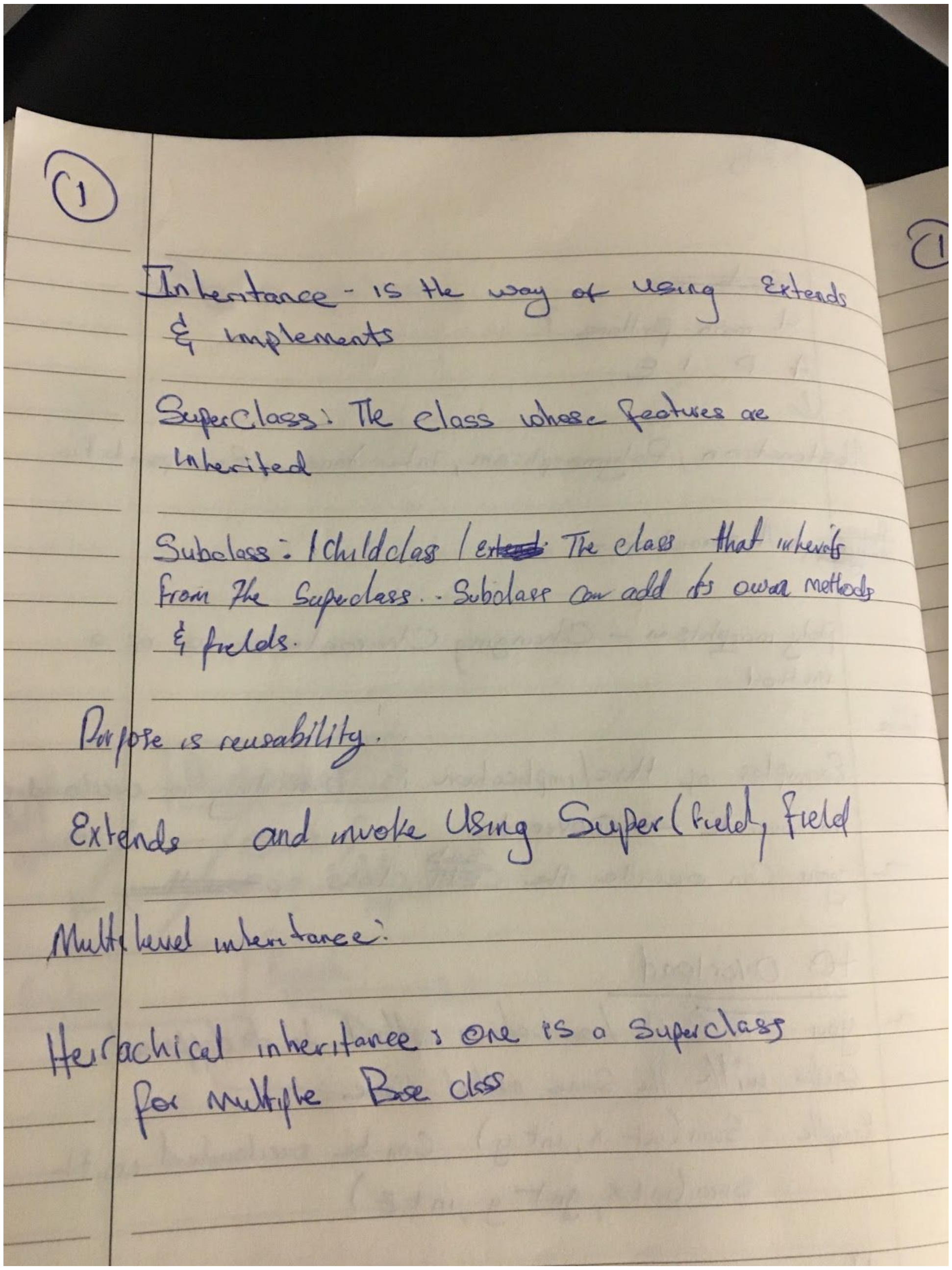
1. What are the 4 main pillars of Object Oriented Programing and give description of each and how they are applied



