Teaching Outline

Week1	Content
1	Overview – editor tools, assessment scheme, past assignments, VSC quick tour, auto booking demo
2	Live Coding – Largest 3 (basic construct, refactoring, functions, generalization) + Eval Live Coding – Caesar Cipher (functions, loop, layout, controls)
3	Coding Styles – Naming Convention, Layout, Variable Scope, Functions, Comments Assignment 1 (User Interface/Console)
4	Live Coding – Simple board game (number-flipping) – (Scope, Spec, Design, Implementation) Program structure - Part I & Part II (console based)
5	Live Coding – Simple board game (number-flipping) – (Scope, Spec, Design, Implementation) Program structure - Part I & Part II (console based)
6	Turtle Graphics (Building blocks – turtles, pen, draw, motion) Turtle Graphics (Screen, Timer, Click, Events, Mouse Tracking, Canvas)
7	Live Coding – porting number-flipping to GUI-based using Turbo Graphics Assignment 2 (Graphical User Interface)
8	Live Coding – Assignment 1 Review
9	Tools - Debugging, Source Control, Virtual Environment
10	Assignment 3 (Overview, Scope, Spec, Design) – Turbo Graphics
11	Assignment 3 (Program Structure, Data Model & Design) – Turbo Graphics
12	Live Coding – Simple board game (flipping number) – Non-Recursion vs Recursion
13	Others – KNN & KMeans

Assessment Scheme

See BB for Info

Assessment Scheme

Component/Method	% Weight	Remarks
Programming Assignment I	30%	To be Completed Individually
Programming Assignment II	30%	To be Completed Individually
Programming Assignment III	40%	To be Completed Individually

DELIVERABLES

1. Design documentation (AX_School_StudentID_Design.doc/pdf), if required
2. Program source code (AX_School_StudentID_Source.py)

where X is the assignment number (1, 2, and so on), School is abbreviated name of the school you are presently enrolled in (SSE, SME, HSS, FE, LHS, …etc) and StudentID is your 9-digit student ID.

Zip all files above in a single file (A1_School_StudentID.zip) and submit the zip file by due date to the corresponding assignment folder under "Assignment (submission)"

For instances, a SME student with student ID "119010001" and the zip file is named as follows:

- A2_SME_119010001.zip:
  - A2_SME_119010001_Design.doc/pdf
  - A2_SME_119010001_Source.py

5% will be deducted if any files are incorrectly named!!!

For the design document kindly refer to section "Design Documentation" for details.

MARKING CRITERIA

- Coding Styles – overall program structure including layout, comments, white spaces, naming convention, variables, indentation, functions with appropriate parameters and return.
- Design Documentation
- Program Correctness – whether or the program works 100% as per Scope.
- User Interaction – how informative and accurate information is exchanged between your program and the player.
- Readability counts – programs that are well structured and easy-to-follow using functions to breakdown complex problems into smaller cleaner generalized functions are preferred over a function embracing a complex logic with nested conditions and sub-functions! In other words, a design with clean architecture with high readability is the predilection for the course objectives over computational performance.
- KISS approach – Keep It Simple and Straightforward.
- Balance approach – you are not required to come up with a very optimized solution. However, take a balance between readability and efficiency with good use of program constructs.

With Design Documentation:

ITEMS	PERCENTAGE	REMARKS
DESIGN DOC	10%-15%	
CODING STYLES	20%-25%	0% IF PROGRAM DOESN'T RUN
USER INTERFACE	15%-20%	0% IF PROGRAM DOESN'T RUN
FUNCTIONALITY	>40%	REFER TO SCOPE AS PER ASSIGNMENT

Without Design Documentation:

ITEMS	PERCENTAGE	REMARKS
CODING STYLES	20%-25%	0% IF PROGRAM DOESN'T RUN
USER INTERFACE	15%-20%	0% IF PROGRAM DOESN'T RUN
FUNCTIONALITY	>55%	REFER TO SCOPE AS PER ASSIGNMENT

DESIGN DOCUMENTATION

For the design document provide write-up for the following sections:

1. Design
    a. Write up an overview of your thoughts (design) and approach used (logic), including the breakdown of your logic into logical sub-components (functions) and how these sub-components are organized and their key functionality, the data model (the kind of data elements you chose to use to implement your solution such as dictionary, list, tuples, string and so on) and the description of the data model in relation to your sub-components (interaction between sub-components and data model). In summary, your design documentation should have minimum following three sessions:
        i. Overview (Design & Approach)
        ii. Data Model
        iii. Program Structure (Data Model vs Components)
2. Function Specifications
    a. Describe usage of all your own defined functions, including details of parameter(s) and output if any.
3. Output
    a. Show samples of output (including status) from your program as per assignment scope.

LATE SUBMISSION

For late submission, a daily penalty amounting to 10 points will be deducted against the final grade, up to maximum 3 days (30 points).

No further submission will be accepted 3 days after the set deadline.