# **CVI620 – Assignment 1**

Summer 2025

| Total Mark: | 7.5 marks (7.5% of the total course grade) |
| --- | --- |
| Submission file(s): | * Assignment1.py or Assignment1.ipynb (you can upload multiple files) * Assignment1.docx (this document with your answers) |
| Deadline | * October 11th, 2025 |

If you are unable to complete the assignment on-time for any legit reason, please provide documentation explaining your absence (e.g., an appointment confirmation or a work letter).

Please be aware that the assignment is designed to make you **research** and develop.

Please submit the submission file(s) through Learn@Seneca. Make sure to use GitHub and provide the link to your GitHub account for all your contributions in the box below:

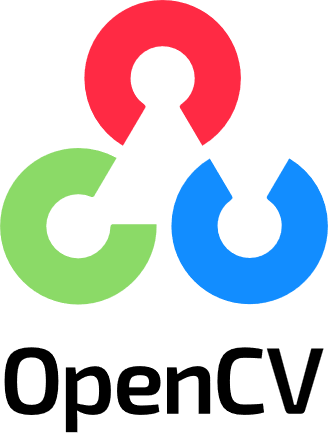
|  |  |
| --- | --- |
| Project GitHub repository: |  |

Share with Ellie749 if private.

**Please paste the resulting images and answers in this document.**

## **Part I: Shapes**

Recreate the OpenCV logo using OpenCV drawing functions only (e.g., cv2.circle, cv2.line, etc.).



You must draw:

* Three colored shapes (ellipse: Blue, Green, Red) arranged in a triangular pattern.
* Proper positioning and size of circles.
* The text "OpenCV" at the center/bottom.

## **Part II: Image Arithmetic**

You are asked to build a simple invisible cloak effect using OpenCV.

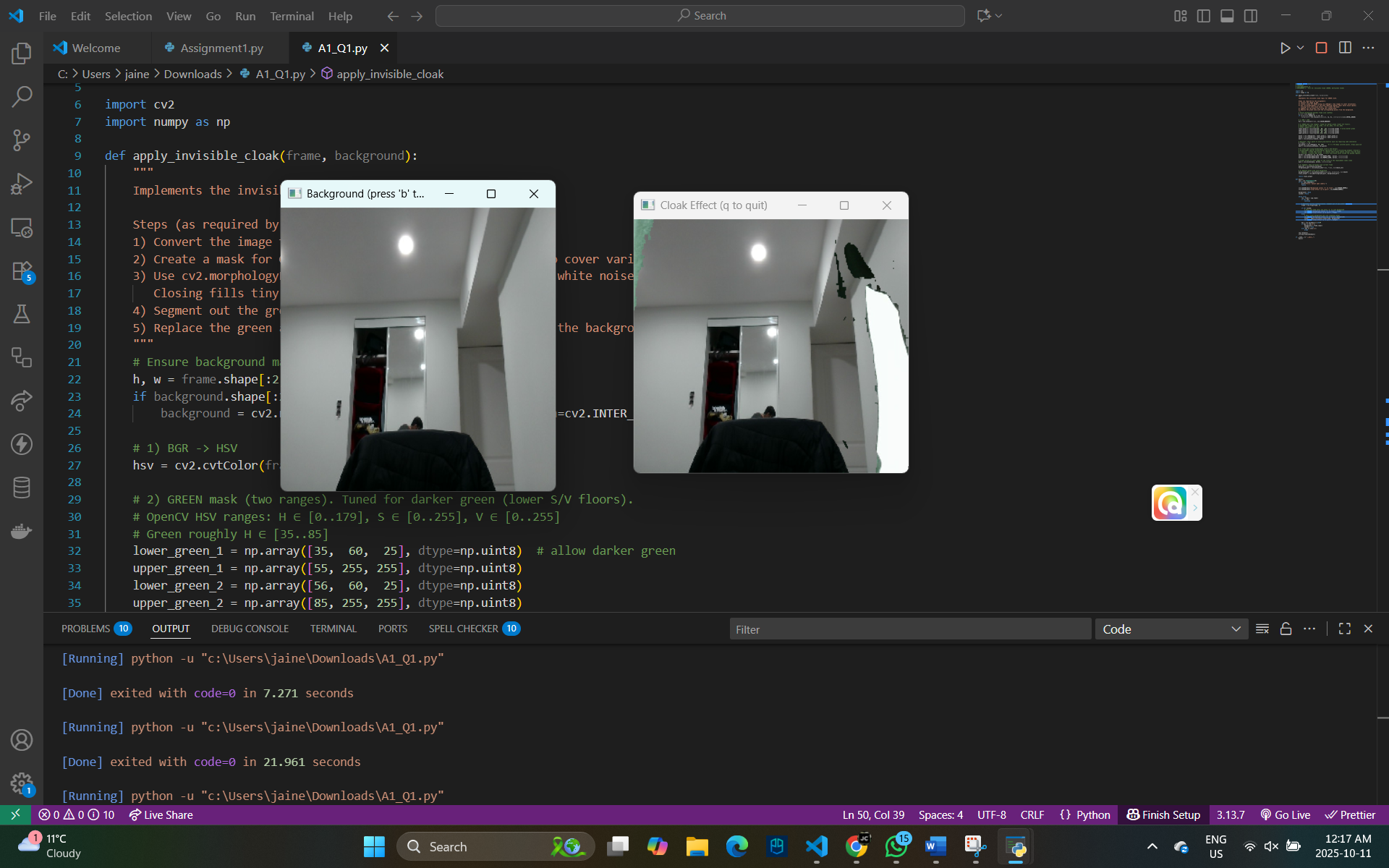
The idea is to make a specific colored cloth worn by the user become invisible by replacing it with the background.