1

Inheritance - is the way of using Extends & implements

SuperClass: The class whose features are inherited

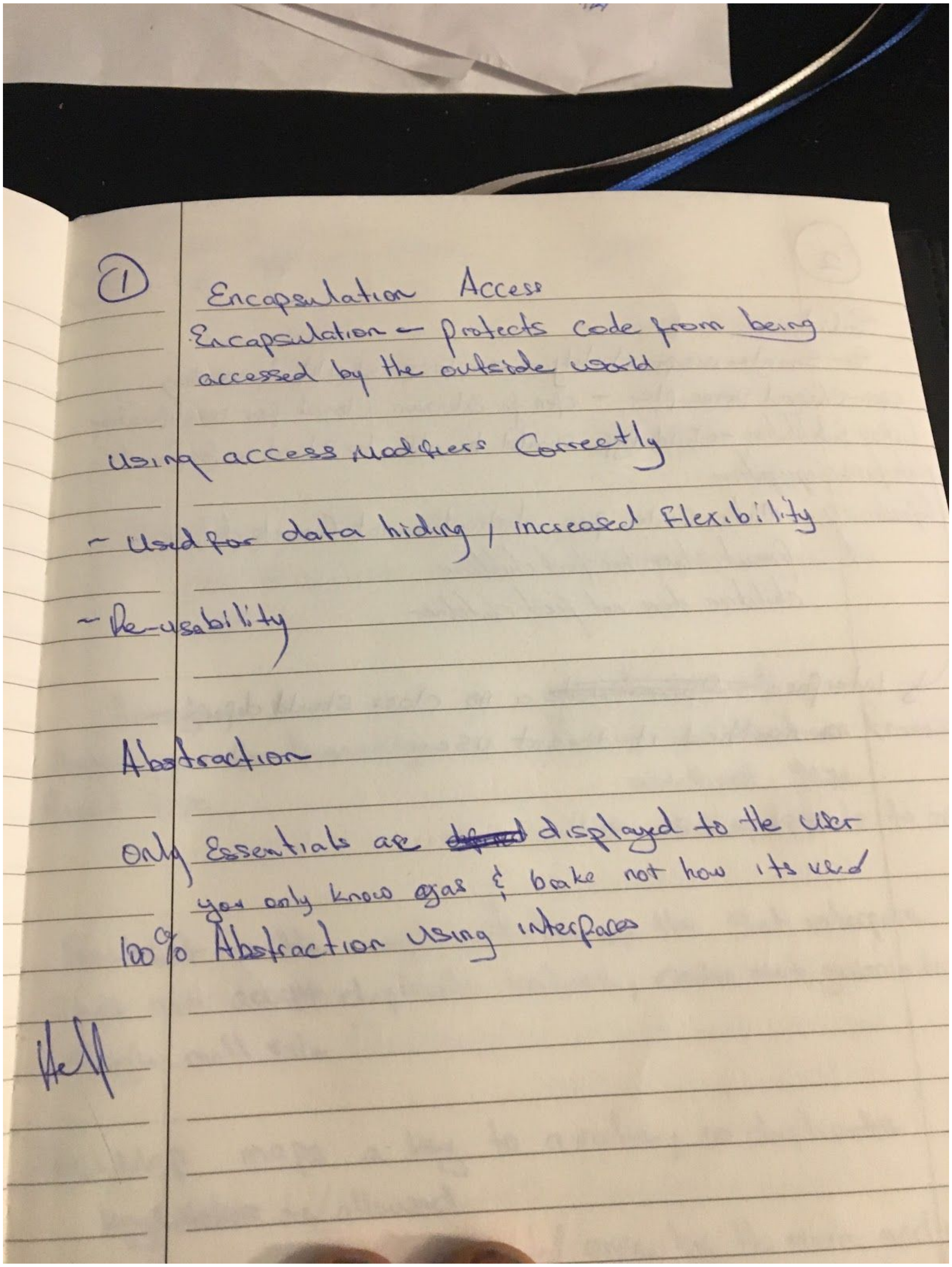
Subclass: / Childclas / ext~~end~~: The class that inherits from the Superclass. - Subclass can add its own methods & fields.

Purpose is reusability.

