

Lyhyt osittainen johdatus C-kielen perusteiden alkeisiin

Heikki Korhola, tuntiopettaja

Tämä ohje on laadittu [Tietokoneen toiminta](#) -kurssia silmälläpitäen, sillä kurssin materiaalissa ja tehtävissä tulee jo aikaisessa vaiheessa vastaan C-kielen käsitteitä, mutta merkittävälle osalle opiskelijoita C-kieli ei ole ennestään tuttu. Kurssin esitietovaatimuksetkin edellyttävät vain **Javan** alkeiden osaamista (Ohjelmoinnin perusteet). Tämänkaltaisen lyhyt johdatus on siis sikäli perusteltu.

Huomaa että tämä ohje on tarkoituksella yksinkertaistettu. Lisää (paljon lisää) ja yksityiskohtaisempaa (paljon yksityiskohtaisempaa) tietoa C:stä löytyy Internetistä ja laitoksen muilta kursseilta, kuten esimerkiksi C-ohjelmointi/Programming in C.

19.03.2016 -- päivitetty viimeksi 06.04.2016

Osoittimet (pointer), viitteet (reference), yksinkertaisen syötteen luku ja tulostus (input/output)

C on laajalti käytetty matalan tason (laitteistoläheistä) ohjelmointia tukeva ohjelmointikieli. Oleelliset erot korkeamman tason kieliin tulevat siitä, että C:ssä muistinhallinta on kokonaan ohjelmoijan vastuulla. Periaatteessa mitä tahansa muistialueita voidaan lukea ja mille tahansa muistialueelle voidaan kirjoittaa (ohjelmointikielen puolesta, nykyiset käyttöjärjestelmät asettavat tälle kyllä rajoitteita).

Javan syntaksi pohjautuu C:hen, joten sikäli sitä on enimmäkseen helppo lukea, mutta tutuille symboleille * (asteriski) ja & (ampersandi) on C:ssä enemmän ja erilaisia käyttötarkoituksia, joista oleelliset käyvät ilmi tästä esimerkistä. Huom. tämä ei (tarkoituksella) ole kokonainen C-ohjelma, mutta kuvitelkaa että nämä lauseet ovat main-funktion sisällä.

```
1| int x = 7;  
2| int* px;  
3| px = &x;
```

Ensimmäisellä rivillä määritellään tavallinen arvotyyppinen muuttuja x ja samalla alustetaan se arvolla 7. Tässä ei ole mitään uutta tai ihmeellistä.

Toisella rivillä sen sijaan **on** jotain uutta. Asteriski muuttujan tyyppin perässä ilmaisee, että kyseessä on osoitintyyppinen muuttuja eli *osoitin (pointer type)*. Muuttujan px arvo on täten merkitty jonkin muun tiedon osoitteeksi ja jatkossa sitä käytettäessä tämä on otettava huomioon. Osoitin on muuttuja siinä missä muutkin muuttujat ja sen arvo voi C:ssä olla mielivaltainen (tässä sille ei ole heti annettu alkuarvoa). On kuitenkin tarkoituksenmukaista että osoitintyyppisen muuttujan arvo on muistiosoite, tämänkaltaisissa yksinkertaisissa tapauksissa **tämän** ohjelman osoiteavaruudessa, sillä jos se on jotain muuta, ohjelman suoritus päättyy virheelliseen muistiviittaukseen.

Seuraavaksi, rivillä 3, muuttujan px arvoksi asetetaan muuttujan x **osoite**, joka saadaan käyttämällä ampersandia eli *viiteoperaattoria (reference tai address of)* tunnuksen edessä.

Lopulta arvomuuttujan x sisältämään tietoon päästään käsiksi **epäsuorasti**, osoittimen kautta:

```
4| int y = *px;
```

Asteriski on tässä jälleen uudessa ja eri roolissa kuin aiemmin, jolloin se oli lauseen vasemmalla puolella muuttujatyypin perässä. Osoittimen edessä käytettynä asteriski toimii *viitteen purku* (*dereference*)-operaattorina, ja tällaisen ilmaisen arvo on siis "osoittimen päästä löytyvä arvo" (tässä y:n arvoksi tulee siis 7).

Tiedon lukeminen käyttäjältä tapahtuu `scanf`-funktiolla:

```
scanf("%d", &x);
```

Funktion ensimmäinen parametri on merkkijono, jonka sisällä määritellään luettavan tiedon muoto (f niinkuin *formatted*). %d on tässä sopivin määrittelijä koska haluamme lukea desimaalimuotoisen kokonaisluvun. Toinen parametri on **osoite**, johon syöte luetaan.

Tulostus puolestaan onnistuu `printf`-funktiolla:

```
printf("%d\n", *px);
```

Tämä toimii samalla periaatteella kuin `scanf`. Erona on, että ensimmäinen parametri voi olla mielivaltainen merkkijono (esimerkiksi "arvo on: %d \n"). Tulostuksen muotoilua ohjataan määreillä kuten %d joka merkitsee että aikomuksena on tulostaa **tähän kohtaan** kokonaisluku. Yhdistelmä \n on rivinvaihtomerkki. Toinen parametri on arvo, joka sijoitetaan merkkijonossa edellämainitun "paikanpitäjän" kohdalle. Koska px on osoitintyyppinen muuttuja, sen viite on purettava jotta varsinaiseksi parametriksi tulee haluttu arvo.

Mainitaan vielä kattavuuden vuoksi yksi piirre. Ns. hyviin tapoihin kuuluu, että C-ohjelmien `main`-funktion lopussa on lause **return 0;**. C-ohjelmien `main`-funktio on määritelty `int`-tyyppiseksi ja palautusarvolla 0 ilmaistaan kutsujalle, eli käyttöjärjestelmälle että ohjelman suoritus on päättynyt onnistuneesti.
