Data Structure Notes

Ch1 Recursion

- Two of the most used methods in coding is iteration and recursion , the latter is more powerful due to the fact that it can be used in situations that the number of loops needed is unknown and its more versatile than iteration .

- By using recursion we can break a problem into smaller bits which as the saying goes " divide and conquer " , a recursion calls on its own function until it can fulfil the designated purpose .

- Understanding recursions are harding than iteration , the best way would be using a method called box trace , each box would mean an activation of the function , and a recursive function would keep on activating itself until it complies with the ending requirements .

- Simple recession functions include :

1.Factoiral
Calling on itself to return the value of the factorial of the previous number until it reaches 1
2.GCD (greatest common divisor)
3.Writing an array backwards

- Josephus Problem
  n is the number of people
  k is the number of passes

```
int josephus(int n, int k)
{
    if (n == 1)
        return 1;
    else
        return (josephus(n - 1, k) + k - 1) % n + 1;
}
```

Ch2 Date Abstraction

- A class is the main thing of OOP (object oriented programming)

- A class mainly consist of two things data and actions , its more productive and easier for others to understand , the data includes data type for example a dog class would include breed , weight , color , fur type etc , the action would include eating , barking , playing catch , peeing etc .

- The difference between using class and not using is the we can have different breeds of dogs do the same thing (using the same function) , if we don't use class we would have to write different functions for each breed of dog and would be way harder and complicated to update after the initial coding is done , plus its just easier to perform on datas .

- An ADT (abstract data type) is composed of a collection of data and a set of operations . It only defines the operation contract but doesn't specify how to perform the operation .

Ch3 Linked List

- A linked list is like an array but to delete or insert a new thing in the middle of an array requires shifting nearly the whole array and its just not effective and sometimes impossible a link list breaks the array into little things chained together by a pointer and to delete or insert , you simply have to disconnect one or two chains .
- Example :
  If I were to insert 'b' into the chain a c d e f g …… , without the link list I would have to move every letter after 'c' a block back but since everything is chained together in a link list all I have to do is chain 'b' to 'a' and 'c' to 'b' and 'd' would still be chained to 'c' thus more effective and easy .
- The "chains" are pointers in programing and one of the most important thing to do would be remembering to delete unused space , done wrong it will lead to a memory leak and would stack up and ultimately crashing the computer .