

Effektiv Kode med C og C++

Forelesning 1, vår 2014
Alfred Bratterud

Agenda:

- * Litt om meg, og dere
- * Om C og C++: historie, relevans, egenskaper
- * Praktisk om kurset
- * Kode!
- * Opgaver for uka

Om meg:

- * Alfred Bratterud, Høgskolelektor og PhD-stipendiat
- * Veldig glad i programmering:
 - * C/C++, Python, javaScript, PHP
 - * Java, Lisp, Perl, ActionScript, C#, ASP...
- * Nå mest ifbm. forskning

Litt om dere

- * En kjapp quiz av hva dere kan...
- * if, else, for, while, switch, foreach ...
- * public, static, void, main
- * namespace
- * peker
- * generics
- * templates

Litt om dere

- * En kjapp quiz av hva dere kan...

- * **if, else, for, while, switch, foreach ...**

- * **public, static, void, main**

- * **namespace**

- * **peker**

- * **generics**

- * **templates**

Bør sitte i fingrene,
sammen med variabler, arrays etc.

Overordnet plan

- * Etter kurset skal man trygt kunne si at man kan C++
 - * Det meste av språket (keywords, operatorer etc.)
 - * En del av standardbiblioteket - for C og STL
 - * Kompilering, linking, preprosessering
 - * ISO C++, mulig noe C++11
- * Noe GUI, da kun med Qt

Overordnet plan

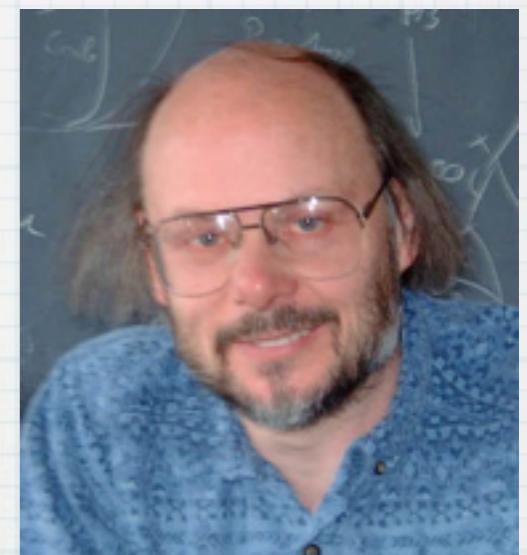
- * “Bottom up” i stedet for “Top down”
 - * Motivere “ålt vi gjør”
 - * Vi skal “forstå alt vi skriver”
- * Forstå “effektiv kode” på “høynivå” og “lavnivå”
- * Skrive lynraske, brukbare programmer
- * Mest kommandolinje, litt GUI

Litt historie

- * C ble utviklet av Dennis Ritchie v. Bell Labs, og kom ut i 1972
- * C++ ble utviklet av Bjarne Stroustrup, først kalt også ved Bell labs. C++ kom ut i 1983
- * Objektorientert programmering ble oppfunnet av to nordmenn (Yay!) Ole Johan Dahl og Kristen Nygaard, allerede i 1967



