The `final` Keyword

1. We discussed keywords such as `static`, `public` etc. in the videos. We did NOT discuss the `final` keyword. Google it. Find out what the `final` keyword does in JAVA.

Then write down what it means to have:
a) A `final` variable
b) A `final` class
c) A `final` method

Encapsulation

2. Create a class and use proper encapsulation techniques.

The class can be called Printer. It will simulate a real computer printer.

It should have attributes such as ink level, number of pages printed, number of pages left in the printer (in case you need to reload pages) and also whether it is a duplex printer (if it can print on both sides of a page).

Add methods to fill up the cartridge with ink (up to a maximum of 100%) and another method to simulate printing a page (which should increase the number of pages printed and decrease number of paper left).

Important: Decide on the scope of variables and methods (`public`, `private` etc.), whether to use constructors or not, and anything else you think you need to add that is interesting or functional for a printer.

Make sure your setter methods are helping you keep control on how attributes can be changed or how attribute values can be set.
(Chapter 3 - Part 9)

Make sure your constructor is not breaking encapsulation.
(Chapter 3 - Part 13)

Make sure your getter and setter methods are not breaking encapsulation.
(Chapter 3 - Parts 10,11,12,14,15)

Bonus: In a real printer, documents get added to a queue when we try to print them. Documents that went to the printer first get printed first.

Keeping the above in mind, create an attribute named printingQueue. This should simulate adding and removing documents to and from the printing queue. This can be a non-primitive data type where you can add and remove queued documents for printing (*Be creative with this. Play around with different ways of designing the printing queue. Don't just think of using arrays. Google for other data structures that might be more appropriate. Here a* list *or* queue *will be good to use. Just google how they work and how to add and remove elements from them. They are easy to understand so don't be scared*☺️*. Remember to have fun when designing this*).

Here you would want to have other methods like `addToQueue()` when a document is sent to the printer for printing, and `removeFromQueue()` for simulating that a document has finished printing. Both of these methods should control the `printingQueue` attribute. **Use proper encapsulation techniques.**