Instructions:

Complete the function in A1\_Q2.py file to implement the logic. The function should:

* Convert the image to HSV.
* Create a mask for green color using cv2.inRange() (you can use two ranges).
* Search about cv2.morphologyEx() and **explain it in your code**. Then use it on your mask if needed.
* Segment out the green area using the mask.
* Replace the green area with the background.

Upload your .py file and **a attach few sample photos here.**

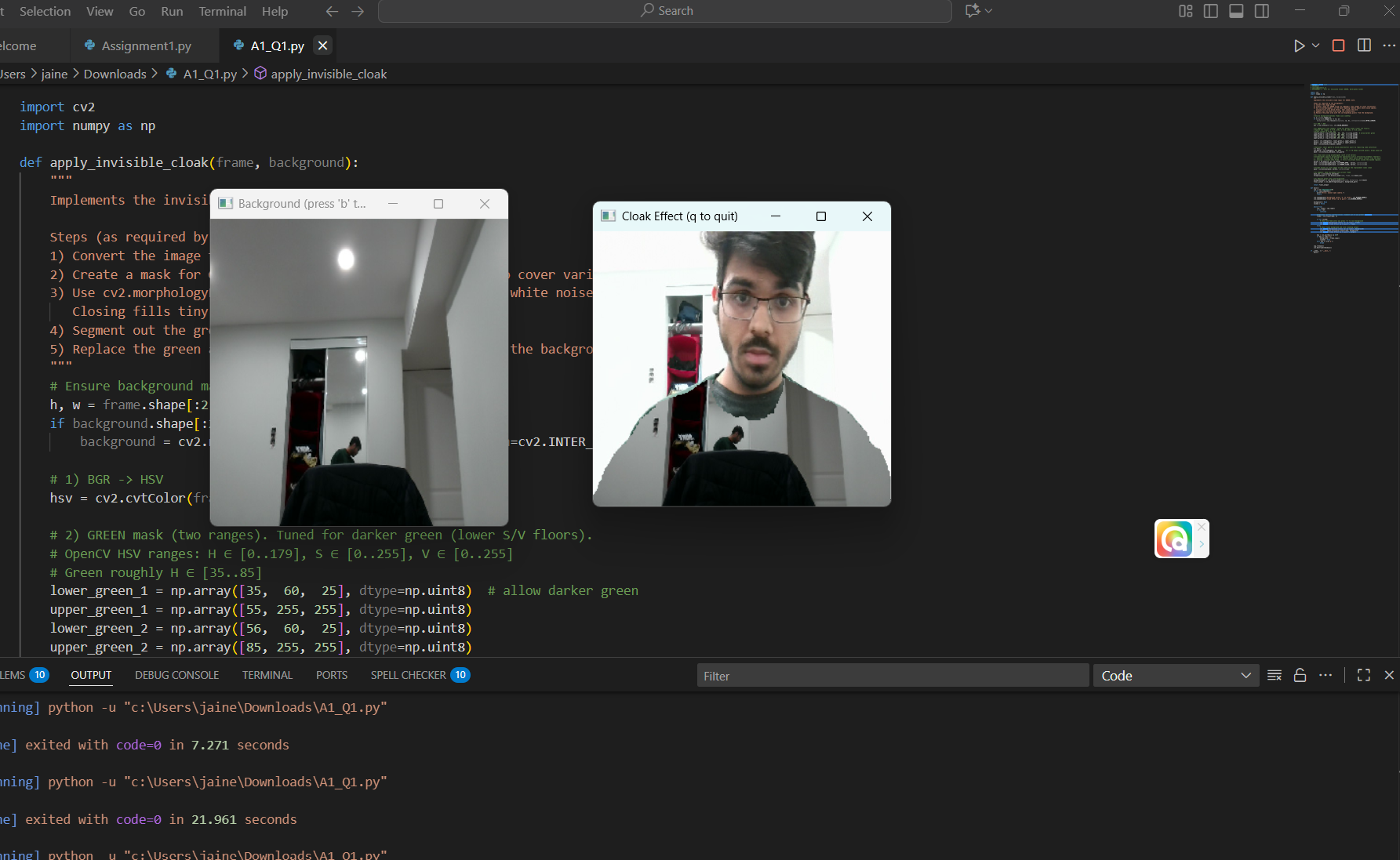


Only background:-

A screen shot of a computer

AI-generated content may be incorrect.

The result:-



## **Part III: A photo booth application**

Design a modular photo editing application in Python using OpenCV and NumPy. The app should allow users to load an image and apply a sequence of processing steps interactively through a menu interface.

Required Functionalities:

* Load an image from file.
* Show the following menu:

==== Mini Photo Editor ====

1. Adjust Brightness

2. Adjust Contrast

3. Convert to Grayscale

4. Add Padding (choose border type)

5. Apply Thresholding (binary or inverse)

6. Blend with Another Image (manual alpha)

7. Undo Last Operation

8. View History of Operations

9. Save and Exit

* Brightness/Contrast
* A cat lying on its back on a carpet

  AI-generated content may be incorrect.Padding: Ask the user to specify the padding size and border type (constant, reflect, replicate, etc.). Additionally, include an option for the user to choose the padding proportion: Square, Rectangle, Custom Ratio (e.g., 4:5)

If the user selects a custom ratio like 4:5, your program should calculate and apply the padding so that the final image respects the chosen aspect ratio, regardless of the original size. The user should also be able to adjust the total padding size, and your code must maintain the proportion accordingly. (add smallest padding at the beginning to make it rectangle, then increase or decrease the padding size of the user wants)

* Thresholding: Let user choose between cv2.THRESH\_BINARY and cv2.THRESH\_BINARY\_INV.
* Blending: Ask for a second image path and alpha (0 to 1).