Extends and invoke Using Super(field, field

Multilevel inheritance:

Heirachical inheritance: One is a Superclass for multiple Base class



① Encapsulation Access

Encapsulation - protects code from being accessed by the outside world

using access modifiers Correctly

- Used for data hiding, increased Flexibility

- Re-usability

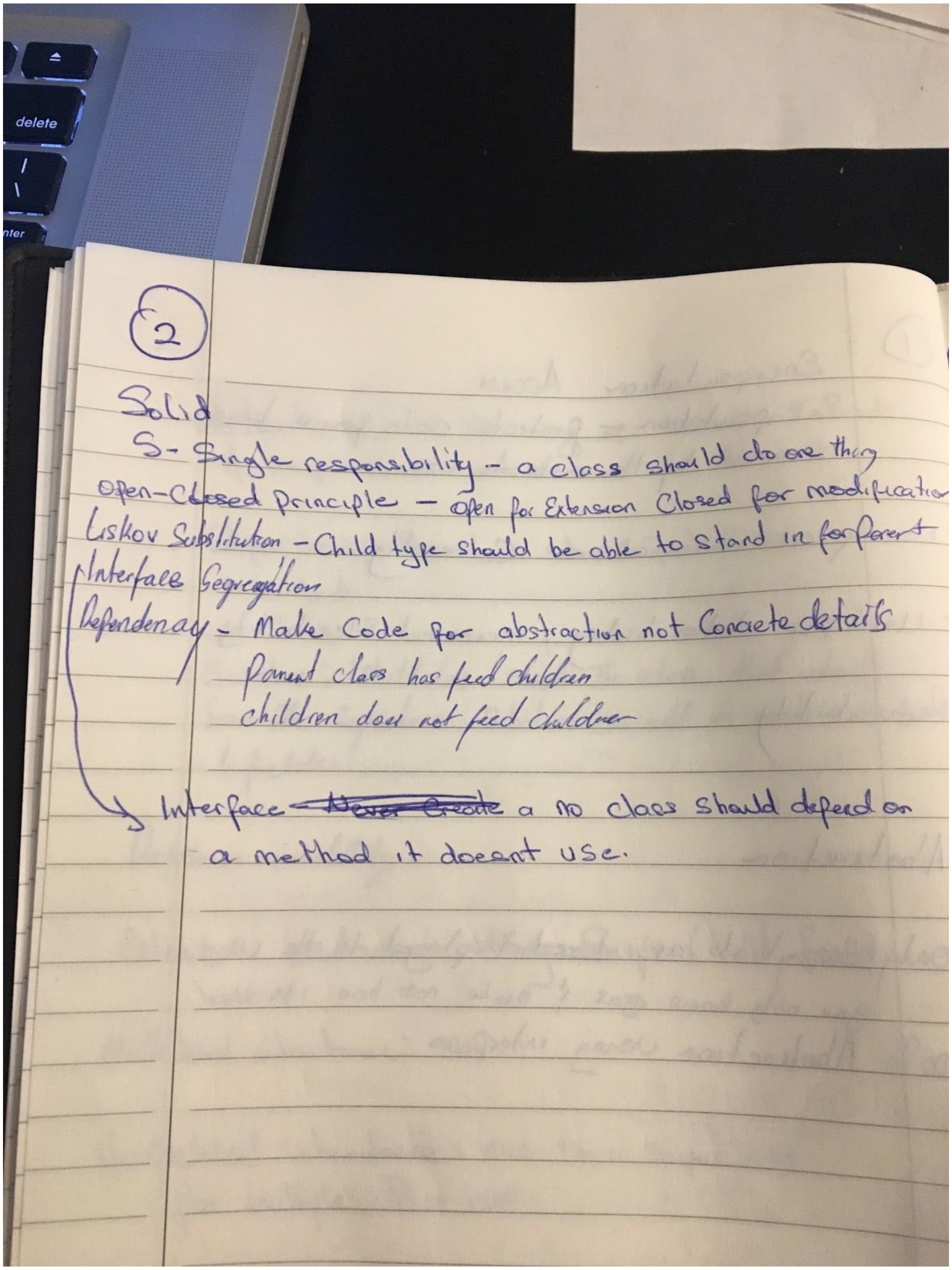
Abstraction

Only Essentials are displayed to the user

you only know gas & brake not how its used

100% Abstraction using interfaces

2. What are SOLID programming principles and what does each section detail?



2

Solid

S- Single responsibility - a class should do one thing

Open-Closed Principle - open for Extension Closed for modification

Liskov Substitution - Child type should be able to stand in for parent