Dennis Ritchie, 1941-2011



Bjarne Stroustrup, f. 1950



Kristen Nygaard



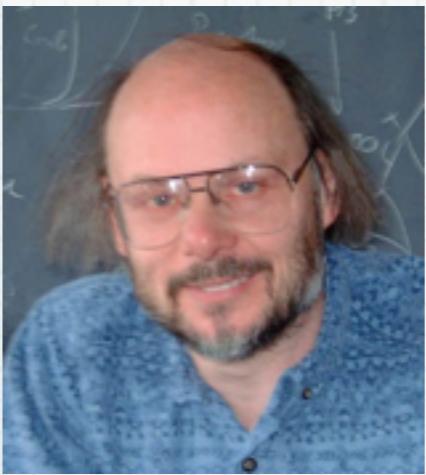
Ole Johan Dahl

To viktige - og bra bøker

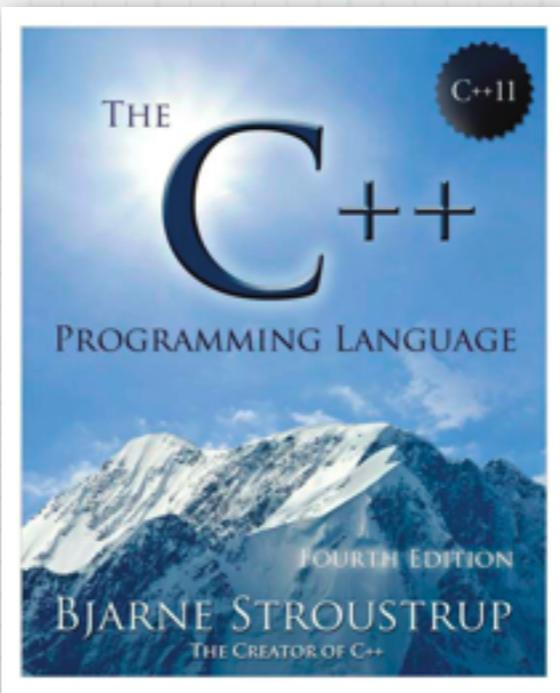
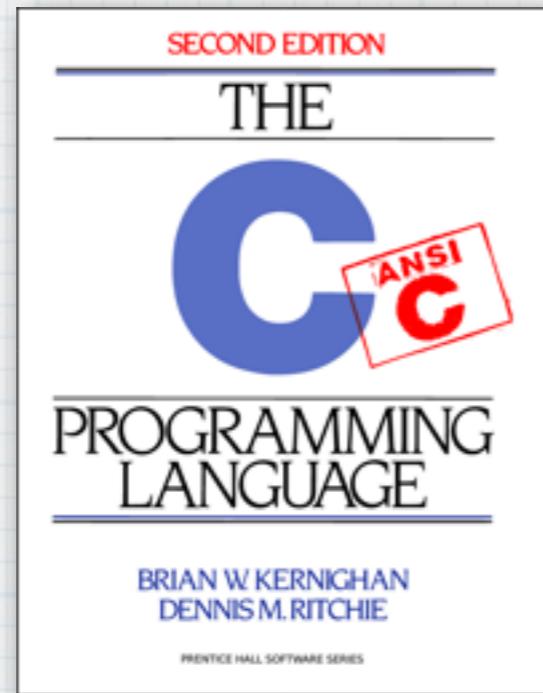
C++ - boken er støttelitteratur i faget



Dennis Ritchie



Bjarne Stroustrup

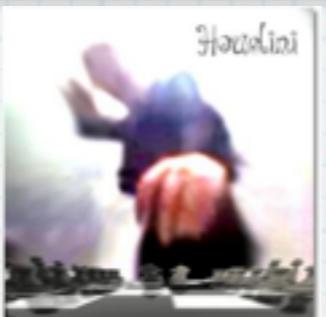
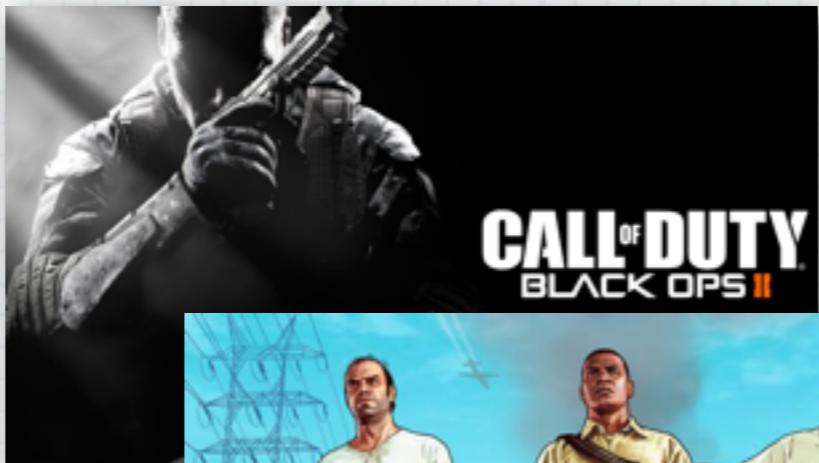
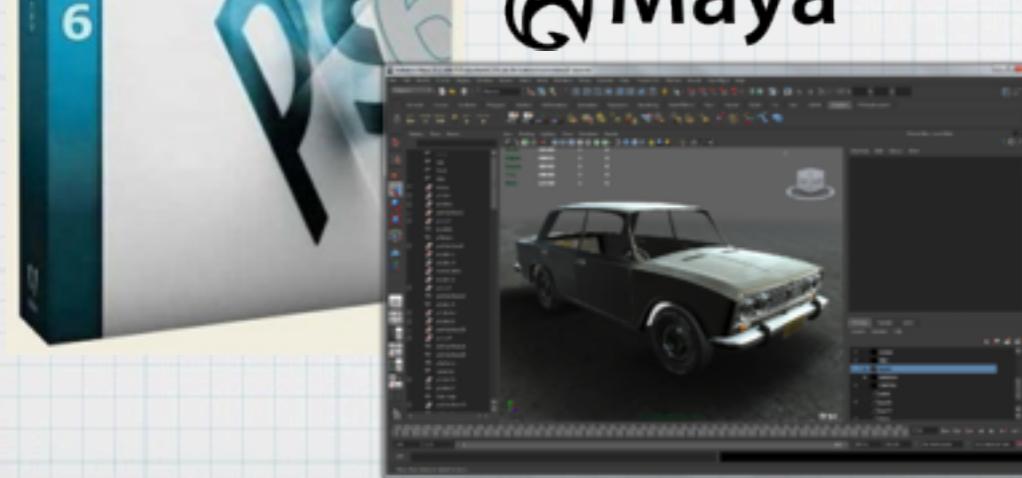


