

IT 140 Final Project Script Three 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:

- Apply appropriate data structures to store and manipulate data for solving computational problems
- Utilize branches and loops to insert decision points for managing the execution of code
- Utilize functions for producing organized, reusable code
- Write accurate regular expressions that search for specific patterns in text material
- Explain problem solving approaches required for implementing and troubleshooting scripting tasks

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

Part Three: ATM Script

A function is a named series of statements. Invoking a function's name, known as a function call, causes the function's statements to execute. Python comes with a number of built-in functions, such as *input()*, *int()*, *type()*, and others. A programmer can also create a new function using the *def* keyword, the new function's name, and a block of statements. A block is a series of indented statements following the function definition.

To call a function, use the function's name followed by braces: *function_name()*. This causes execution to jump to the function's statements. The function's return causes execution to jump back to where the original call occurred, to the next statement after the call. A good practice is to follow the convention of naming functions with lowercase letters and underscores, such as *print_names* or *print_face*.

Your task for this project is to create a simple ATM script. The script emphasizes the importance of using functions in the creation of more organized and reusable code.

You will be working on this project in the Project Three: ATM Script module in Codio on this project. 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 should do the following three things:

1. Collect customer input
2. Calculate the ending balance
3. Display the results to the customer

Specifically, your 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 **custom functions** in your script using comments in your code.
- B. Identify examples of **input (parameters)** that are utilized within the **function(s)** in your script using comments in your code.
- C. Identify examples of **functions that return the correct output** in your script using comments in your code.

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 and modify custom functions, inputs (parameters) within functions, and functions to return the correct output. Support your response with examples from the activity of the types of errors and your method for fixing them.

Final Project Script Three 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
Custom Functions		Identifies examples of custom functions in your script using comments in your code (100%)	Identifies examples of custom functions in your script using comments in your code, but examples are inappropriate or inaccurate, or comments lack key details (55%)	Does not identify examples of custom functions in your script using comments in your code (0%)	24
Input (Parameters) Function(s)		Identifies examples of input (parameters) that are utilized within the function(s) in your script using comments in your code (100%)	Identifies examples of input (parameters) that are utilized within the function(s) in your script using comments in your code, but examples are inappropriate or inaccurate, or comments lack key details (55%)	Does not identify examples of input (parameters) that are utilized within the function(s) in your script using comments in your code (0%)	24
Functions That Return Correct Output		Identifies examples of functions that return the correct output in your script using comments in your code (100%)	Identifies examples of functions that return the correct output in your script using comments in your code, but examples are inappropriate or inaccurate, or comments lack key details (55%)	Does not identify examples of functions that return the correct output in your script using comments in your code (0%)	24
Applying Your Experience		Explains how using functions is important in the creation of more organized and reusable code, and submission is supported with examples from the activity about the types of errors and your method for fixing them (100%)	Explains how using functions is important in the creation of more organized and reusable code, and submission 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 using and modifying variables and branches (55%)	Does not explain how using functions is important in the creation of more organized and reusable code (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%