

Sammanfattning av begrepp

Detta är ett *försök* att räkna upp begrepp som jag tagit upp på föreläsningarna. Det är *inte* säkert att jag fått med allt, detaljer finns inte med, till exempel finns inte alla typer här.

OBSERVERA!! Det som kommit upp på laborationerna finns inte med här, ni bör titta även på dem.

2001-12-10 Jan-Erik Jonsson

Introduktion, grundläggande begrepp

Vad är en dator? Datalogi/datakunskap

"löken": maskinvara/hårdvara, Operativsystem, stöd/hjälp/tillägg, tillämpningsprogram, användare, omgivning.

IT $\leftrightarrow$ ADB, Information $\leftrightarrow$ Data, tolka, koda

Databehandling. Vad är datorn är bra på?

Moores lag.

Operativsystem och ordbehandling

Vad är ett OS? Vad gör ett OS? Vad har ett bra OS?

Inbäddade datorer.

Ordbehandling: innehåll, utseende, formathierarki, dokumentmall, styckeformatmall, teckenformatmall, normal.dot

Intern datarepresentation och logik

bit, byte, ord, heltal, flyttal, 2-komplement, tecken, binär, hexadecimal, oktal, decimal, bas, position, ASCII

och $\bullet$ & & , eller + || , icke $\neg$!= , xor $\oplus$, sanningstabeller

Excel och datakommunikation

beräkning $\leftrightarrow$ svar på beräkning $\leftrightarrow$ presentation/format

överföringshastighet i bit (inte Byte) per sekund

seriell, parallell, simplex, halv duplex, full duplex, synkron, asynkron, protokoll, modem, RS-232, startbit, stoppbit, lsb/msb, paritet

Nätverkstopologi: stjärnnät, ringnät, bussnät.

Datorsystem och datorarkitektur

datorarkitektur, SISD mfl., von Neuman-arkitektur

CPU, SE, ALU, RAM, ROM, I/O-enheter, Primärminne(arbetsminne),

Sekundärminne(lagringsminne), register, adressbuss, databuss, styr/kontrollbuss.

Algoritmer och programutveckling

Vad är en algoritm? Vad ingår i en algoritm? Vad är en bra algoritm?
 exakthet, enkelhet, abstraktionsnivåer, fördelar med abstraktionsnivåer.
 Programvaruutveckling: programmera direkt, programming in small,
 programming in big, kostnader, analys, design, implementering/verkställande,
 testning, drift och underhåll och utveckling.

C++ , datatyper, in och utmatning, aritmetik

Vad behövs.

#include, main(), return-sats, cout, cin, <<, >>, satser, ; {} // /* */
 reserverade ord/nyckelord, variabler, konstanter, deklarerar, tilldela, initiera,
 variabelnamn, datatyper, heltal int m.fl , flyttal float m.fl , tecken char m.fl
 Aritmetik: operator, operand, operation, unär operator, binär operator,
 typomvandling, a++, ++a,
 länkning, kompilering, källkod, objektкод, exekverbart program, maskinkod,
 assembler, lågnivåspråk, högnivåspråk

Mera typer och operationer

Fält, element, index, sträng, dåliga minnesskyddet
 = + - * / % () , heltalsdivision, rest/modulus, prioriteter, paranteser

Styrande satser

Sekvens, Selektion, Iteration
 if, if_else, while, do_while, for, oändlig loop
 logiskt uttryck, true, false, bool, > < >= <= == != && ||
 satsblock/sammansatt sats

Strukturerad programmering, modulär abstraktion

algoritmer, moduler, modularisering, återanvändning m.m., Topdown-design,
 utöka språket, gränssnitt, parameter, pseudokod.

Standardfunktioner, bibliotek, texthantering

string.h, specialtecken, stdlib.h, time.h, iostream.h, iomanip, math.h
 strlen(), strcpy(), strcmp(), strcat(), cin.getline(), specialtecken

Funktioner

Returvärde, returtyp, void, deklaration, prototyp, parameter, argument,
 anrop/kalla på (-> kopia), värdeanrop, referensanrop (aut. eller ej), definition,
 funktionshuvud, funktionskropp, funktionsnamn, gränssnitt, return-satsen,
 parametertyp, utbytbarhet, inkapsling (svart låda).
 Fält som parameter, referenser/referensvariabel, (orienterande om pekare).
 Deklarationsområde, synlighet, maskering.

Fel i program:

Tvetydighet i algoritmen, kompileringsfel, syntaxfel, exekveringsfel, logiska fel,
 semantiska fel.