# PROG 1700 PYTHON - Assignment 3

# Loops and Lists

Assignment Value: 10% of overall course mark.

Due Date: **See due date designated on the assignment dropbox on Brightspace.**

Late submissions will receive the standard late submission penalty as stated in the course outline. (5% overall deduction per day late and 0% after assignment handed back to the class.)

#### Assignment Instructions:

Create console applications (.py files) in which you’ll code the answer for each of the following problems. You must create a new .py file for each question in this assignment.

#### Submissions:

You will submit your work for this assignment via GitHub. While you will have frequent commits/pushes of your assignment code to GitHub as your work on it, the instructor needs to know which version to mark and when it was committed. So, when you have completed all assignment work, put a “Ready for Marking” comment on the last code committed into GitHub. Then, submit a simple text document to the Brightspace Dropbox that contains the git commit number/hash string (e.g. “b180b37”) that identifies that commit. It is this Dropbox submission that will be used to determine late penalties, so make sure to do so prior to the assignment deadline.

**Once you have committed the code, make sure to visit the repository page on GitHub’s website to verify that the final version has been pushed to GitHub as that is where the instructor will go to get your code.**

#### Evaluation:

**Program 1 (Time Sheet) will be marked as code review, in class on the day the assignment is due.**

To insure the greatest chance of success on this assignment, be sure to check the marking rubric contained at the end of this document or in D2L. The rubric contains the criteria your instructor will be assessing when marking your assignment.

## Program 1 – Time Sheet

## Design and write a program that accepts the number of hours worked on each of five work days from the user, then displays different information calculated about those entries as output.

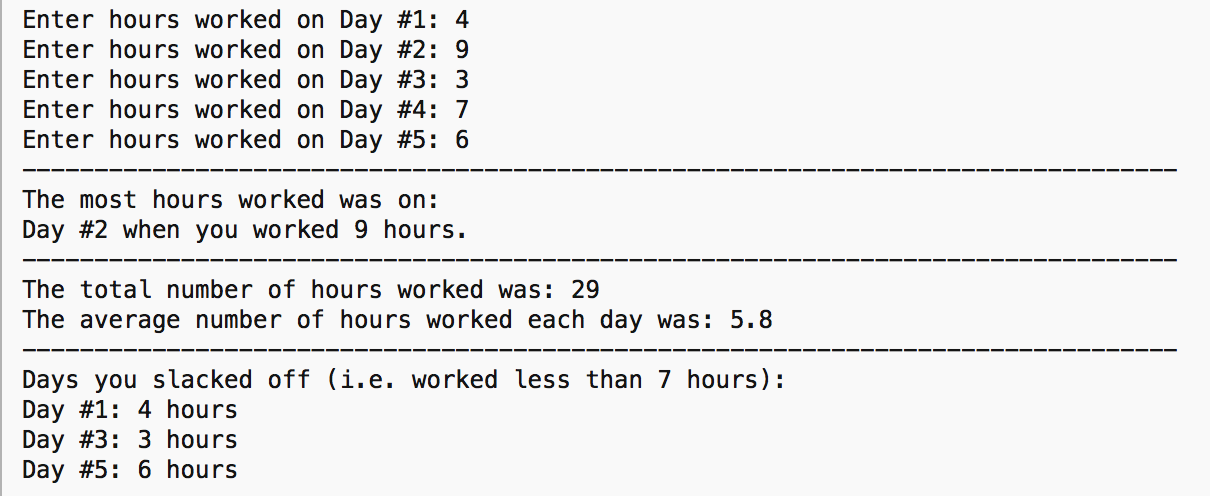
Your solution should demonstrate an understanding of how to apply list/loop concepts in a program that should:

* Include a flowchart that clearly shows all the paths of execution that will exist within your designed solution to this problem.
* Prompt the user to enter the five daily hours worked amounts and store them in the program.
* Determine the day(s) on which the most hours were worked and display the day(s) and hours onscreen.
* Calculate and display both the total and the daily average of hours worked.
* Display a list of all days that had insufficient hours, which is defined as less than 7 hours.

