**CFRS 772: Forensic Artifact Extraction**

**Lab and Homework Project 3**

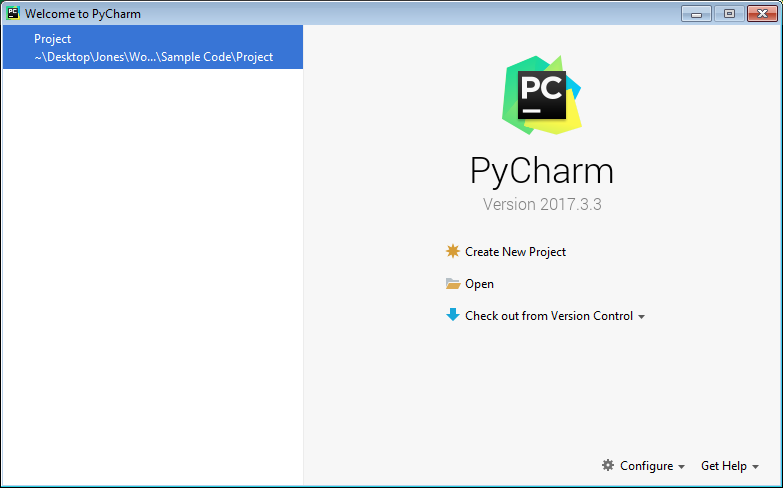
***(In Lab)***

1. Download and install (use defaults) pyCharm community edition
   1. On BlackBoard under Misc folder (zipped)

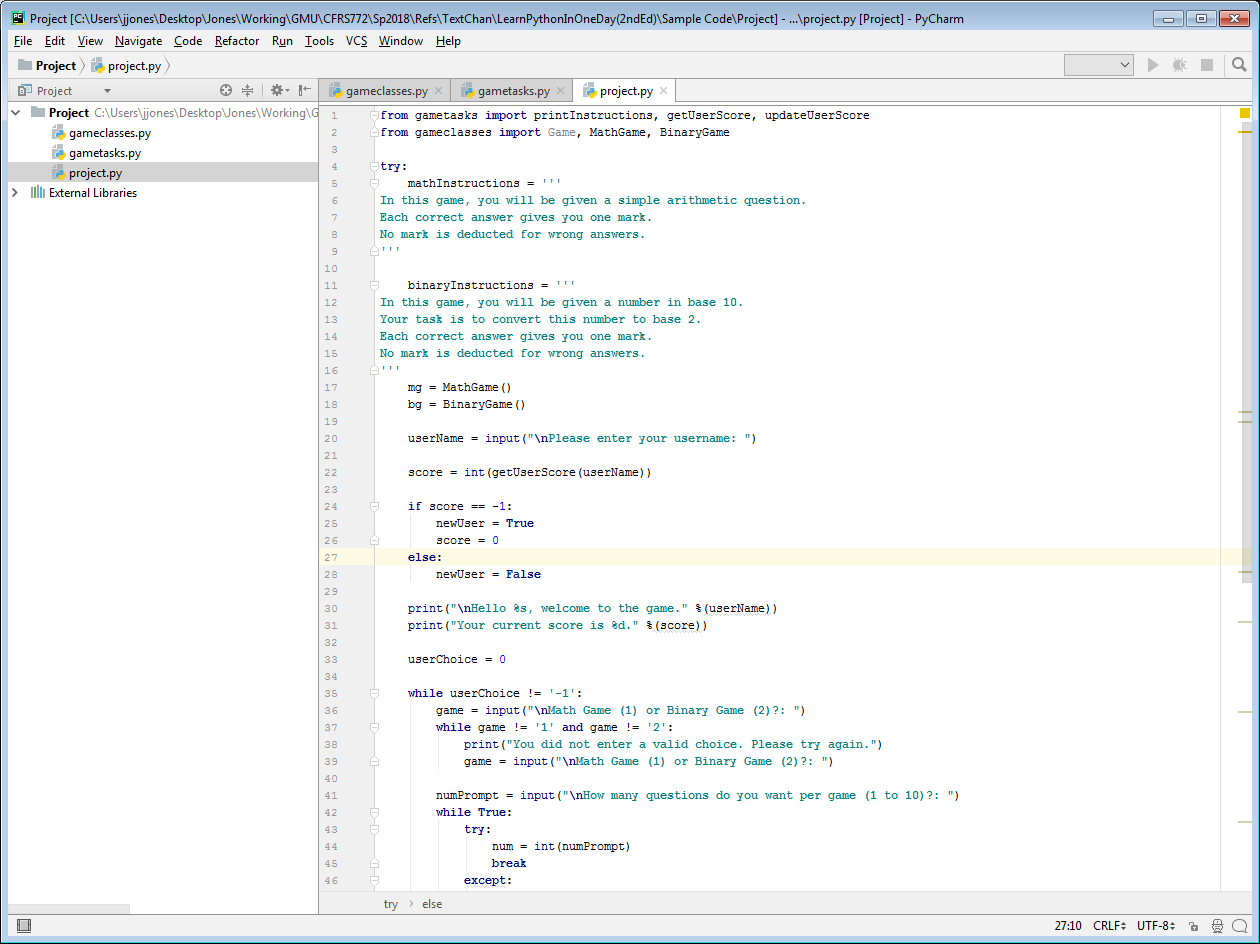
or

https://download.jetbrains.com/python/pycharm-community-2018.3.4.exe

1. Download the Chan text project files (in the Project folder of LearnPythonInOneDay(2ndEd).zip on BlackBoard, Misc folder)
2. Run pyCharm (use defaults at startup)
   1. In the first dialog box, select "Open" and navigate to the folder (Sample Code/Project) containing the Chan text project files (should contain 3 files). Open the **Project** folder (not the individual files).



