# Tråder 2

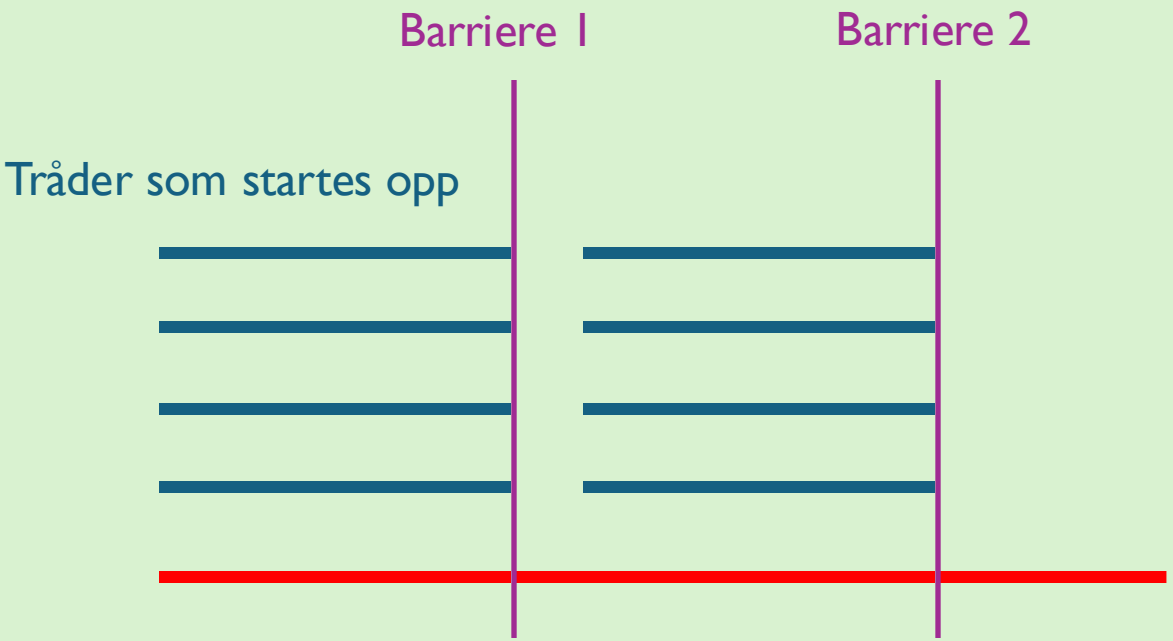
agneshb@uio.no

# I dag

- Fortsetter med tråder
- Barrierer
- Legger på barrierer på forrige ukes bankkonto-eksempel

# Barrierer

- Barrierer er synkroniserings-mekanismer som lar tråder møtes ved ett bestemt punkt i programmet.
- Sikrer at tråder ikke går videre før alle har nådd barrieren.
- Kan derfor kontrollere rekkefølgen av operasjoner på en trygg måte



