ITEC 1150 Week 2 Chapter 1 part 1 Lab Projects

PROGRAM DEVELOPMENT CYCLE, DATA TYPES, EXPRESSIONS, FORMATTING

Program Development Plan (PDP)

This is a step-by-step process for successfully constructing an application. Follow these steps and repeat until you have successfully completed the program.

- PDP template:

1. Problem definition
2. Analysis of variables & functions
3. Steps in algorithm
4. Code (separate .py file with added comments)
5. Screen snips from testing
6. Keep track of any wanted or needed improvements for a future version

PDP Example: Averaging Grades

Pull up previous code from the lesson exercises, and then open a Word doc as your first program development plan – name: PDP_avg_grade.docx. Step-through the program design model, based on a cycle of continuous improvement. NOTE: this is an EXAMPLE! You need to use your own words!

PDP Step 1. Problem definition: write a program to calculate and display the average quiz score of three students.

Step 2. Analysis of variables & functions:

score1 = hard coded int →45

score2 = hard coded int → 74

score3 = hard coded int → 63

score_avg = (score1 + score2 + score3)/3

→ Rounding may be needed

PDP template -

1. Problem definition
2. Analysis of variables & functions
3. Steps in algorithm
4. Code (separate file w/comments)
5. Screen snips from testing
6. Improvements for future version

PDP Example: Averaging Grades (cont.)

PDP Step 3. Algorithm steps in order (use your own words)

1. Save the first student score under a variable name score1
2. Save the second score as score2
3. Save the third score as score3
4. Save average calculation result as score_avg
5. Display the average in a meaningful sentence. Note: requires rounding and/or formatting.

PDP Example: Averaging Grades (cont.)

```
""" put author name, date, and brief program description here """

# Define variables
score1 = 45
score2 = 74
score3 = 63

# Calculate the average of the 3 scores
average_score = (score1 + score2 + score3) / 3.0 # add rounding!

# Display the average
print ('The average test score is', average_score)
```

PDP Example: Averaging Grades (cont.)

PDP Step 5. Test your program to debug it – your copy should reflect your own actual testing! Example:

On first test, the program output had a long decimal and no period at the end of the sentence:

```
The average test score is 60.666666666666664
```

Fixed by using the round() function against the average score calculation and adding a '.' after the variable in the output string. To get the period to add on properly, I had to switch from comma method of string concatenation, to the + sign or f' string method:

```
The average test score is 60.67.
```

PDP Step 6. Fixes & Improvements

Example (yours should be different): This program would be more useful if the teacher could enter scores, rather than hard coding scores. Also, you could collect "points possible" as a separate variable, so the % correct could be calculated. More data analysis could be provided like min and max. It would be better if you asked for the number of students in the class, and then you could ask for scores for each one.

General Requirements (cont.)

All assignments must meet the following requirements:

The program must start with header at top and include appropriate comments throughout. Header example:

```
"""
Author: Erik Granse
Date: 2024-09-02
Description: Calculate and display student's average grades
"""
```

- Ensure the output is *information;* it needs to be a statement which explains the value being displayed (for example, “The average grade is 12.34”). Simply outputting “12.34” is not sufficient.

General Requirements

All assignments must meet the following requirements:

- The data in the program must be stored in variables.
- The output **must** come from variables in the program—do not simply hard code the output value in the `print()` statement.

Lab Section 1: Average Grade

- Using code provided in slide 5, finish this lab by adding code to get rid of the repeating decimal.
- Print the average grade rounded to two decimal places. Ensure both decimal places are shown even if the last one is zero (See lesson slide 19 for hints on formatting).
- Sample output: `The average test score is 60.67.`

Lab Section 2: Wage Calculation

- Use the previous lab as a model.
- Create a program that calculates and displays a sentence with a wage total that includes both regular and overtime pay.
- Facts:
  - Assume 40 hours worked this week, for which the regular wage is $15.34 per hour.
  - Assume an additional 10 hours of overtime worked this week, where the pay rate is 1.5 times the regular wage.
- Ensure the output value is rounded to two decimal places (no more, no less). See lesson slide 35 for tips.
- Sample output: `Your gross pay is $843.70.`

Lab Section 3: Album Counts

- Adapt the code from the lesson slide 26 – number of albums.
- Currently, when you run the code, it just displays the number 68 in the console/ interpreter. In your program, make sure the output explains what's going on, as shown in this sample output:

```
With 45 CDs and 23 LPs, the number of albums you have is 68.
```

Note: Use PDP steps to think through variables, calculations & printing.

Lab Section 4: Bus Fare

- Create a new program to calculate spending on bus fares for the month.
- In this program, regular bus fares are $1.75.
- Rush hour bus fares are $3.
- This month, you rode the bus 7 times during regular hours, and 12 times during rush hours.
- Calculate the amount spent on bus fare and print the result.
- Sample output: `I spent $48.25 this month on bus fare.`

Note: Use PDP steps to think through variables, calculations & printing.

Upload LAB files by the deadline

- Submit the code files for all four labs to the assignment in D2L
  - avg_grade.py
  - wage_calc.py
  - Albums.py
  - bus_fare.py.
- For each Python program, remember to add comments into your code, and check informational quality/ format of the output, by running, tweaking and re-running your code.
- Questions? Just ask me or one of the wonderful tutors!