UI (optional): Bonus 30 points

## Part 1 (100 points) – Due in week 6

1. Provide a brief summary describing your **program goals, functionality and target audience**. Include **strength and weakness of the program and outlook for program future improvement**. (1 page double-spaced. Submit a document that contains program description. (60 points)
2. Write pseudocode for the program. Be sure to include conditional branching, input, output and all program operations. See [How To Write Pseudocode](https://blog.usejournal.com/how-to-write-pseudocode-a-beginners-guide-29956242698). Submit a document that contains pseudocode. (40 points)

## Part 2 (200 points) – Due in week 6

Write the program that meets the requirements the above option (option 1, 2 or 3). Test the limitations of the program. Submit Python files in zipped folder. (200 points)

## Part 3 (50 points) – Due in week 6

1. Complete a survey for Project Evaluation. If you are working with a team member, rate YOURSELF and TEAMMATE’S performance. \*Evaluation information will be used in grading Part 3 and Part 2 of the project. Submit a screen capture of the Survey Submitted Page (25 points)
2. Use [Google Sites](https://support.google.com/sites/answer/6372878?hl=en) to create a portfolio. Design your portfolio and add components of this project (**documentation, pseudocode, code/GitHub URL**). **Publish the portfolio** and **submit URL of your portfolio**. (25 points) \*Note: In Canvas, click URL tab and paste in URL.