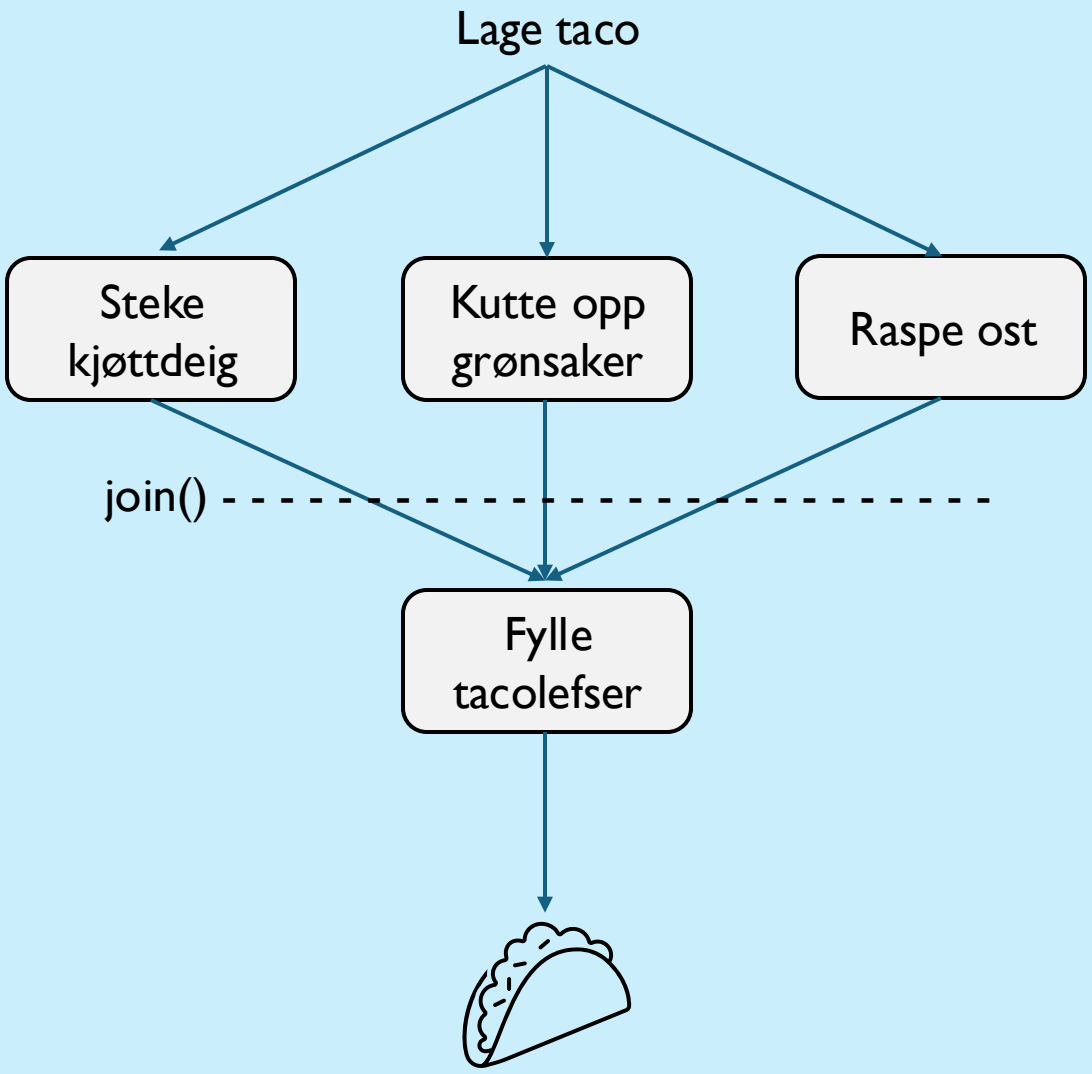

Maintråd med barrierer – maintråden fortsetter ikke videre før trådene har kommet til barriere.

# Barrierer og monitorer

- Barrierer kan brukes i kombinasjon med monitorering.
- To former for synkronisering samtidig;
  - én i form av barrierer og
  - én i form av kritiske regioner med låsing i en monitor.
- Nyttig der tråder deler data samtidig som en eller flere tråder må vente på andre tråder underveis i programmet.