Hva brukes C++ til i 2014?

Linux™

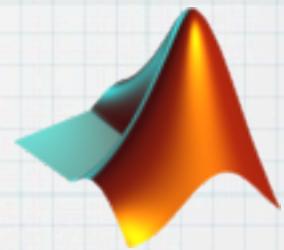


Mountain Lion OS X



XBOX 360™

PhysX™
by NVIDIA



MySQL™

python™





“you must know windows and C++ programming”

<http://msdn.microsoft.com/en-us/library/windows/apps/hh452744%28v=VS.85%29.aspx>

“Developers generally create games using C++ and DirectX 11.1”

<http://www.xbox.com/en-US/developers/faq>

Hvorfor bruker de C++ ?

- * Kort oppsummert: Effektivitet
- * Hvorfor så effektivt?
- * Direkte kompilert for hardware
- * Lavnivå minnetilgang
- * Avansert objektorientering
- * Kan “Alt”

Språklige forskjeller

(Ikke fullstendig, ikke representativt utvalg)

Egenskap	C	C++	Java	PHP
Imperative	X	X	X	X
Structured	X	X	X	X
Static types	X	X	X	
Type safe by default		(x)	X	-
OS-uavhengig	(x)	(x)	X	X
Generics		X	X	-
Object Oriented		X	X	X
Friendships		X		
Multiple Inheritance		X		
Functional	(x)	(x)		
Low-level memory	X	X		
Natively compiled	X	X		
Inline Assembly	X	X		
Operator overloading		X		
Garbage Collection			X	X

Så C++ er best i alt?

- * Vel...
 - * C++ er vanskeligere (dyrere) å bli god i
 - * Det er lettare å gjøre alvorlige feil
 - * ...Som java, men med "All the guns and all the knives"
 - * C++ følger "Zero Overhead principle": Ingen ting som går utover ytelsen blir tatt med uten at du ber om det (eller lager det)
 - * Man har altså ingen hushjelp (Garbage collector)
 - * Men man kan lære å lage 0 søppel og ikke å slose.

Blir det C eller C++ ?

- * C ⊂ C++ (nesten*)
- * Vi skal lære C++
- * ...Men bli kjent med mange av de tingene som kommer fra C
- * Er du i tvil, gjør det på C++ -måten

* http://en.wikipedia.org/wiki/Compatibility_of_C_and_C%2B%2B

Praktisk

Tid og sted

- * Denne uken: Intro i dag, forelesning/lab på onsdag
- * Deretter: (forhåpentlig)
 - * Kombinert forelesning/lab, onsdager, 08:30-12:15
 - * Forelesning fredager, 08:30-10:15
- * Funker det?

