4COM2005/10 Computational Problem Solving

Programming Exercise

Assignment set by: Stephen Hunt

Weighting: This assignment is worth 30% of the marks for the module

Authorship: This is an **Individual** Assignment. You must work on your own.

Number of hours you are expected to work: It should take about 4 hours to complete

Hand-out date / time: 16:00 hrs BST, Thursday 5 May 2022

Submission deadline: 18:00 hrs BST, Monday 9 May 2022

Target date for returning marked work: Friday 27 May 2022

Due to the nature of the assignment and the proximity of the marking deadline **late submissions cannot be accepted** without a Short-Term Extension agreed before the start of the assignment.

Task 1 (20 marks) is to implement a set of functions that operate on / produce data structures

You are provided with a set of function specifications in the skeleton Python script **`task1.py`**, which contains a series of incomplete function definitions and some instructions and a tester program, along with a test plan for the functions, and a PDF of a Venn diagram for use with function 1

Task 2 (10 marks) is to implement / modify a Python class

You are provided with a set of specifications for the class **coin_game** in the Python script **`task2.py`** along with a program that uses the class and some instructions on what to do

There is no Task 3

Submission Requirements:
Upload a zip archive containing modified copies of the two Python scripts `task1.py` and `task2.py` to Canvas. **Make sure you have inserted your SRN where required by the instructions given in each script.**

Marks will be awarded for program correctness

- Each correctly implemented function / method you provide is worth a different number of marks).
- The functions will be tested by an automated system
    - If your code for a task fails to run due to a syntax error the automated tester will award 0 marks for the whole task
    - If a function returns an incorrect result for one or more the tests the automated tester will award 0 marks for those tests
    - If a function crashes your program when tested with data that it should be able to handle the automated tester will award 0 marks for those tests
- Automated testing will be backed up by human inspection

Module Learning Outcomes assessed (from Definitive Module Document)

On completion of the module successful students will have a knowledge and understanding of

• *how discrete structures, including self-similar structures, may be represented and manipulated in Python..*

In addition, successful students will typically be able to:

• *write Python programs that solve well-specified problems*
• *choose appropriate data structures and make judicious use of recursion and iteration to manipulate them.*

Additional information:

- Regulations governing assessment offences including Plagiarism and Collusion are available from https://www.herts.ac.uk/__data/assets/pdf_file/0007/237625/AS14-Apx3-Academic-Misconduct.pdf (UPR AS14).
- Guidance on avoiding plagiarism can be found here: https://herts.instructure.com/courses/61421/pages/referencing-avoiding-plagiarism?module_item_id=779436