**Oppgave1:**

**1.a)** Skriv for deg selv ned noen positive og negative sider ved array

**1b)** Skriv for deg selv ned noen positive og negative sider ved ArrayList

**1c)** Diskuter dette i små grupper

**1d)** Ta dette i fellesskap

**Oppgave 2:**

Basert på Noden vi programmerte sammen:

class Node {

 Node neste; //= null hvis ingen neste

 String verdi;

public Node(String verdi){

 this.verdi = verdi;

 }

 public void settNeste(Node n){

 neste = n;

 }

 public Node hentNeste(){

 return neste;

 }

 public String hentData(){

 return verdi;

 }

}

**2.a)** Skriv et hovedprogram, hvor du oppretter Noder og lenker dem sammen slik at du får følgende datastruktur:



**2.b)** Skriv deretter en while-løkke som skriver ut alle verdiene til Nodene.

**2.c):**

Du skal nå endre “Node-klassen”, bytt ut det som trengs for å gjøre den generisk. Skriv også om hovedprogrammet slik at den passer til den nye “Node-klassen”.

**Oppgave 3:**

Du har fått interfacet:

public interface Par<K, V> {
    public K hentNokkel();
    public V hentVerdi();
}

**3.a)** Skriv klassen “OrdnetPar”, denne skal implementere interfacet “Par”.

**3.b)** Et ordnet par har to instansvariabler, en nøkkel og en verdi, begge er generiske typer. Skriv konstruktøren for OrdnetPar, her settes instansvariablene.

**3.c)** Implementer deretter metodene som kreves av interfacet.

**Løsning:**

public class OrdnetPar<K, V> implements Par<K, V> {

    private K nokkel;
    private V verdi;

    public OrdnetPar(K nokkel, V verdi) {
 this.nokkel = nokkel;
 this.verdi = verdi;
    }

    public K hentNokkel() { return nokkel; }
    public V hentVerdi() { return verdi; }
}

**Oppgave 4**



Tegn datastrukturtegningen etter følgende operasjoner:

**4.a)** Legg til  stringen “!” bakerst i lista

**4.b)** Legg til stringen “planter” på indeks 3

**4.c)** Endre stringen på indeks 0 til å bli “Du”

**4.d)** Fjern elementet på indeks 2

**4.e)** Hvis du hadde brukt en lignende while-loop som i oppgave **2.b**, hva hadde da blitt skrevet ut ??