Practical 4 (due 2022-08-26 @ 09:00)

The purpose of this practical is for you to become familiar with inheritance and polymorphism. Create a set of data structures as follows:

- An `Image2D` is an abstract base class which contains a two-dimensional dynamically allocated array of integers. The internal array must be visible to the `Image2D` class and any derived classes created from it. It must have the following:
  - Constructors:
    - A no-args constructor
    - A fully parameterized constructor which takes in a set of three integers representing the number of rows, columns, and the initial value of each integer in the underlying array.
    - A copy constructor
  - A destructor
  - Accessors for the numbers of rows, columns, and individual values in the underlying array.
  - Mutators for changing the underlying values in the array (by way of their row / column value)
  - A pure virtual function called `toString()` which does not have any parameters and returns a string.
- A `PGMImage` **is a** `Image2D` which overrides the `toString()` member function to return a string in the P2 PPM/PGM format (**Note:** a PGM Image is not a colour/PPM image. It does make use of have RGB pixels or a P3 PPM string):
  - https://en.wikipedia.org/wiki/Netpbm#PGM_example
- A `PBMImage` **is a** `Image2D` which overrides the `toString()` member function to return a string in the P1 PPM/PGM format (**Note:** a PBM Image is not a colour/PPM image. It does make use of have RGB pixels or a P3 PPM string):
  - https://en.wikipedia.org/wiki/Netpbm#PBM_example
  - For each value in the underlying array a pixel is translated as follows:
    - If the value is greater than 0, the pixel value in the PBM string must be set to 1.
    - If the value is 0, the pixel value in the PBM string must also be set to 0.
- The derived classes must make use of the base class's constructor (constructor chaining)
- Create a `printImage` function in `main.cpp` that takes an `Image2D` reference and outputs the contents of the image using its `toString` method.
- You will learn about abstract base classes and pure virtual functions in the coming week when we cover polymorphism in class.
- Create a main function which takes in command line arguments which control whether the system outputs a random `PGMImage` or a random `PBMImage` (0 for `PGMImage` and 1 for `PBMImage`) of a given size (`rows` and `cols`).
- **Bonus**: upload PNG image versions of an example of your output using a program like GIMP (https://www.gimp.org/downloads/)

Mark sheet		
	Design	10
	Image2D abstract base class	10
	Appropriate member visibility	10
	toString pure virtual member function	10
	Derived classes	10
	Constructor chaining	10
	Overriden toString in PGMImage	10
	Overriden toString in PBMImage	10
	printImage function that calls toString polymorphically	10
	main function uses command line arguments to create the correct instance	10
	Bonus	(10)

	Total	/100