## Introduction to Computer Programming

## Exercises - Week 10: Classes and Modules

1. **Review Tuesday's lecture.**

   - Study the slides from this week, type and test a couple of the programs. Try to understand the syntax, and what the code is doing.

2. **Defining Classes**
   The goal of this exercise is to write a program that creates a shopping list and then prints out all the items and the total price. There is a similar example in the slides if you need help.

   - Write a class called `Item` that has a constructor `__init__(self)` that prints "This is an item". From your main program, create an object of class Item called Apple. Run the program.
   - Change the constructor to include 3 additional arguments: `__init__(self, Description, Number, UnitPrice)`.
     From your main program, create an object of class Item called Apple with parameters "Apple", 1, and 0.5. Call `print(Apple.Description,Apple.Number,Apple.UnitPrice)` to print out the information.
   - Include a function in your class called `PrintItemInfo(self)` that prints all the information about the item. Call `Apple.PrintItemInfo()` from the main program.
   - Create a list called `ShoppingList`, add the Apple, and 2 other items to the list. Write a loop in the main program to go over all items in `ShoppingList`, print out the information and sum the total price (the price for one item is `Number*UnitPrice`). Print out the total at the end.

3. **Using Modules**

   - **Turtle**
     - From the Python Interpreter, type `import turtle`. This should open a small window.
     - Have a look at the turtle documentation or type `help(turtle)`: `https://docs.python.org/3.0/library/turtle.html`
     - Test the functions `forward()`, `backward()`, `right()`, `left()`, `goto()`, `setheading()`, `circle()`.
     - Experiment with the pen functions `penup()`, `pendown()`, `pensize()`.
     - Write a program (in the editor) to make the turtle draw a square. Use `begin_fill()` and `end_fill()` to fill the square.
     - If you have time, experiment with the recursive functions here by copy-pasting them into your editor: `http://interactivepython.org/courselib/static/pythonds/Recursion/graphical.html`. Notice how the functions are called from themselves. Can you make changes to the drawings generated?
   - **TkInter**

- Have a look at the tkinter documentation: `https://docs.python.org/3/library/tkinter.html#module-tkinter`
- Download the code `tkinter_hello_world.py` from blackboard and run it. Study the code.
- Add a third button that prints "Hello Again" when you click it.

4. **Bonus Exercise**
   The goal of this exercise, if you have extra time, is to program a text version of the game Knots and Crosses. This could be an example of a basic pass for your final Python project. Each player, through the Python Interpreter, is asked for their name. The players are then asked, by name, to enter coordinates for their knot or cross. The input is checked and the new board is drawn (use character printouts to show the board). The program then detects the winner at the end of the game, increments scores accordingly, and starts a new game. Think of how you might use a graphical use interface to improve this project.