Noah Parker
Chapter 7 Study Sheet

A class should represent a single concept. Example would be class MilesPerHour{ double miles , hours , mph; mph = miles/hours} Class MilesPerGallon{ double StartMiles,EndMiles,GallonsUsed; (EndMiles-StartMiles)/GallonsUsed } in this example we have classes that do very specific things such as the first class being used to calculate miles per hour and the second class being miles per gallon. These classes could be used to calculate how long it takes to go somewhere and then how many gallons you will use going there roughly.

Separating accessors and mutators. A mutator method changes the state of an object and an accessor method asks an object to compute a result without changing the state. a class that only had accessor and not a mutator are called immutable class an example of this would be

```
class money {
  private final int
dollars;

  public money(int i) {
    dollars = i;
  }

  public int getDollars()
{
    return dollars;
  }
}
```

←The class has to have the final in there so that child classes are not created. This class also does not have any of its members exposed. This only has a getter and not a setter so it will get dollars but cannot set dollars. Immutable classes are useful because they express in your design that you can't change this.

When working on certain programs you might need to keep track of a value that can go up or down such as number of cakes left in a bakery. When you bake a cake the number of cakes in your bakery's possession goes up but as you sell a cake the number in your possession goes down. A good example be the table to the right →

When a cake is purchased, it takes the cake count down with cakeCount- -
When a cake is baked it adds the cake count up with cakeCount++
This is so we can keep an inventory count of cake made and sold.
At the end of the day when we need to throw away all the cakes or they are all sold we use the Clear method to reset the values for a new day. This is assuming we sell all the cakes or throw them away if we didn't, we would implement a better way to do this.

```
private int cakeCount;

public void
recordPurchaseOfCake(double amount)
{
 purchase = purchase - amount;
 cakeCount--;
}
public void recordBakingOfCake(double
amount)
{
 baked = baked + amount.
 cakeCount++;
}
public void clear()
{
 purchase = 0;
 cakeCount = 0;
}
```

The static variable can be used to refer to the common property of all objects for example, the company name of employees, college name of students, etc. Static variables are also known as Class Variables. Unlike non-static variables, such variables can be accessed directly in static and non-static methods. Static Methods can access class variables without using object of the class. In non-static methods and non-static variables can only be accessed using objects. Static methods can be accessed directly in static and non-static methods.

Problem solving solve a simpler problem first. When solving a problem with code we need to be able to break it down into a smaller problem. For example, say we need to bake a cake the way we would break this down with java code would be

```
public void startoven
public int getButter
public int getEgg
public int getFlour
public int getPan
public int getBowl
public void add (int egg , int flour,int bowl)
public void mix (add)
public void addPan (int butter , int pan)
public void addAll (addPan , mix)
public void putinOven(addAll)
```

This is a very crude way of coding it but you can see you have to have a lot of different methods which could have classes. At the premise of the idea, you will be breaking it down to very easy instructions that can be followed and as simple as possible so when you add the butter to pan that's separate and done then when you add the flour and egg to the bowl you will have to do the method mix to mix it then add that mix to the pan. So something that people know how to do easily can be broken down into many very simple steps.

Importing packages when having to take user input you need to the scanner class which isn't in java by default you must import the utilities package which has the class scanner using in code. import java.util.Scanner; this calls on the scanner class specifically but when you can also call import java.util.*; which will call on all the classes in utilities package.