Interface Segregation

Dependency - Make Code for abstraction not Concrete details

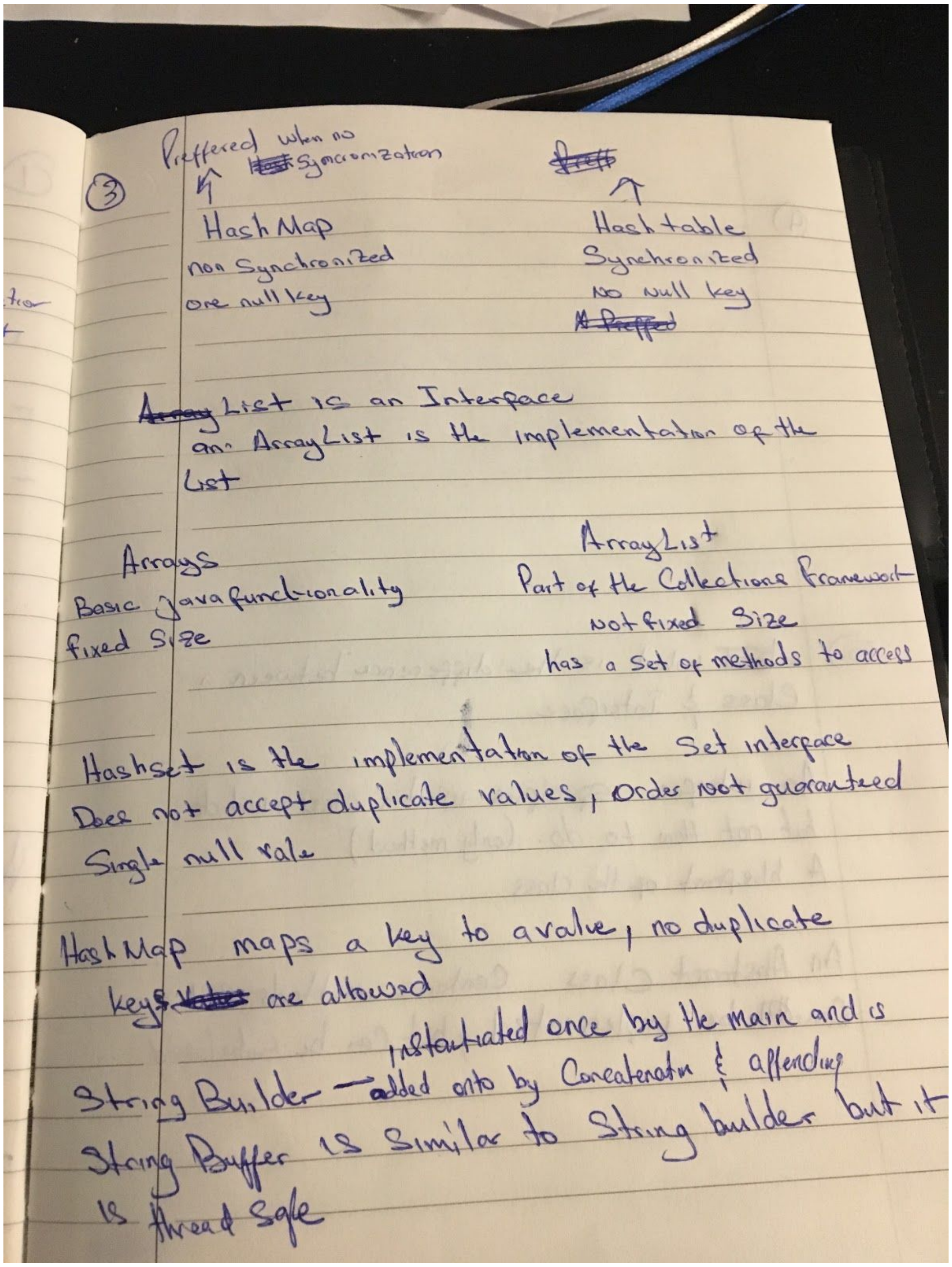
Parent class has feed children

children does not feed children

Interface ~~Never Create~~ a no class should depend on a method it doesent use.

3. What are the differences of the Following:

- HashMap vs HashTable
- ArrayList vs List
- Array vs ArrayList
- HashSet vs HashMap
- StringBuilder vs StringBuffer



③

Preffered when no ~~Hash~~ Syncronization ~~Preff~~

HashMap	Hashtable
non Synchronized	Synchronized
one null key	No null key
	~~A Preffed~~

~~Array~~ List is an Interface
an ArrayList is the implementation of the List

Arrays	ArrayList
Basic Java functionality	Part of the Collections framework
fixed Size	not fixed Size
	has a set of methods to access

