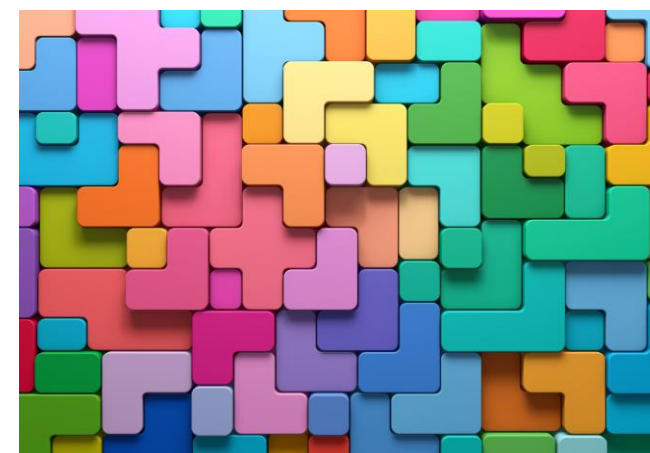


LOGGING OG DEBUGGING

FORELESNING 17

ONSDAG 23/10



(bilder generert av bing image creator)



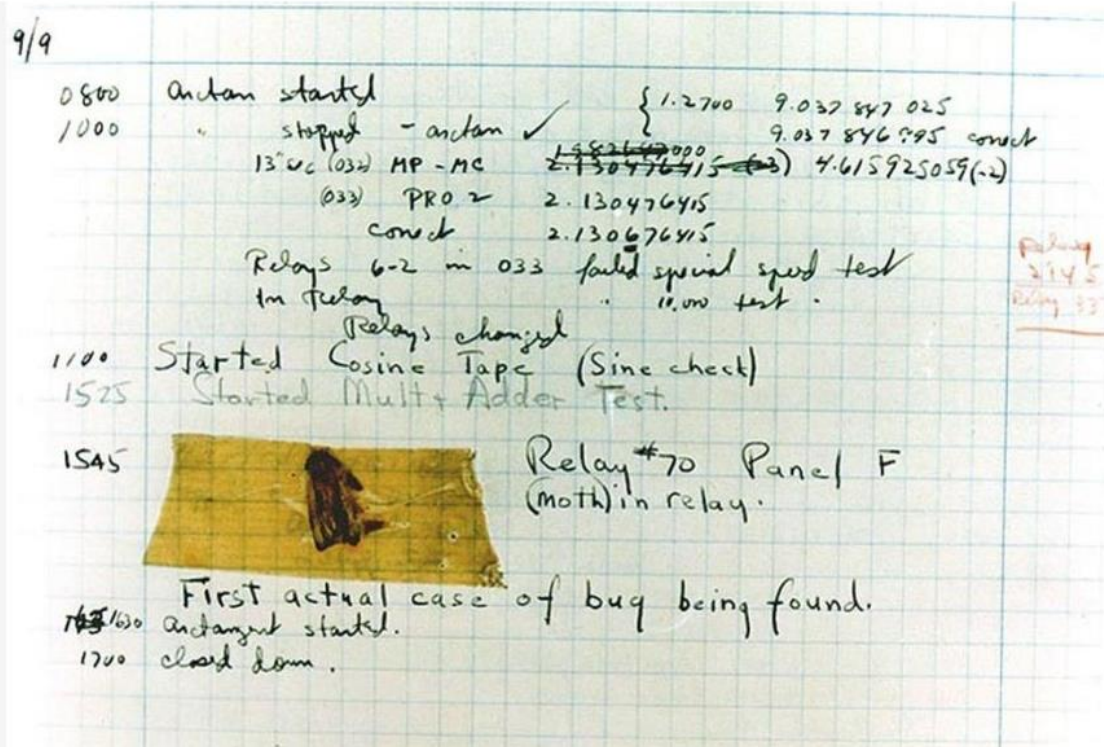
SUBJECTS

English Language Arts, Experiential Learning

Computer Bug

"First actual case of bug being found," according to the brainiacs at Harvard, 1945. The engineers who found the moth were the first to literally "debug" a machine.

