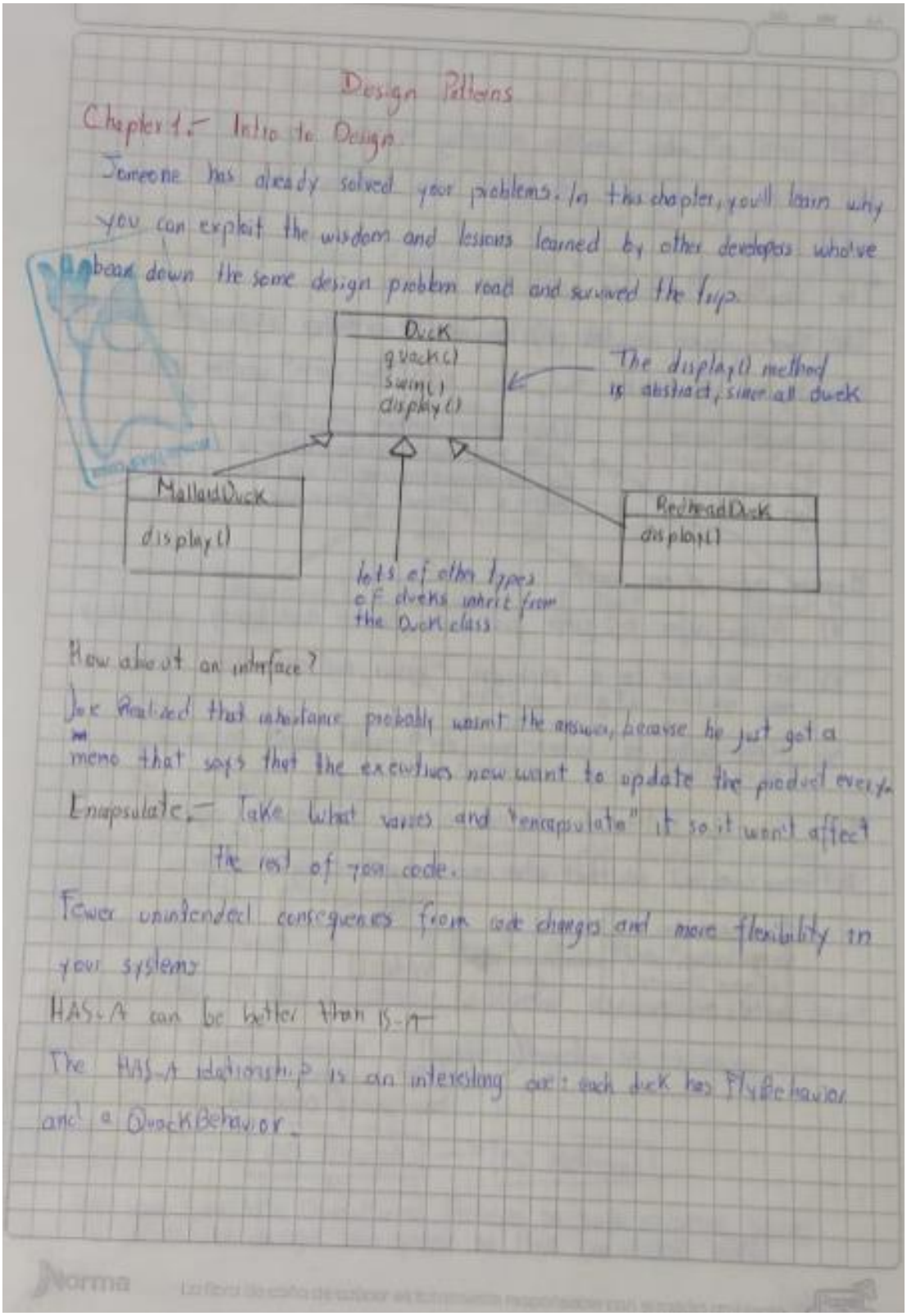
Solid Principles

Name: Josue Marin



Design Patterns

Chapter 1:- Intro to Design.

Someone has already solved your problems. In this chapter, you'll learn why you can exploit the wisdom and lessons learned by other developers who've been down the same design problem road and survived the trip.

Duck
quack()
swim()
display()

The display() method is abstract, since all ducks

MallardDuck
display()

RedheadDuck
display()

lots of other types of ducks inherit from the Duck class

How about an interface?

Joe Realized that inheritance probably wasn't the answer, because he just got a memo that says that the executives now want to update the product every

Encapsulate:- Take what varies and "encapsulate" it so it won't affect the rest of your code.

Fewer unintended consequences from code changes and more flexibility in your systems

HAS-A can be better than IS-A

The HAS-A relationship is an interesting one: each duck has FlyBehavior and a QuackBehavior.