Jacob Dimoff

AC3 Documentation

Prog 1700

Table of Contents

[Part 1. 2](#_Toc120804117)

[penniesForPay.py Run Back 2](#_Toc120804118)

[starPattern.py Run Back 3](#_Toc120804119)

[Part 2 4](#_Toc120804120)

[Hierarchy Chart: 4](#_Toc120804121)

[Input Output Chart: 4](#_Toc120804122)

[Function Run Back 6](#_Toc120804123)

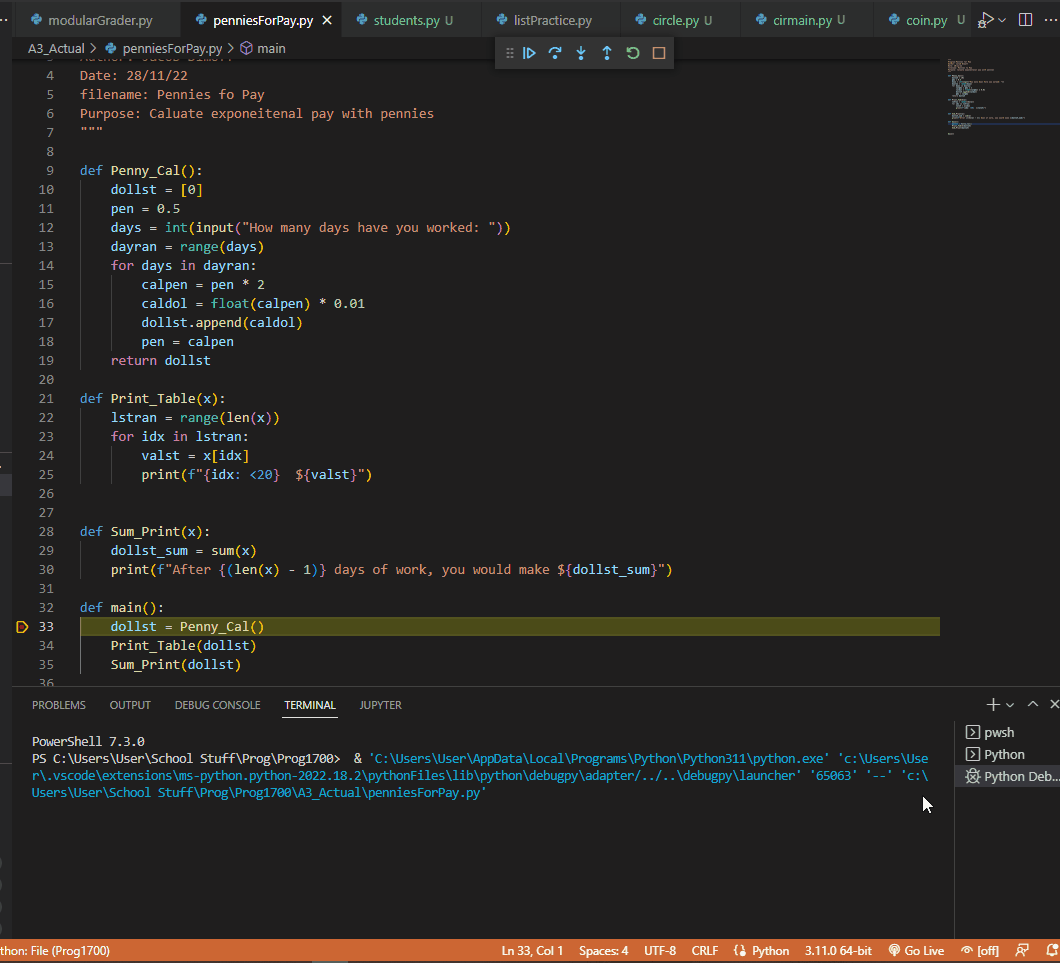
[Part 3: 6](#_Toc120804124)

[Reflection: 6](#_Toc120804125)

[References: 7](#_Toc120804126)

