Southern New Hampshire University

IT 140 Final Project Script Two Guidelines and Rubric

Overview

Apple cofounder Steve Jobs once said "I think everybody in this country should learn how to program a computer because it teaches you how to think." Regardless of your future career path, the programming concepts and the computational thinking practices gained while learning to script will prove useful. In this course, you will have the opportunity to explore some of the introductory types of scripting and computational thinking that information technology professionals engage in when they design solutions to social and business problems.

Programming is a both a scientific and creative undertaking. The fundamentals you learn in this course will provide the scientific foundation. Armed with these skills, you will have the ability to expand the scientific elements to more advanced concepts that can help you solve everyday problems using tools in creative ways.

For your final project, you will be creating four small scripts. For each script assignment, you will also write a small reflective essay (one to two paragraphs in a Microsoft Word document) in which you will discuss the problem-solving approaches and relevance of the programming concepts used in the script. Collectively, all four small scripts and reflective essays will demonstrate your ability to engage in the fundamental scripting and problem-solving approaches that are represented by the course outcomes:

- **IT-140-01:** Apply appropriate data structures to store and manipulate data for solving computational problems **[BS.ITE.CORE.05]**
- **IT-140-02:** Utilize branches and loops to insert decision points for managing the execution of code **[BS.ITE.CORE.01]**
- **IT-140-03:** Utilize functions for producing organized, reusable code **[BS.ITE.CORE.01]**
- **IT-140-04:** Write accurate regular expressions that search for specific patterns in text material **[BS.ITE.CORE.01]**
- **IT-140-05:** Explain problem solving approaches required for implementing and troubleshooting scripting tasks **[BS.ITE.CORE.01]**

You will have an opportunity to submit a practice draft of each script early in the course and receive feedback from your instructor before you finalize the script and submit the final version and reflection in a later module. Take advantage of these practice opportunities! Even if your script is not functioning perfectly, submit your draft and get feedback so that you can improve on it for the final submission. Follow the directions in the practice modules in Codio.

The following table explains when the drafts and final versions of each script are due:

IT 140 Final Project			
Script Number	**Name and Description**	**Draft Submitted Through Codio**	**Final Commented Script and Reflection Due**
One	**Rental Car Billing Script:** This first artifact will address basic data types and decision-making control structures (branches).	**Module Three**	**Module Five**
Two	**Grocery List Script**: This second artifact focuses on collection-type data structures (lists and dictionaries) and iterative loops (for and while).	**Module Four**	**Module Six**
Three	**ATM Script**: This third artifact addresses developing functions.	**Module Five**	**Module Seven**
Four	**Pattern Search and Replace Script**: This fourth artifact involves your use of regular expressions to search for specific patterns.	**Module Six**	**Module Eight**

Prompt

<u>**Part Two: Grocery List Script**</u>

The *list* object type is one of the most important and frequently used types in a Python program. A list is a container, an object that groups related objects together. A list is also a sequence: The contained objects maintain a left-to-right positional ordering. Elements of the list can be accessed via indexing operations that specify the position of the desired element in the list. Each element in a list can be a different type, such as string, integer, float, or even other list.

A dictionary is another type of container object that is different from sequences like strings, tuples, and lists. Dictionaries contain references to objects as key–value pairs: Each key in the dictionary is associated with a value, much like each word in an English language dictionary is associated with a definition. Unlike sequences, the elements of a dictionary do not have a relative ordering of positions. Adding a value to a dictionary can be done either with brackets or using the keyword *dict*.

A programmer can use loops to execute the same code repeatedly as long as some condition is true. The *loop* expression is evaluated when the program reaches the loop statement. If the loop expression is true, then the indented code block, known as the loop body, is executed. At the end of the loop body, execution goes

back to the while loop statement. The loop expression is evaluated again, and if true, the loop body is executed again. But, if the expression evaluates to false, then execution instead proceeds to code below the loop body. Each execution of the loop body is called an iteration, and looping is also called iterating.

