# Final exam

CSE241

June 16, 2022

You must submit all code to the final exam assignment dropbox on Blackboard before the deadline. Commenting is not required.

## 1 Shapes

Total 19 marks.

### 1.1 Provided code

Some code is already provided for you on Blackboard. Please download it to get started. The `shape` abstract class is provided, as well as the `read_one_shape` function.

A `Makefile` is provided, as well, which you may modify, if you wish. You are not required to use the Makefile and it is not worth any marks.

Your program will ultimately read in shapes from standard input and store those shapes in a vector. You will then print out all the shapes, along with their area. Finally, you will print out the largest shape and the second-largest shape.

### 1.2 Completing the class hierarchy (9 marks)

You must write two classes, `circle` and `rectangle`. Both must inherit from `shape` (1 mark).

A `circle` consists of only a radius, which is a `double`. Make a constructor for it with one parameter (the radius) (1 mark). Implement the `area` method for it (1 mark). Implement the `print` method, which must print the circle to the given `ostream` in the format of `circle(`*`radius`*`)` (e.g., `circle(3.0000)`) (2 marks).

A `rectangle` consists of a width and a height, both `double`s. Make a constructor with two parameters (the width and the height). Implement the `area` method (1 mark). Implement the `print` method, which must print in the format of `rectangle(`*`width`*`,height)` (1 mark).

You must also implement the following function:

```
std::ostream& operator<<(std::ostream&, shape const &);
```

You must use the `print` method to implement that function. (2 marks)

### 1.3 read_all_shapes (4 marks)

The `read_one_shape` function provided to you reads in only one shape, which is allocated on the heap. It returns a pointer to that shape if the line could be read successfully, and returns `nullptr` otherwise.

You must write a `read_all_shapes` function in `main.cpp`:

```
vector<shape *> get_all_shapes(istream& in);
```

It must continue reading in shapes using `read_one_shape` and adding them to a vector until EOF (end-of-file) is reached, at which point it will stop and return the vector (2 marks).

If `read_one_shape` returns `nullptr`, it must *not* add to that to the vector (1 mark). If the `istream` object `in` enters an error state, you must put it back into a good state, ignore to the end of the current line, and continue reading. (1 mark)

### 1.4 main (6 marks)

Your `main` function must read in all shapes using the `read_in_shapes` function (1 mark). It must then print out the area of each shape, and the shape itself, for all shapes, using 10 characters to print the area, with 4 digits after the decimal point (2 marks).

Then, it must print the largest shape (1 mark). Then, it must print the second-largest shape (1 mark).

There should be no memory leaks or memory errors (1 mark).

### 1.5 Final remarks

If your code does not produce the correct result, or crashes, or does not compile, please do not panic. We understand that you are working under time constraints and may not be able to complete perfect code. Unlike your assignments, we will not place a strong emphasis on having everything working. Each function will be marked individually, and not many students will get it working perfectly.

You do *not* receive any marks specifically for working with the test data provided. That is for your testing purposes, only.

You may submit multiple times. Your last submission before the deadline will be taken as your final submission.