* Undo Feature: Keep a history stack of image states. Allow the user to revert to the previous state.
* History Log: Keep a list of all actions performed (e.g., "brightness +50", "padded 20px with reflect"). Display the list before exiting or when requested.

Additional Requirements:

* Code must be modular. Each operation must be a separate function.
* Allow the user to apply multiple transformations in a row.
* When exiting, ask whether the user wants to save the final image, and under what filename.
* Use matplotlib to show side-by-side [original | preview] after every transformation.
* Include some of the results in this document.

.

My options: -

Default image: -

A screenshot of a computer

AI-generated content may be incorrect.

1. Choose to change your image Brightness/Contrast

Result:- Brightness changed and contrast too.

A screenshot of a computer

AI-generated content may be incorrect.

You can also see the log for what you changed in the file.

1. Choose if you want to use Padding (Square / Rectangle / Custom Ratio)

Result:- 1:1 applied on the image and now we’ll try to wrap to image with the same image with 1:1 form.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Choose if you want to use Thresholding (Binary / Binary\_Inv)

Result:- image 1 binary threshold applied and image to binary\_inv applied.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Choose for Blending (with second image)

Result:-

The image blended with   
A cartoon of a person with his hands together

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. That is the Undo feature which is used to udo if we mistakenly edit any image.
2. This is a history tab where you can see all the history of the changes committed to an image.

A screenshot of a computer

AI-generated content may be incorrect.

1. Reset option to reset everything how it was to normal.
2. Exit.

## **Part IV:**

[Academic integrity Policy](https://www.senecapolytechnic.ca/about/policies/academic-integrity-policy.html)

I, Jayneel Pratap Jain, declare that I have read and understood the Academic Integrity Policy.

GOOD LUCK!