PHOTOGRAPH COURTESY NAVAL SURFACE
WARFARE CENTER, DAHLGREN, VIRGINIA



○ print-statements for å finne feil

- Ulempe: Må fjernes/skjules etterpå, så brukerne ikke trenger å se masse debugging-informasjon
- Kan likevel være nyttig å lagre debugging-informasjon et sted i tilfelle det oppstår en feil i fremtiden
- Bedre alternativ: logging (til fil eller terminal)
 - Nivå 1: Kun advarsler
 - Nivå 2: Advarsler og informasjon
 - Nivå 3: Advarsler, informasjon og debug-meldinger





L I V E K O D I N G :
L O G G I N G
O G
D E B U G G I N G

○ Debuggeren i VS Code

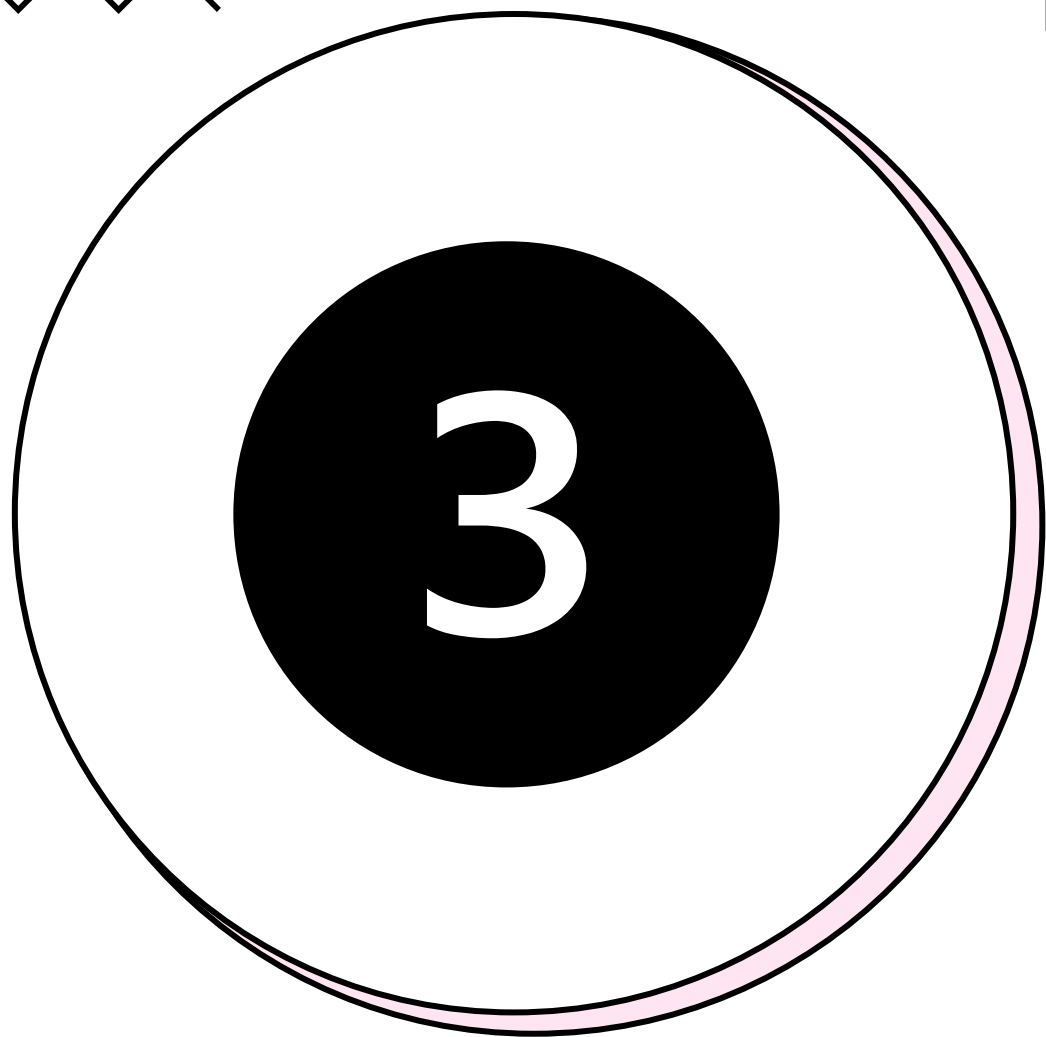
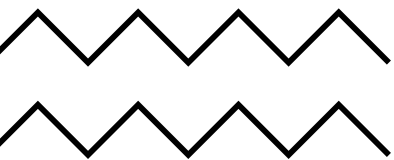
- Kan pause programmet der vi vil og følge med på hva som skjer med objekter/variabler
- Eventuelt pause bare når en betingelse er oppfylt (for å slippe de 100 første stegene der feil *ikke* skjer)
- Kan velge om vi vil jobbe på høyt nivå (ikke se hva som skjer inni en funksjon som kalles) eller lavt nivå (gå inn i funksjonen)
- Kan endre verdier av variabler for å teste
- Kan logge til en egen terminal uten å logge i selve koden



○ Debugging i C++ (Mac)

- Det finnes en [egen oppskrift](#) for dette





Etter forelesningen

- Repo for prosjekt 3 er nå ute
- Prosjektoppgaven kommer plutselig!
- Opptak fra fredag 18.
kommer så snart man får løst
noen tekniske problemer!