# Part 1.

## penniesForPay.py Run Back



## starPattern.py Run Back

A screenshot of a computer

Description automatically generated with medium confidence

# Part 2

## Hierarchy Chart:

Graphical user interface, diagram, PowerPoint

Description automatically generated

## Input Output Chart:

|  |  |  |
| --- | --- | --- |
| Input displayInstructions(): | Process | Output/Return |
| None | Print Instructional String | Instruction String |

|  |  |  |
| --- | --- | --- |
| Input classOrPerson(): | Process | Output/Return |
| None | Prompt user for input to determine if the average is for a class or person  If the user doesn’t input one or two, prompt for an answer  Else add to control var and exit loop. | cvp variable (1 or 2)  printed(error) |

|  |  |  |
| --- | --- | --- |
| Input validateGrade(x): | Process | Output/Return |
| x = grd | Checks grd.  If  By verifying data type.  Compares data to alpha, “Alpha and numeric” and “Alpha and numeric with punctuation” | Explanation printed string if x = -?  Error message if the unaccepted value  Exit message printed  All these return var = false  If data type accepted var = True  Var = ggbool |

|  |  |  |
| --- | --- | --- |
| Input ranCheck(x): | Process | Output/Return |
| x = grd | Checks grades in a range | Return Var = neobool  True if in range  Else false |

|  |  |  |
| --- | --- | --- |
| Input getGrade(): | Process | Output/Return |
| None | Case grdlst to list, and grd into none  Start while loop that ends when grd = exit  Prompt user for grd input  Validates grd with validateGrade which returns a bool.  If true pass grd as a float and check grd with ranCheck.  If true, append grd into grdlst | Return grdlst |

|  |  |  |
| --- | --- | --- |
| Input calculateAvg(x): | Process | Output/Return |
| x = grdlst | Sum grdlst  The average sum of grdlst / its own length returns grdavg | Return grdavg |

|  |  |  |
| --- | --- | --- |
| Input resulType(x): | Process | Output/Return |
| x = cvp | Checks if cvp = 1 | Return printed state. Differs between 1 and 2 |

|  |  |  |
| --- | --- | --- |
| Input displayResults(x) | Process | Output/Return |
| x = grdavg | Print grdavg | Return printed string of grdavg |

|  |  |  |
| --- | --- | --- |
| Input convarLoop(x): | Process | Output/Return |
| None | Prompt user to determine further use.  If yes, the loop continues  If no, the loop ends  If the answer is not accepted, prompt the user again. | Return convar |

|  |  |  |
| --- | --- | --- |
| Input main(): | Process | Output/Return |
| none | Cast control variable to one  Start while statement controlled by convar  Call for cvp value with classOrPerson.  Call for grdlst with getGrades()  Calculate Average with calculateAvg  Check cvp with resluType  Input grdavg into displayResults  Call convarLoop to determine convar  If the loop ends print the exit statement. | Printed closing statement. |

## Function Run Back

Text

Description automatically generated

# Part 3:

## Reflection:

The three functions in question took some time to figure out. The most interesting challenge I faced for the assignment necessary code was figuring out how to print the triangle star pattern using a matrix. In the modularized function section, I got carried away with the information and validation of the function. Even as I was writing this reflection, I found issues with the code and worked to optimize it. Unfortunately, I couldn’t complete the validation completely, But was able to make what was necessary for the assignment.

# References:

Python Files in Git repo

AC3 Folder

<https://github.com/Jacob-D-000/Prog1700/tree/main/A3>

pennieForPay.py

<https://github.com/Jacob-D-000/Prog1700/blob/main/A3/penniesForPay.py>

Commit # 06b05a284fb744436ba8d548b024b7a0e0d8889e

starPattern.py

<https://github.com/Jacob-D-000/Prog1700/blob/main/A3/starPattern.py>

Commit # 06b05a284fb744436ba8d548b024b7a0e0d8889e

Modular Grader

<https://github.com/Jacob-D-000/Prog1700/blob/main/A3/modularGrader.py>

Commit # 6ea3ca277a1ef75171c2929d183cb60434abc58c