# .join()

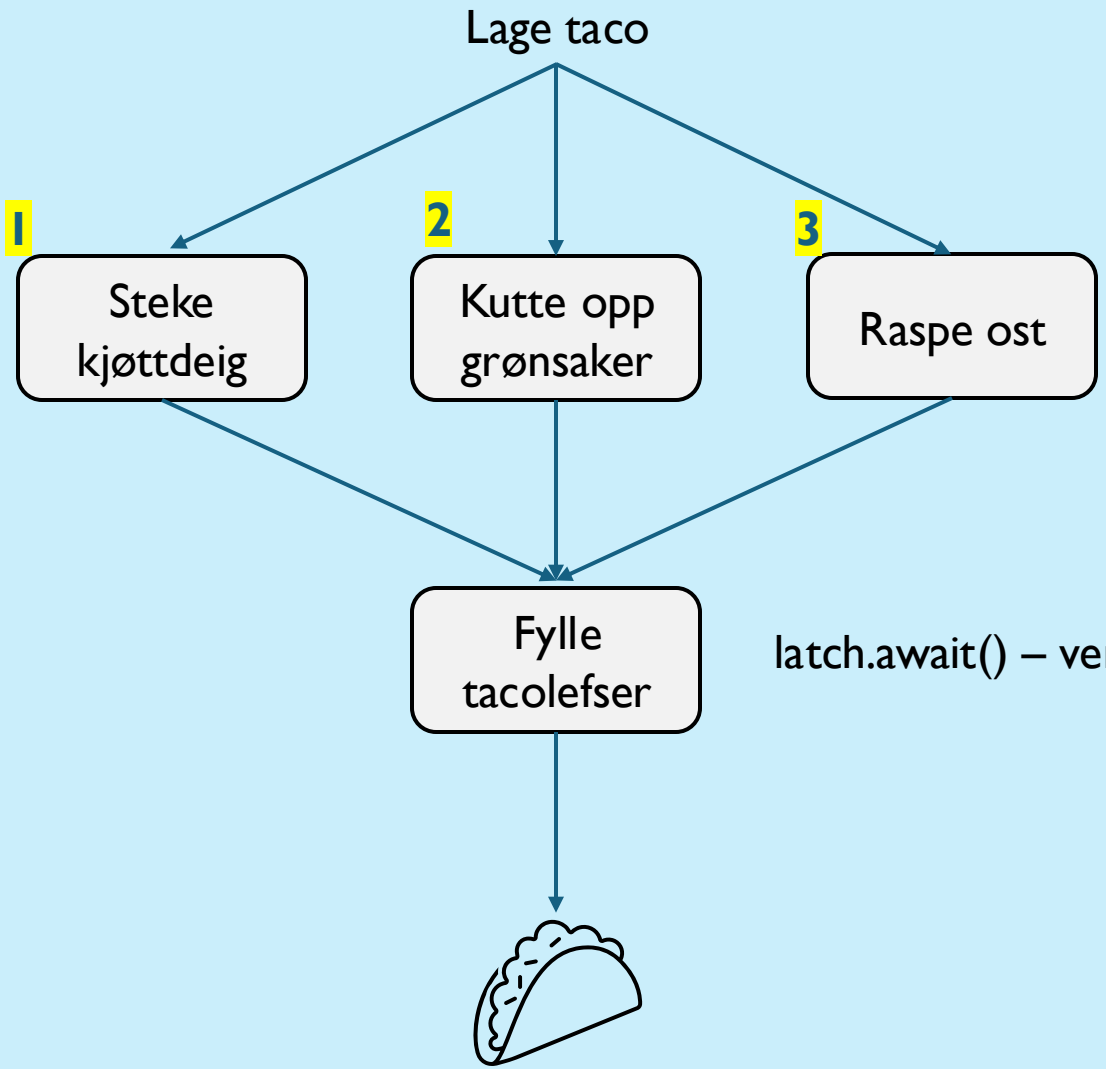
- Tacolaging:
  - Deloppgaver som må gjøres før tacolefsen kan fylles.
- Kaller .join() på alle trådene main-tråden skal vente på.
- Barrieren sikrer at alle ingredienser er klare til tacoen serveres.
- Forhindrer race condition
  - for eksempel at noen starter å dandere sin taco mens kjøttet fortsatt er rått eller grønnsakene ikke er kuttet opp.



# CountDownLatch

- Som en dørstokk som flere tråder må komme seg over.
- Ved opprettelse av en CountDownLatch, sier du hvor mange tråder (eller hendelser) du venter på.
- Hver tråd sier i fra når den er ferdig, og CountDownLatch teller ned. Når nedtellingen har nådd null, får trådene som venter lov til å fortsette.

# CountDownLatch




Her har vi 3 oppgaver som må skje før vi kan fylle tacolefsen.

latch = CountDownLatch(3)

latch.countDown()

0

latch.await() – venter til teller i til latch = 0

# .join() fremgangsmåte

I main-tråden:

- Lager en passende beholder som skal holde på tråder.
- Oppretter instanser av tråder og legger de i beholderen.
- Lager en for-løkke som starter tråder.
- Lager en ny for-løkke som kaller join på hver av trådene
- Nå har alle trådene ferdige og main-tråden kan fortsette ut i fra det.

```java
public static void main(String[] args) throws InterruptedException {
    Bankkonto bankkonto = new Bankkonto();

    ArrayList<Thread> settInnTraader = new ArrayList<>();
    for (int i = 0; i < 5; i++) {
        settInnTraader.add(new Thread(new SettInnTraad(bankkonto)));
    }
    for (Thread t : settInnTraader) {
        t.start();
    }

    for (Thread t : settInnTraader) {
        t.join();
    }

    System.out.println("Saldo etter innsetting: "+ bankkonto.saldo);
```

# Jobbe med oppgaver 

Seminaroppgaver på semestersiden → grupper → gruppe 3 → uke 10