Elementære programmerings**begreper** og **-historie**

1. Abstraksjon = intuisjon (2 uker)

Turing maskin → von Neumann maskin → assembler språk → høytnivåspråk...

2. Imperativ programmering (ref. til historie, Fortran...)

1. implementasjonsmodell: lager + tilstandsoverganger: → tilstand
2. variable, iterasjon
3. parameter passing: call-by-name, call-by-value, ...
4. fordeler/ulemper/hvor brukt

3. Funksjonell programmering (ref. til historie, LISP...)

1. implementasjonsmodell: evaluering av matematiske likninger → en verdi
2. inside-out, ingen side-effekter, rekursive likninger
3. lazy parametre, side-effekter, ...
4. høyre ordens funksjoner
5. fordeler/ulemper/hvor brukt

4. Deklarativ programmering (ref. til historie, Prolog...)

1. implementasjonsmodell: beregning = bevisføring → Ja/Nei/relasjon
2. pattern-matching, unifikasjon, resolusjon
3. Data-base språk
4. fordeler/ulemper/hvor brukt

- pensum: **samling artikler – lesetrening**
- oppgaver: programmering uten datamaskin, +++
- obligatoriske oppgaver: **skrivetrening** (historie av et/noen begrep; sammenheng mellom...)

? Høytnivåstrukturering

1. Typing og typebegrep
2. OO, Moduler, ADT
3. parametrisering (generics....)