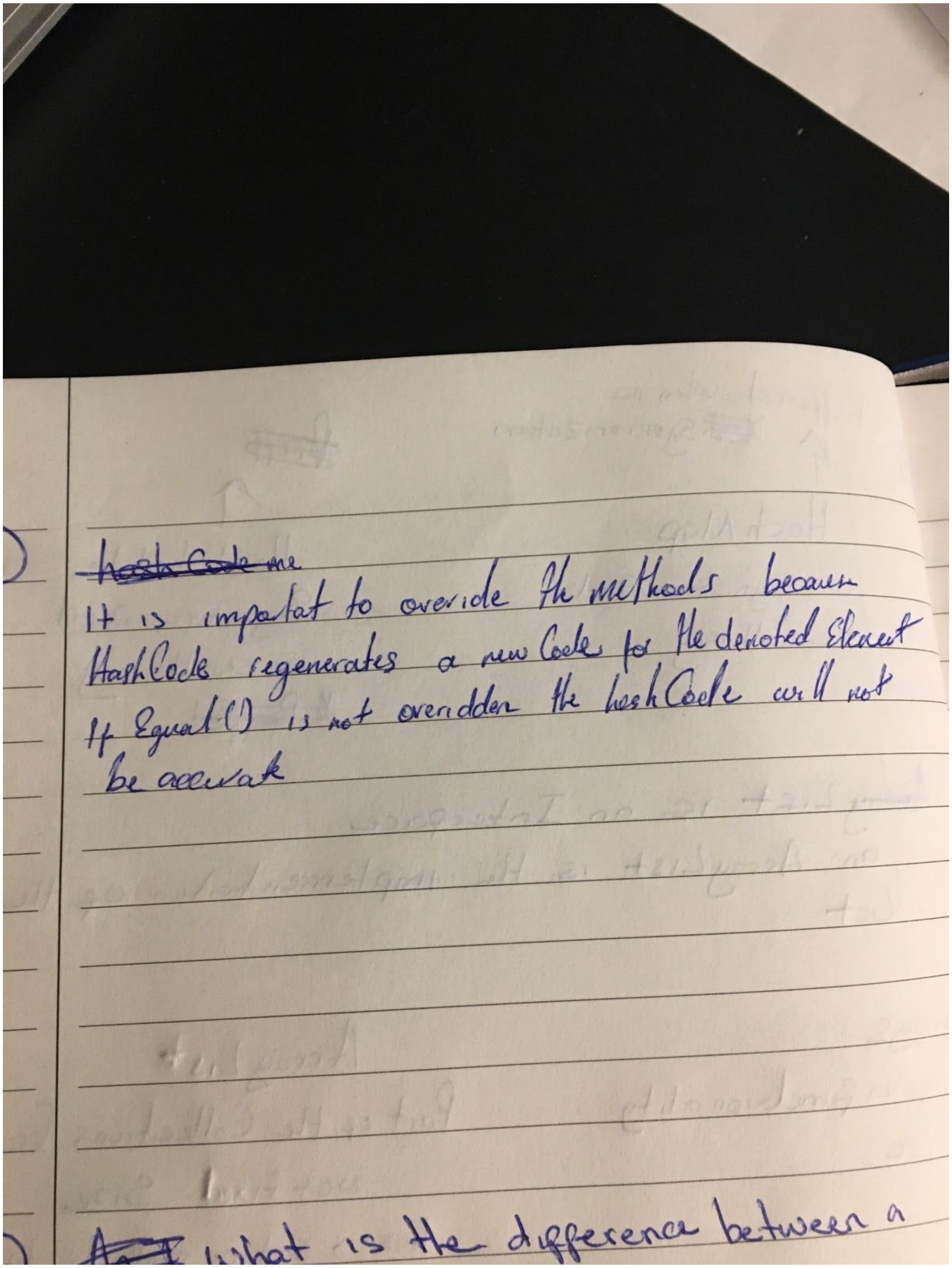
Hashset is the implementation of the Set interface
Does not accept duplicate values, Order not guaranteed
Single null vale

HashMap maps a key to a value, no duplicate
keys ~~Values~~ are allowed

String Builder – instantiated once by the main and is added onto by Concatenation & appending

