

Fra C til grunnleggende C++ - hva er primært annerledes for oss?

C	C++	Kommentar
Prosedyre-orientert	Objekt-orientert	
.c	.cpp	Annen fil-extension (så kompilatoren vet hvilket språk den skal «rette» etter)
#include <stdbool.h>	-----	Uaktuelt i C++, da <code>bool</code> er en del av språket
#include <xxxxxxxxxx.h>	#include <xxxxxxxxxx>	Har <i>ikke</i> med fil-extension .h
#include <xxxxxxxxxx.h>	#include <cxxxxxxxxxx>	Mange standard .h-filer i C, finnes også i C++, bare med innledende 'c' og uten '.h'
#define / const STRLEN/MAX.....	const STRLEN/MAX.....	
printf(.....)	cout <<	(se eksempler)
scanf(.....)	cin >>	(se eksempler)
getchar() / getc()	cin.ignore()	Alle disse forkaster det neste tegnet i input-buffer, som ofte er '\n'
gets / fgets	getline(.....,	(se eksempler)
%...i/f/c/s	set...	setw / left / right / setprecision / fixed / showpoint
char navn[...]	string navn	«Glem» char-array, heretter brukes i stedet <code>string</code> (-klassen)
malloc (calloc) / free	new / delete	(se eksempler)
struct	(identisk)	Men, «glem» det snart. Vi skal primært bruke <code>class</code>
enum/struct Xxx yyyy	Xxx yyyy	Ikke påkrevet (men heller ikke ulovlig) i C++ å ha med <code>enum/struct</code> foran <i>definisjonen</i> av variabelen yyyy
..... foo(.....* t)foo(..... & t)	Oppdatere medsendt parameter direkte gjøres som oftest ikke via en peker (men det er fullt lovlig), men bruk av '&' (referanseoverføring)
NULL	nullptr	Verdi som pekere får/settes til/sjekkes opp mot
Casting		En måte i C, flere spesifikke måter i C++
Filhåndtering (lesing/skriving)	Helt annerledes i C++	(se eksempler)
	Overloading av funksjoner	Likenavnede funksjoner, men med forskjellige parametre og/eller antall
	Default parametre/argumenter	Kan utelate parametre, som da får standard definerte verdier
	(Overloading av operatorer)	(Omdefinere operators betydning/funksjonalitet vha. klasse)