Computer Vision exercises for "Objects, Classes and C++"

Task 1
Using the OpenCV library, implement a function that:

- opens the image provided as a command line argument using the cv::imread() function;
- creates a window using the cv::namedWindow() function;
- shows the image on the window using the cv::imshow() function;
- executes the visualization and blocks the execution using the cv::waitKey() function.

Task 2
In a new file, expand the software developed in Task 1 by adding:

- a safety check on argc - check that the image is provided as a command-line argument. If argc < 2, print a message that warns the user that an image filename shall be provided;
- a safety check on the image returned by cv::imread() - what happens if the filename is wrong? Check on the cv::imread() documentation and handle such condition properly.

Use the executable to open the grayscale and color images provided and test the safety checks above.

Task 3
In a new file, edit the software in Task 2 by adding instructions for printing the number of channels of the image opened. Also, save the output of cv::waitKey() into a char variable and print it before exiting. Check the OpenCV documentation (or the slides) if you do not know how to get the number of channels. Run the executable on all the images provided.

Task 4
In a new file, edit the software in Task 3 by adding a function that checks if the number of channels of the input image is 3. If so, it sets to 0 the first channel and visualizes the image. Which color is missing? Try other versions that set to 0 the second or the third channel. Which color is missing? What is the color coding used by OpenCV? Try with all the images provided.

Task 5
Edit the software in Task 3 by adding a function that checks if the number of channels of the input image is 3. If so, create three images with the same size of the input image, one channel, containing the values found in the first, second and third channel of the original image. Visualize such images. Try with all the images provided.