Your solution must contain examples demonstrating your understanding of appropriate use of functions and core assignment concepts (loops and lists).

### Examples & Testing

In the section below you will be presented with at least one screenshot of a successful execution of a sample solution to the program, which should help demonstrate how your input/output on the program should work. In addition to the sample values used in the screenshot(s), additional testing values are given in a chart along with the output values that they should produce. You can expect your instructor to grade your assignment by using all of these listed input values, but additional values may also be used. In other words, **you should thoroughly test your code before submitting!**

Sample Output  


### Additional Testing Values & Expected ResuLts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Five Input Hours | Highest Day(s) | Total | Average | Less than 7-hour Days |
| 6, 8, 5, 8, 4 | Day 2 – 8 hours  Day 4 – 8 hours | 31 | 6.2 | Day 1 – 6 hours  Day 3 – 5 hours  Day 5 – 4 hours |
| 4,7,11,10,7 | Day 3 – 11 hours | 39 | 7.8 | Day 1 – 4 hours |

## Program 2 – Message Redaction

Design and write a program that removes all desired letters from any user-entered sentence or phrase.

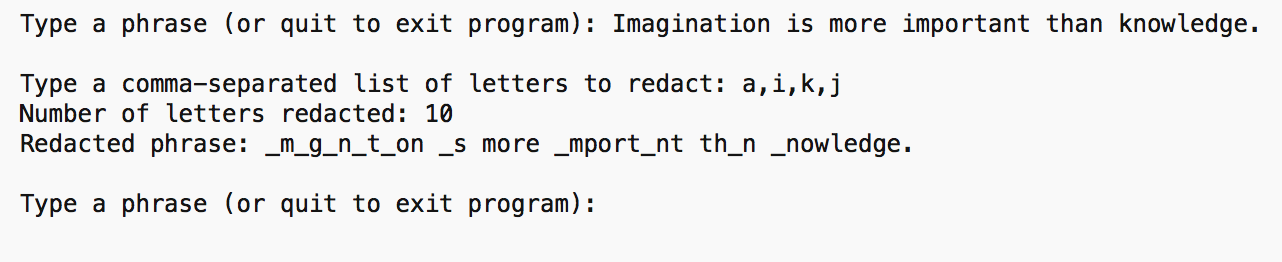
Your solution should demonstrate an understanding of how to apply list/loop concepts in a program that should:

* Take a sentence or phrase as input,
* Take a comma-separated list of letters to remove as input,
* Replace all occurrences of each target letter in the user-entered sentence, regardless of case sensitivity, with an underscore (“\_”) character.
* Display the number of letters removed to the user,
* The program will keep asking the user to enter a new sentence until the user enters the command ‘quit’.

Your solution must contain examples demonstrating your understanding of appropriate use of functions and core assignment concepts (loops and lists).

### Examples & Testing

In the section below you will be presented with at least one screenshot of a successful execution of a sample solution to the program, which should help demonstrate how your input/output on the program should work. In addition to the sample values used in the screenshot(s), additional testing values are given in a chart along with the output values that they should produce. You can expect your instructor to grade your assignment by using all of these listed input values, but additional values may also be used. In other words, **you should thoroughly test your code before submitting!**

Sample Output  


### Additional Testing Values & Expected ResuLts

|  |  |  |  |
| --- | --- | --- | --- |
| Entered Phrase | Letters to redact | # letters removed | Redacted Phrase |
| The medium is the message! | m,e,b | 8 | Th\_ \_\_diu\_ is th\_ \_\_ssag\_! |
| I want to put a ding in the universe. | a,g,x,n | 7 | I w\_\_t to put \_ di\_\_ i\_ the u\_iverse. |

## Program 3 – Girl Guide Cookie Sell-off

The organizers of the annual Girl Guide cookie sale event want to raise the stakes on the number of cookies sold and are offering cool prizes to those guides who go above and beyond in their sales efforts. The organizers want a program to print the guide list and their prizes.

