Name: Jeff Alkire

Date: October 8, 2022

Due Date: October 23, 2022

Class: CIS 087 Python

Program Purpose:

To solve a problem I have. The problem is as follows:

1. I have a media server I use to host music, videos, photos, etc.
2. This includes recaps of each Dodger game for the last 3 years.
3. The media server stores an image (called a poster) with each game.
4. Sometimes the media server includes an image that includes the winner of the game
5. I don’t want to know the winner before I watch the video.

The solution to the problem is as follows:

1. The media server allows the user to specify a chosen image for the poster.
2. The administrator does this by creating a file names almost identically to the original video file. The only difference is that the extension of the poster must be an image. (In our case the extension will be .jpg).
3. I will the posters all to be the same (a simple Dodger logo).

The current situation:

1. The current situation is unfortunate.
2. I have begun creating the poster files for the three seasons, but I have not finished.
3. Also, I will add games as the Dodgers play them.
4. So, generally, I do not know which videos have posters and which do not.
5. My program will need to check a certain set of directories for video files (which will end with .mkv or .mp4 extensions). For each video file found, it will need to determine if a poster file exists. For those situations where it does not, it must copy the poster.jpg file to the proper directory and give it the proper name.

Methods, tools, and strategies by chapter:

Chapter 2: Data Types and Expressions, do some Math

Page 42: String Literals

Lines 10 & 11 contain examples of string literals used as constant values.

Page 43: String Concatenation

An example of string concatenation is found on line 104.

Page 44: Variables and the Assignment Statement

Line 104 also contains as assignment as do lines 72 and 73.

Page 45: Program Comments and Docstrings

The file begins with a docstring.

Each function contains a docstring

Lines 76, 78, and 81 contain simple comments as well.

Page 54: Calling Functions: Arguments and Return Values

The string function append is called on lines 80 and 83

The library function os.listdir() is called on line 75

The library function shutil.copyfile() is called on line 106

Several instances of calling user created functions are found throughout the utility including lines 129, 124, 126, and 117.

Chapter 3: Control Statements, use both loops and selection

Page 65: The for loop

for loops are found on lines 77, 115, and 125

Page 77: if and if-else statements

If statements is found on line 116 and 128

Page 80: Multi-Way if statements

An if-elif block is found beginning at line 79

Page 82: Logical Operators and Compound Boolean Expressions

Boolean expressions are found on lines 39 and 48 in the return statements.

Boolean expressions are also found in the if statements on lines 79, 82, 116 and 128.

Line 39 contains an or making it a compound boolean expression.

Chapter 4: Strings and Text Files

Page 103: Accessing Characters and Substrings in Strings

Line 38 contains an example of a substring.

Page 104: The Subscript Operators

Line 38 uses the subscript operator ([]) to create the substring.

Page 105: Slicing for Substrings

Line 48 makes use of slicing.

Page 105: Testing for a Substring with the in Operator.

Line 116 makes use of the in operator. It does not use it with a string, but rather a list of strings. This is similar because the in operator is a list operator that works on strings because a string is a list of characters.

Page 115: String Methods

Line 57 makes use of the rfind (reverse find) string method to properly find the last period/dot in the filename.

Page 120: Reading Text from a File

The function on lines 13-29 reads text from a file. This text is the list of directories to process.

Page 122: Accessing and Manipulating Files and Directories on Disk

open is used on line 22

os.listdir() is used on line 75

shutil.copyfile() is used on line 106

Chapter 5: Lists and Dictionaries

Page 135: Lists

Lists are used in this program to keep track of all video and image files in a single directory. For example, the read\_dir\_list function (line 13) returns a list of strings, and the read\_dir\_videos\_and\_images() returns 2 lists of filenames as strings.

Page 140: Searching a List

Line 116 makes use of the in operator to search a list for a value.

Page 146: Defining Simple Functions

Several function are defined in this program by the author (me) including at lines 87, 95, and 108.

Page 147: Parameters and Arguments

String (str) arguments are accepted by functions at lines 95, 108, and others.

Page 147: The return statement

return statements are found on lines 29, 39, 48, 58, and 85

Page 148: Defining a main function

The main function is defined at line 119 and an entry point if statement at line 128.