

Honors Computer Science Python

Mr. Clausen

Program 7A, 7B

PROGRAM 7A Turtle Graphics Animation (100 points)

Here is the overview of the program. Use functions to draw two background scenes. These can be nature scenes, or city scenes with lots of details in either choice. Since I am asking for nature or city scenes, no “abstract art” is permissible, you have to draw something recognizable as a nature or city scene. You may use some algorithmic drawing to add realism, but only as a small percentage of your code. For example, you may use fractals to draw better looking tree branches or blades of grass. You can use any of the commands that are in the PowerPoint presentation or on the [Turtle Graphics Documentation](#) web page. Try not to make everything in your scene the same Geometric shape (so, no “space” scenes where everything is a circle). Start by going to the Resource folder for this class “HnrCSPythonFiles”, and find the folder “Turtle Graphics”. Copy the two files “TurtleGraphicsTemplate.py”, and “turtle.cfg” to your “S:” directory. Open the template and use this to create your artwork. Do not change any of the screen sizes in either file.

After you create the two background scenes, use the PrintScreen key on your keyboard to do screen captures, edit the pictures to crop, resize to 800 by 400 and save as transparent GIF files.

Next, comment out the function calls to all the functions that draw Scene1 and Scene2 (do NOT delete the functions) and load each background picture when necessary. Instantiate a minimum of two turtles and assign transparent gif files to each turtle. One of the turtle shapes must be drawn in a paint program, while the other can be downloaded. Move these turtles across the screen simultaneously for your animation (this program is like program L5 in the Intro to Programming class).

- 1) Use a `DocString` at the beginning of the program for your comments.
- 2) Use the following import statement: `from turtle import *`
- 3) Use print statements to display your name and period output using a function named “displayMyInfo” just like program 5B.
- 4) Make sure that you use descriptive identifiers for all of your variables and function names. You can use “t” for the Turtle object.
- 5) Now for the fun part. Create a separate function for each part of each scene. For instance, if you draw a mountain scene, you will need separate functions to draw the mountain, the sky, trees, etc. If the mountain has a lot of detail, you should also have separate functions for each part of the mountain.
- 6) You must use functions throughout this program. You must have a main function that declares all the variables (except variables that may be local to a function), and calls all the other functions. Don’t forget that the last line of code should be “main()” so the program will run. Remember to include a docstring in every function that includes the “type contract”, a

description of the function, and a sample function call where applicable.

- 7) For some guidelines as what to include in your drawing, you must use color in your drawing. If you use any custom colors based upon RGB ratios or hexadecimal numbers, you must include a comment after the color to tell me what color it is. You must also FILL in all closed shapes.
- 8) After you create the two background scenes, use the PrintScreen key on your keyboard to do screen captures, edit the pictures to crop, resize to 800 by 400 and save as transparent GIF files.
- 9) Next, comment out the function calls to all the functions that draw Scene1 and Scene2 (do NOT delete the functions) and load each background picture when necessary.
- 10) Instantiate a minimum of two turtles and assign transparent gif files to each turtle. One of the turtle shapes must be drawn in a paint program, while the other can be downloaded.
- 11) Move these turtles across the screen simultaneously for your animation (this program is like program L5 in the Intro to Programming class).
- 12) Save your program as LastNameFirstNameP7A.py.

PROGRAM 7B Image Processing (30 points)

Here is the overview of the program. We are writing an image processing program that will take a .gif file (smokey.gif – the cat picture that came with our textbook) and do some simple image processing similar to a graphics program like Photoshop. **Write code to flip the picture vertically, flip the picture horizontally, rotate the picture 90° clockwise, and rotate the picture 90° counterclockwise.** Note: the picture, smokey.gif is not a square, but rectangular in its dimensions. Make sure that your source code and the picture “smokey.gif” are in the same folder. You can find the picture of Smokey.gif in the folder path: \Resource\HnrCSPythonFiles\Ch 7 Image Files. There is a sample program in the same folder that copies the picture pixel by pixel and saves the copy. Feel free to use this as a reference, since you will be changing a copy of the picture and not the original image.

You need to display the original picture and the processed **copy** of the picture after performing each effect. After processing the picture, save the processed copy using a different file name. For example, after rotating the picture clockwise 90°, save the picture as: “YourLastNameYourFirstNameSmokeyRotate90CW.gif”.

- 1) Use a DocString at the beginning of the program for your comments.
- 2) Use the following import statement: from images import Image
(Note: this will only allow us to do image processing on .gif files and not any other format.)
- 3) Use print statements to display your name and period output using a function named “displayMyInfo” just like program 5B.
- 4) Make sure that you use descriptive identifiers for all of your variables and function names.

- 5) You must use functions throughout this program. You must have a main function that declares all the variables (except variables that may be local to a function), and calls all the other functions. Don't forget that the last line of code should be "main()" so the program will run. Remember to include a docstring in every function that includes the "type contract", a description of the function, and a sample function call where applicable.
- 6) Display and use the menu listed below in your program.

```
Image Processing
1. Flip Horizontal
2. Flip Vertical
3. Rotate 90 degrees Clock Wise
4. Rotate 90 degrees Counter-Clock Wise
Q. Quit the program
Enter your choice
```

- 7) Follow the sample program that displays the original picture and prints the message: "Close the image window to continue." After closing the original picture, display the modified picture. After closing the modified picture, return to the menu (so you will need a loop with the menu).
- 8) You need to display the original picture and the processed **copy** of the picture after performing each menu choice effect. After processing the picture, save the processed copy using a different file name. For example, after rotating the picture clockwise 90°, save the picture as: "YourLastNameYourFirstNameSmokeyRotate90CW.gif".
- 9) Save your program as LastNameFirstNameP7B.py.