Your solution should demonstrate an understanding of how to apply list/loop concepts in a program that should:

* A flowchart that clearly shows all the paths of execution that will exist within your designed solution to this problem.
* Ask the user how many guides sold cookies in the current event,
* For each numbered guide up to the user-entered count of guides, allow the user to enter a name and the number of boxes of cookies that person sold.
* Calculate and output a list of all guide names,
* Display the average sales, and the prize that each guide won,
* The highest selling guide gets a trip to the Girl Guide Jamboree, any guides selling above average get a badge, and any guides selling at least one box get to split the remaining cookies, with guides selling no boxes shut out of all prizes (as they should be).

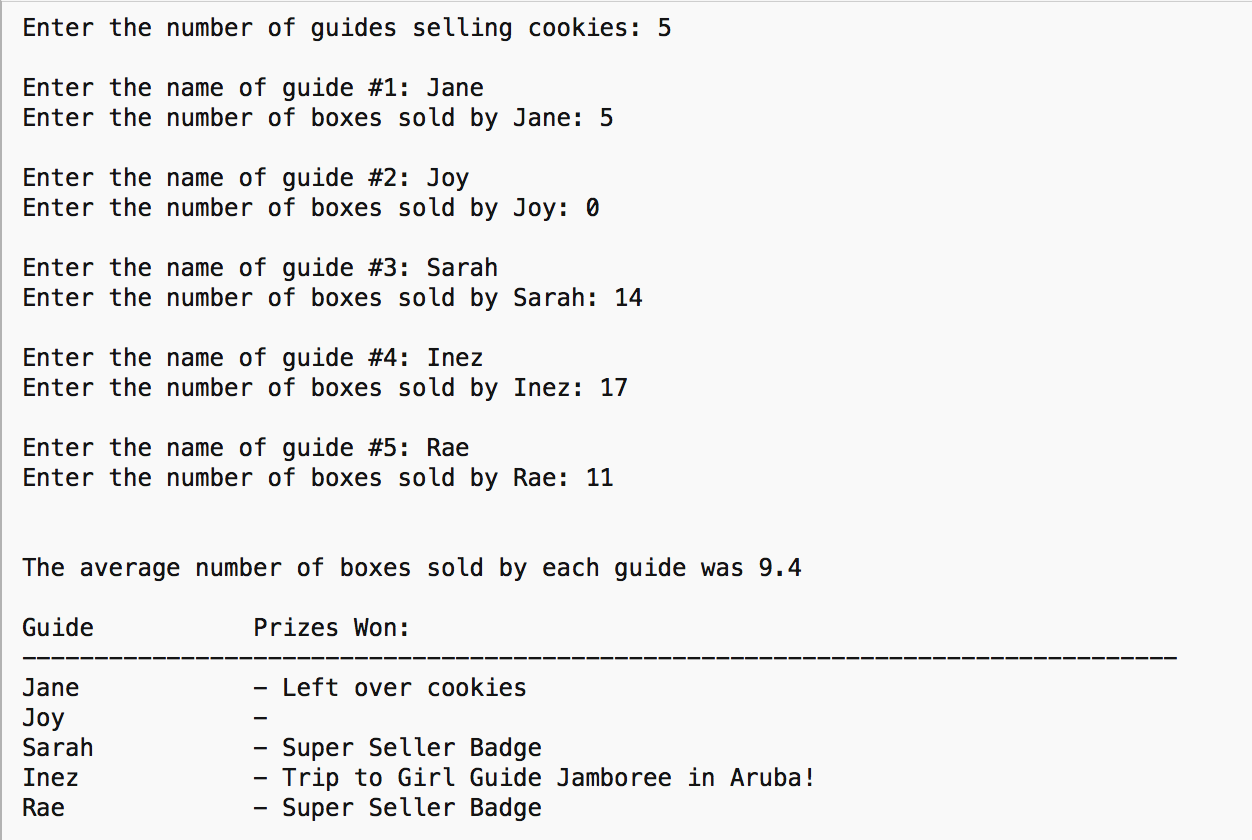
*(Hint: Some potential solution ideas: Research parallel arrays or two-dimensional lists)*