Your task for this project is to create a very simple grocery list script. This script emphasizes the importance of using lists, dictionaries, and loop types within your script, and how the use of those functions shapes your approach to creating a script.

You will be working on this project in the Project Two: Grocery List Script module in Codio. Following the directions in that module, you will also be able to determine the exact placement of the comments you will need to make in the code. Follow the directions in the module in Codio to walk through the activity.

Your script and word response for this script should address the following critical elements:

I.   In Your Script (Annotated Text File):
Refer to the directions in the module in Codio for how to export out and comment your completed script.
A. Identify examples of four uses of **list** operations in the script using comments in your code. Be sure your examples address each of the following:
i.   Creating lists
ii. Adding and removing data from a list
iii. Accessing values in a list
iv. Modifying values in a list
B. Identify examples of four uses of **dictionary** operations using comments in your code. Be sure your examples address each of the following:
i.   Creating dictionaries
ii. Adding and removing key–value pairs
iii. Accessing values using keys
iv. Modifying values
C. Identify examples of three uses of **loop** structures using comments in your code. Be sure your examples address each of the following:
i.   Item-based for loops
ii. Index-based (range) for loops
iii. While loops

II.   **Applying Your Experience**
Making mistakes when you learn to write code is common. It is part of learning. What is important is developing the skill of learning how to understand your errors and then fix them (debugging). For this part of your final project, you will respond to the following:
A. Reflecting on your experience with this activity, explain the importance of knowing how and when to use lists, dictionaries, and loop types. Support your response with examples from the activity of the types of errors and your method for fixing them.

Southern New Hampshire University

Final Project Script Two Rubric

Guidelines for Submission: Your submission should include all annotated text files of code for the script, as well a Microsoft Word document that includes the reflection part of your assignment in which you explained the problem-solving approaches you employed (critical element II).

Critical Elements	Exemplary	Proficient	Needs Improvement	Not Evident	Value
List **[IT-140-01]**		Identifies examples of four uses of list functions in the script using comments in your code (100%)	Identifies examples of four uses of list functions in the script using comments in your code, but uses are inappropriate or inaccurate, or comments lacks key details (55%)	Does not identify examples of four uses of list functions in the script using comments in your code (0%)	24
Dictionary **[IT-140-01]**		Identifies examples of four uses of dictionary functions using comments in your code (100%)	Identifies examples of four uses of dictionary functions using comments in your code, but uses are inappropriate or inaccurate, or comments lack key details (55%)	Does not identify examples of four uses of dictionary functions using comments in your code (0%)	24
Loop **[IT-140-02]**		Identifies uses of loop functions using comments in your code (100%)	Identifies uses of loop functions using comments in your code, but uses are inappropriate or inaccurate, or comments lack key details (55%)	Does not identify uses of loop functions using comments in your code (0%)	24
Applying Your Experience **[IT-140-05]**		Explains the importance of knowing how and when to use lists, dictionaries, and loop types, and explanation is supported with examples from the activity about the types of errors and your method for fixing them (100%)	Explains the importance of knowing how and when to use lists, dictionaries, and loop types, and explanation is supported with examples from the activity about the types of errors and your method for fixing them but is illogical, lacks key details, or is not specific to lists, dictionaries, and loop types (55%)	Does not explain the importance of knowing how and when to use lists, dictionaries, and loop types (0%)	24

Articulation of Response	Submission is free of errors related use of citations, grammar, spelling, syntax, and organization and is presented in a professional and easy-to-read format (100%)	Submission has no major errors related to use of citations, grammar, spelling, syntax, or organization (85%)	Submission has major errors related to use of citations, grammar, spelling, syntax, or organization that negatively impact readability and articulation of main ideas (55%)	Submission has critical errors related to use of citations, grammar, spelling, syntax, or organization that prevent understanding of ideas (0%)	4
				Total	100%