PROG2100: Object Oriented Programming in C++

the Final Exam

Your name: ____________________

Score:__________________________

This is an open books exam.
Real value: 20% of the final grade.

Part A. Fundamentals [60%]

Review the following class *definitions:*

```
class Point
{
public:
    Point(double x = 0.0, double y = 0.0) : x_(x), y_(y) {};
    virtual ~Point(){};
    double x() const { return x_; }
    double y() const { return y_; }
private:
    double x_;
    double y_;
};


class Shape
{
public:
    Shape(){};
    virtual ~Shape(){};
    virtual double area() = 0;
    virtual bool isExternal(const Point&) = 0;
    // isExternal verifies whether or not a given Point
    // lies outside the shape
};
```

1. [30%]
   Define class **Circle** that would utilize both of the above definitions.
   Draw a rough class diagram showing relationships between Point, Shape and Circle in the space provided below.

   Your definition must be sufficient to make the following work:

```
Circle foo(5.0);                 // Creates a circle with
                                 // radius 5.0
foo.isExternal(Point(4,4));      // Returns true

Circle bar(5.0, Point(4,4));     // Creates a circle with
                                 // radius 5.0 and centre (4,4)
bar.isExternal(Point(4,4));      // Returns false
```

Class diagram:

2. [30%]
Write a short mainline that creates 3 Circle objects, reading the coordinates of their centres and their radii from a *binary* file called 3C9D; the 3C9D file (*supplied*) contains 9 double numbers. Then: print the areas of your circles.

Part B. Program Design & C++ intricacies [60%]

1. [30%]
Design Universal Player application that can play *anything playable*.
The client expects these 2 methods: play(), and stop(); but what/how is being played is not fully decided yet *(and, maybe, will never be)*: we could play music or video files in various formats; or chess; or roulette (*maybe the Russian one ;*).

Define the Playable abstract class; and a client class with 2 methods: **void** setUP(Playable *) and **void** playIt(**int**=1) – the latter executes Universal Player's play a requested number of times.

2. [30%]
Define a PlayTxt4us class *(not to be confused with duck-billed platypus ;)* derived from Playable that allows for "playing" a text file specified in its constructor, one line at a time, waiting 1 second between lines[1]. Depending on your approach, the stop method may close the file, or do nothing; but, as we do not use threads, do not attempt any "stop in the middle" action - I do not expect that.
Finally, implement the main() that would create suitable objects and make the client object play the Book.txt *(included in the resources provided)* twice.

Remember that the quality of your design is as important as having working implementation.

How to deliver your answers:
Place all your source/text files in a directory Final2018, zip it, and submit on BrightSpace/D2L.

Merry Christmas !

[1] Use simple sleep *(may require inclusion of the unistd.h header).*