Your solution must contain examples demonstrating your understanding of appropriate use of functions and core assignment concepts (loops and lists).

### Examples & Testing

In the section below you will be presented with at least one screenshot of a successful execution of a sample solution to the program, which should help demonstrate how your input/output on the program should work. In addition to the sample values used in the screenshot(s), additional testing values are given in a chart along with the output values that they should produce. You can expect your instructor to grade your assignment by using all of these listed input values, but additional values may also be used. In other words, **you should thoroughly test your code before submitting!**

Sample Output



More testing values & expected results on next page.

### Additional Testing Values & Expected ResuLts

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Guide 1 | | Guide 2 | | Guide 3 | | Guide 4 | | | | |  | | |
| Number of  Guides | **Name** | **Boxes** | **Name** | **Boxes** | **Name** | **Boxes** | | | **Name** | | **Boxes** | | | **Results** | | |
| 3 | Emily | 12 | May | 3 | Keisha | 14 | | - | | - | | |  | | |
| 4 | Ming | 45 | Kay | 11 | Jessie | 1 | | Newt | | 16 | | |  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Program 1 – Time Sheet** | |  |  |  |  |  |
| **Criteria** | **Insufficient**  **(0 pts)** | **Needs Development**  **(1-2 pts)** | **Sufficient**  **(3-4 pts)** | **Excellent (5 pts)** | **Mark** | **X** |
| **Flowchart** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | A comprehensive effort was made to plan out the program using a flowchart that demonstrates the necessary paths through the program based on all potential decision points. |  |  |
| **Input / Output** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | All of:   * The five user input values can be successfully captured using descriptive prompts * All output lines are well-formatted and contain all expected information * All inputs are appropriately validated for expected values/data types. |  |  |
| **Highest Day(s)** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Any and all days with the maximum number of hours worked are displayed, with both the number of the day and the hours worked reported. |  |  |
| **Total** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | The total number of hours worked is always calculated and displayed correctly, using an appropriate method. |  |  |
| **Average** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | The average number of daily hours worked is always calculated and displayed correctly, using an appropriate method. |  |  |
| **Less than Sufficient** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | All days with hours worked less than the normal amount are calculated and displayed correctly, using a loop or list-appropriate method. The normal hours worked, which is set at 7, is not used as a “magic number”. |  | 2 |
| **Use of Functions & Core Concepts** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Solution contains at least one example demonstrating strong understanding of appropriate function use, including use of parameters and return values, and with no global variable use. (Using or adapting the standard main() function does not count) and looping and list concepts. |  |  |
| **Comments & Best Coding Practices**  (At least 60% of the functional requirements must be complete) | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Organizational or explanatory comments are used extensively, most are meaningful and easily understood. A consistent naming convention was used for most of the program and deviated very little. Source code was clean, consistently well-formatted and easy to read |  | 2 |
|  |  |  |  | **Total:** |  | **/50** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Program 2 – Message Redact** | | | |  | |  |  | |  |
| **Criteria** | **Insufficient (0 pts)** | | **Needs Development**  **(1-2 pts)** | **Sufficient (3-4 pts)** | | **Excellent (5 pts)** | **Mark** | | **X** |
| **Input / Output** | Little to no effort was made, or contains too many errors / omissions. | | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | | All of:   * The user input values can be successfully captured using descriptive prompts * All output lines are well-formatted and contain all expected information * All inputs are appropriately validated for expected values/data types. |  | |  |
|  | |  |
| **Phrase Reading** | Little to no effort was made, or contains too many errors / omissions. | | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | | Any user-entered phrase is completely evaluated when searching for letters to remove, using a loop or list-appropriate method. |  | | 2 |