* 1. When the project opens, expand the Project folder and open each of the source files by double clicking on them. At this point, the PyCharm window should look like this (you might have another item on the left menu and/or notice other minor differences due to version differences):



1. Run gameclasses.py (highlight the gameclasses.py source tab, then select Run: Run from the menu and select gameclasses from the popup.

What was the result? \_\_**Running from PyCharm, the result is: “Process finished with exit code 0”.**\_

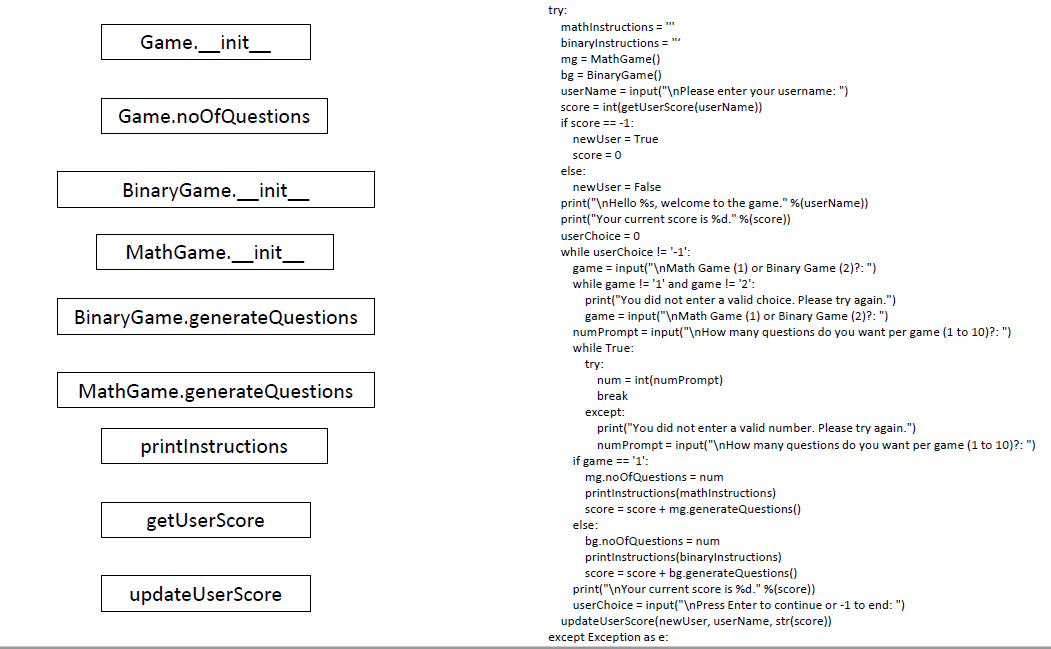
1. Run gametasks.py (highlight the gametasks.py source tab, then select Run: Run from the menu and select gametasks from the popup.

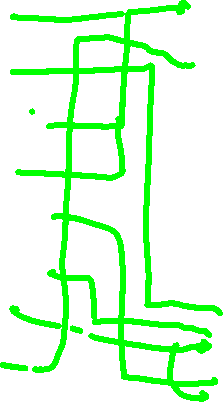
What was the result? \_**Running from PyCharm, the result is: “Process finished with exit code 0”.**\_\_

1. Run project.py (highlight the project.py source tab, then select Run: Run from the menu and select project from the popup. Play the game, and exercise all options (Math Game, Binary Game, valid and invalid inputs, etc.). Open userScores.txt from the Project folder in PyCharm and record the contents here:

\_\_\_\_\_\_\_\_**userScores.txt contains <username>, <score>. In my case: Gully, 4**.\_\_\_\_\_\_\_\_\_\_\_\_

1. Examine the source code and trace the execution of the program at the function level, i.e., use the attachment to draw arrows between the main code and functions, and indicate inputs and outputs on the arrows.
2. Show me your answers to the questions above and program flow diagram at the end of the lab tonight.





***(For HW)***

Use PyCharm to add a Hex game to the project:

* the HexGame will be similar to the BinaryGame:
  + the user will convert a decimal number to hex
  + you're adding a new game, not replacing an existing one
* add HexGame as a class to gameclasses.py
* modify gametasks.py and project.py to accommodate the new game
* make sure the Hex game behaves like the others (instructions, input checking, score recorded, ...)
* test the revised project with all three games

**On BlackBoard, submit (1) a single PDF document with screenshots or copy/paste from the shell showing your revised project code in operation, and (2) your revised project code as three Python files (gameclasses.py, gametasks.py, and project.py)**