RMIT UNIVERSITY

Advanced Programming Techniques
COSC1076 | Semester 1 2022
24 hour Programming Exercise - Sample Question

Assessment Type	24 hour Programming Exercise - Sample Question
Start Date	
Due Date	
Weight	
Submission	.
Learning Outcomes	This assignment contributes to CLOs: 1, 2, 3, 4

General Information

- You must not share your questions or answers. These are individual questions. It is **academic misconduct** to share your questions or work with any individual.
- Please follow the specific instructions for each question that describe where to place your answers in your submission.
- Submission instructions are located in the Appendix on the last page.
- Any code you write must comply with the Course Style Guide.
- You may only use C++ language features and C++ STL elements that have been taught in the course. The reason for this is to limit the scope of what you are expected to answer to the contents of the course and therefore be fair to all students.

Question 1:

Instructions:

- Place all of your code files for this question in a sub-folder called `question1`.

Continued ...

Question 2: Generics & Inheritance (0 marks)

Instructions:

- Place all of your code files for this question in a sub-folder called `question2`.
- The sub-folder `question2` in "starter code" provides a starting point. The starter code contains errors and you should fix those while developing your solution.
- You should not use libraries that are not already included in the starter code.

In this question you will *implement* a set of classes to represent an image and image pixels. Typically, an image is represented as a matrix (2D) of pixels (In this question, you should represent an image as a two-dimensional vector (vector of vector) of pixels). While there are many representations for a pixel, the most common are the RGB-pixels and grey-scale pixels. A grey-scale pixel has only one attribute which represent the intensity of the pixels. An RGB pixel contain 3 attributes which represent the 3 colour channels ('red', 'green' and 'blue').

You should *carefully consider* implementation as this question contains a number of details. You are marked on the *correctness* and *quality* of your implementation.

The starter code for `question2` consists of the following files:

- `main.cpp`: Contains code that create an RGB image and a Grey-scale image and manipulate them. *This file should not be modified.*
- `Pixel.h`: Abstract class to represent general functionality of a pixel.
- `RGBPixel.h`/`RGBPixel.cpp`: Represent a RGB pixel. Derived from base class Pixel.
- `GreyscalePixel.h`/`GreyscalePixel.cpp`: Represent a grey-scale pixel. Derived from base class Pixel.
- `Image.h`/`Image.cpp`: Represent an image. Has to be able to use either grey-scale or RGB pixels.
- `Makefile`

2.1 Pixel class (1 mark)

The class `Pixel` is an abstract class and it should contain functions: `getBrightness`, `operator[]`.

2.2 RGBPixel & GreyscalePixel classes (3 mark)

Both classes should derive from the `Pixel` class. You should also implement appropriate constructors/deconstructors. The brightness of a grey-scale pixel is the intensity of that pixel and the brightness of RGB pixel is to be computed as $(r + b + g)/3$. Here r, g and b are the values of the colour channels red, green and blue respectively. The `operator[]` should return the value of the appropriate channel based on a character input. See the header files in the starter code for more information.

You **should not modify** the private data members in the two classes RGBPixel & GreyscalePixel.

2.3 Image class (2 mark)

This class should store an image as a two-dimensional vector (vector of vector) of pixels. An image could be initialized to have either grey-scale or RGB pixels. You should also implement a `get` and a `set` method to access/modify the pixels in an image given the row and column.

Programming by Contract Paradigm

In the *implementation* of your classes, you must use *Programming by Contract* paradigm. That is, The programmer should specify a "contract" (where applicable) that must be complied with when using a specific functionality of the class.

Continued ...

Question 3:

Instructions:

- Place all of your code files for this question in a sub-folder called `question3`.
- The sub-folder `question3` in "starter code" provides a starting point.
- You should not use libraries that are not already included in the starter code.

Appendix A: Submission Instructions

Combined all of the files for the questions into a single ZIP file called `s123456.zip` (change filename to match your student number). Upload this file to the Canvas assessment module for the `24 Hour Programming exercise - Alternate`.

Finally you **must** also include *this* PDF files of questions in your ZIP.

Appendix B: Late Submission Policy

Ensure that you give yourself sufficient time to prepare and submit your work. The submission is a hard deadline. That is, there will be no leeway on late submissions. The late submission policy is:

- 20% penalty if submitted up to 1 hour late
- 50% penalty if submitted up to 2 hours late
- Grade of 0 if submitted 2 or more hours late