Vurdering

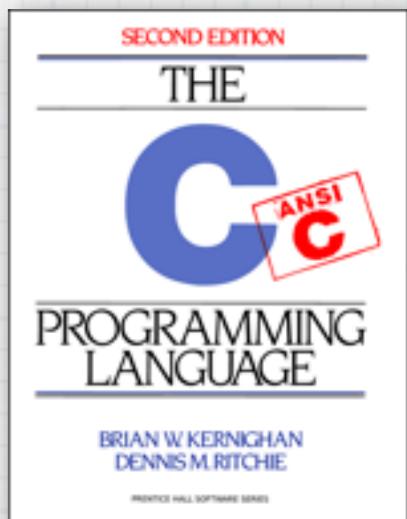
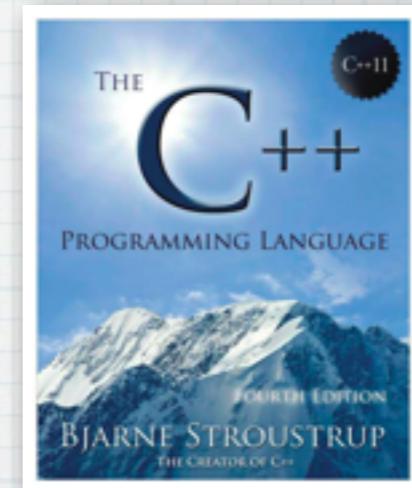
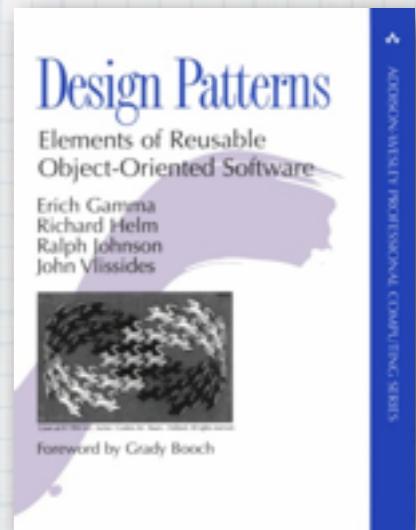
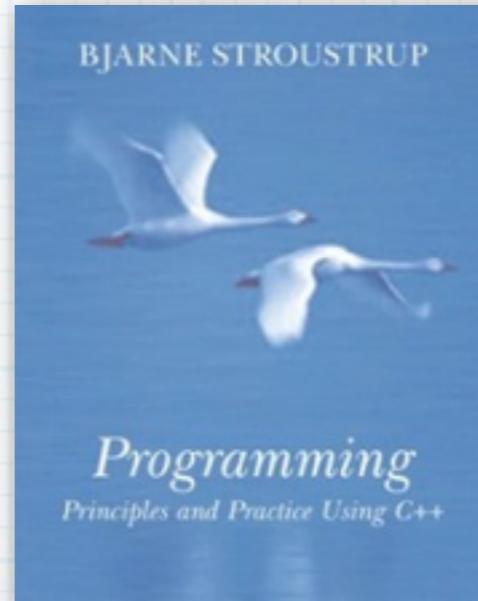
- * 2 obliger (arbeidskrav)
- * 1 individuell prøve
- * 1 prosjekt i gruppe
- * Presentasjon av prosjektet for meg
- * Mappeevaluering (A-F)
- * Detaljert plan kommer!

Github

- * Alt blir lagt ut på github:
https://github.com/hioa-cs/cpp_v2014
- * “Watch” repoet for å få varsler om oppdateringer!
- * Alle innleveringer skjer *også* gjennom github.
- * Alle må ha studentkonto på github, registrert med sin skole-e-post:
<https://github.com/edu>
- * HiOA-adresser skal være på whitelist
- * Når du har registrert deg, lever link til din github-bruker på fronter. Obligatorisk!

Pensum

- * "Programming: Principles and Practices Using C++", B. Stroustrup
- * Web: cplusplus.com tutorial
- * Støttelitteratur
 - * Web: C++ Annotations
 - * Gamma, Helm ++ "Design Patterns"
 - * Stroustrup, "The C++ Prog.Lang."
 - * Ritchie, "The C Prog. Lang."
 - * C++ GUI Programming with Qt4



Nå: Kode!

hello_world.cc
hello_world.cpp

Oppgave: Roulette

- * Hvis man da ser det har kommet 9 like på rad - bør man ikke da stase stort på motsatt farge?
- * Ved et "fair" roulette-bord skal det være 50/50 for sort og rød
- * Utvalg av tilfeldige variable er normalfordelt
- * Altså bør det veldig sjeldent komme seks like på rad, og ekstremt sjeldent 10 på rad...?
- * Kan løses med enkel statistikk - men hvorfor ikke bare simulere det?

Oppgave: Roulette

- * Jeg har laget en Python-implementasjon på github
- * Til Onsdag: Forsök å implementere det samme programmet i dit favorittspråk
- * Hva hvis jeg ikke liker/kan matte?
- * Matte er ikke fokus i kurset - og du trenger ikke skjonne den. Men, hvis du er interessert, prøv dette på wolframalpha.com
plot 100000*((2^x)^-1), x= 0 to 20, y=0 to 100000