| **Letter Removal** | Little to no effort was made, or contains too many errors / omissions. | | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | | All occurrences of the entered letters are removed for any word or phrase, using a loop or list-appropriate method.  Letter casing is properly considered. |  | | 2 |
| **Letter Count** | Little to no effort was made, or contains too many errors / omissions. | | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | | The correct number of removed letters is displayed for any entered phrase and combination of letters to remove. |  | |  |
| **Use of Functions & Core Concepts** | Little to no effort was made, or contains too many errors / omissions. | | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | | Solution contains at least one example demonstrating strong understanding of appropriate function use, including use of parameters and return values, and with no global variable use. (Using or adapting the standard main() function does not count) and looping and list concepts. |  | |  |
| **Comments & Best Coding Practices**  (At least 60% of the functional requirements must be complete) | Little to no effort was made, or contains too many errors / omissions. | | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | | Organizational or explanatory comments are used extensively, most are meaningful and easily understood.  A consistent naming convention was used for most of the program and deviated very little.  Source code was clean, consistently well-formatted and easy to read |  | | 2 |
|  |  | |  |  | | **Total:** |  | | **/45** |
|  | | |  | |  |  | |  |  |
|  | | |  | |  |  | |  |  |
| **Program 3 – Girl Guides** | | | | |  |  | |  |  |
| **Criteria** | | **Insufficient**  **(0 pts)** | **Needs Development**  **(1-2 pts)** | | **Sufficient**  **(3-4 pts)** | **Excellent (5 pts)** | | **Mark** | **X** |
| **Flowchart** | | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | | A good effort was made, but at least one error or omission exists. | A comprehensive effort was made to plan out the program using a flowchart that demonstrates the necessary paths through the program based on all potential decision points. | |  |  |
| **Input / Output** | | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | | A good effort was made, but at least one error or omission exists. | All of:   * All user input values can be successfully captured using descriptive prompts * All output lines are well-formatted and contain all expected information * All inputs are appropriately validated for expected values/data types. | |  |  |
|  |  |
| **Guide Names/Boxes** | | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | | A good effort was made, but at least one error or omission exists. | All given candidate names/boxes were stored in dynamically-sized lists, based on the number of guides entered by the user. Employs appropriate use of lists and/or loops. | |  | 3 |
| **Average Boxes** | | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | | A good effort was made, but at least one error or omission exists. | The average boxes sold for each candidate calculated without errors. | |  |  |
| **First Prize** | | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | | A good effort was made, but at least one error or omission exists. | The first prize winner was determined correctly using any number of guides, in any order, with any number of boxes sold | |  |  |
| **Alternate Prizes / No Prizes** | | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | | A good effort was made, but at least one error or omission exists. | The guides that were not the highest are assigned an alternate prize appropriately if they have sold at least one box using any number of guides, in any order, with any number of boxes sold.  A guide selling no boxes is granted no prize. | |  | 2 |
| **Use of Functions & Core Concepts** | | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | | A good effort was made, but at least one error or omission exists. | Solution contains at least one example demonstrating strong understanding of appropriate function use, including use of parameters and return values, and with no global variable use. (Using or adapting the standard main() function does not count) and looping and list concepts. | |  |  |
| **Comments & Best Coding Practices**  (At least 60% of the functional requirements must be complete) | | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | | A good effort was made, but at least one error or omission exists. | Organizational or explanatory comments are used extensively, most are meaningful and easily understood. A consistent naming convention was used for most of the program and deviated very little. Source code was clean, consistently well-formatted and easy to read | |  | 2 |
|  | |  |  | |  | **Total:** | |  | **/60** | |