String Buffer is Similar to String builder but it is thread safe

4. Why is it important to override the equals and hashCode methods for Java objects?



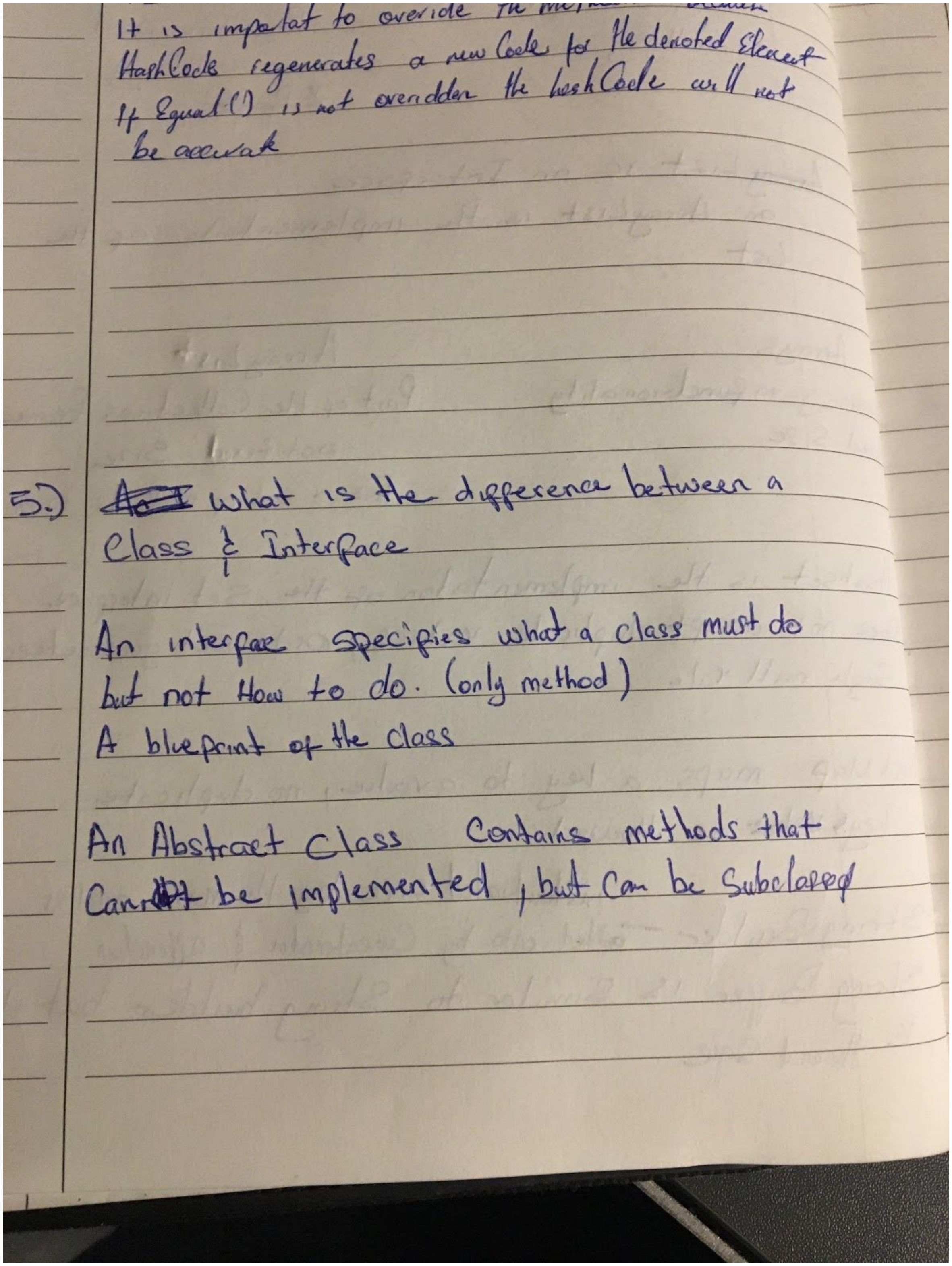
~~hash Code~~ me

It is impotat to overide th methods because HashCode regenerates a new Code for the denoted Element If Equal() is not overidden the hashCode will not be accurate

~~A I~~ what is the difference between a

5.

5.)What is the difference in an Abstract Class and an Interface?



It is impotat to overide the

HashCode regenerates a new Code for the denoted Element

If Equal() is not overidden the hashCode will not be accurak

5.) ~~An I~~ what is the difference between a Class & Interface

An interface specifies what a class must do but not How to do. (only method)

A blueprint of the class

An Abstract Class Contains methods that Cann~~ot~~ be implemented, but Can be Subclased