

**CSE-A1111 Ohjelmoinnin peruskurssi Y1****T-106.1208 Ohjelmoinnin perusteet Y (Python)****Tentti 29.11.2014**

Kirjoita jokaisen vastauspaperisi alkuun kurssin nimi, kokeen päivämäärä, nimesi, opiskelijanumerosi, vastauspaperiesi kokonaismäärä sekä allekirjoituksesi.

**Tärkeitä ohjeita vastausten kirjoittamiseen:** Kun kirjoitat ohjelmakoodia, käytä kahden ruudun levyisiä sisennyksiä. Jos sisennyksiä ei ole käytetty tai niistä ei saa selvää, vähennetään siitä pisteitä. Kirjoitettavaan ohjelmakoodiin ei tarvitse lisätä kommentteja. Missään tehtävässä tulostusta ei tarvitse muotoilla. Voit myös olettaa, että käyttäjän antama syöte on virheetöntä, ellei tehtävässä erikseen käsketä käsittelemään virhetilanteita.

**Tentissä ei saa käyttää laskimia eikä lisämateriaalia. Opiskelijat, joiden äidinkieli ei ole suomi, saavat kuitenkin käyttää sanakirjaa, jos siinä ei ole merkintöjä (tentin valvoja tarkistaa sanakirjan).**

1. Kohdissa a, b, c ja d kerro, mitä annettu ohjelma tulostaa. Vastausta ei tarvitse perustella. Kohdissa e, f ja g kerro, mitä tehtävässä esitetty funktio tekee. Älä selitä funktion toimintaa käsky käskyltä, vaan selitä parilla lauseella, mikä on funktion tarkoitus (esimerkiksi: "funktio laskee ja palauttaa parametrina annetussa listassa olevien lukujen summan"). Funktioille annettavien parametrien luonne on selitetty kunkin kohdan yhteydessä. Huomaa, että annetuissa ohjelmissa tai funktioissa voi olla myös virheitä. Kerro siinä tapauksessa, mitä annettu virheellinen ohjelma tulostaa tai miten virheellinen funktio toimii - ei siis sitä, miten ohjelman tai funktion pitäisi toimia, jos siinä ei olisi virheitä.

a) (2 p)

```
def main():
    matka = 3100
    if matka > 2200:
        print("Cooperin testin tulos on hyva.")
    elif matka > 2800:
        print("Cooperin testin tulos on erinomainen.")
    else:
        print("Cooperin testin tulos on huono.")
```

main()

b) (2 p)

```
def funktio(luku):
    muuttuja = 5
    return 2 * luku

def main():
    arvo = 8
    muuttuja = funktio(2 * arvo)
    print(muuttuja)
```

main()

c) (3 p)

```
def main():
    lista = [5, 15, 50]
    tulos = 10
    i = 0
    while i < len(lista):
        if tulos < 20:
            tulos = tulos + 2 * lista[i]
        else:
            tulos = tulos + lista[i]
        i = i + 1
    print(tulos)
```

main()

d) (5 p)

```
def muuttele(arvo, lista):
    arvo = 8
    lista[0] = 50

def main():
    luku = 10
    luvut = [1, 2, 3, 4]
    muuttele(luku, luvut)
    print(luku)
    for alkio in luvut:
        print(alkio)

main()
```

e) Funktiolle annetaan parametreina desimaalilukuja sisältävä lista ja desimaaliluku. (4 p)

```
def mysteeril(listal, raja):
    i = 0
    lista2 = [0.0] * len(listal)
    while i < len(listal):
        if listal[i] > raja:
            lista2[i] = listal[i] * 1.23
        else:
            lista2[i] = listal[i]
        i += 1
    return lista2
```

f) Funktiolle annetaan parametreina kaksi merkkijonoa, joissa molemmissa on vähintään yksi merkki. (4 p)

```
def mysteeri2(merkkijonol, merkkijono2):
    uusi = merkkijonol[0]
    uusi += "."
    uusi += merkkijono2[0]
    uusi += "."
    return uusi
```

g) Funktiolle annetaan ensimmäisenä parametrina positiivisia kokonaislukuja sisältävä lista ja toisena parametrina positiivinen kokonaisluku (5 p)

```
def mysteeri3(lista, luku):
    uusi = []
    i = 0
    while i < len(lista):
        if lista[i] % luku == 0:
            uusi.append(lista[i])
        i += 1
    return uusi
```

**Vastaa myös kurssin palautekyselyyn. Kyselyyn vastaamisesta saa 200 harjoitustehtäväpistettä. Linkki kyselyyn lähetetään kurssille ilmoittautuneille sähköpostitse.**

Tentti jatkuu seuraavalla sivulla

2. a) Olet töissä yrityksessä, joka on pyytänyt kahdelta painotalolta tarjouksen mainoslehtisten painamisesta. Painotalo A laskuttaa painettavien kappaleiden määrästä riippumattomana aloitusmaksuna 150 euroa ja sen lisäksi 5 senttiä jokaisesta painettavasta kappaleesta. Painotalo B ei pyydä aloitusmaksua, mutta se laskuttaa aina vähintään 1000 kappaleesta, vaikka mainoksia painettaisiin vähemmän. Myös kappalehinta voi olla eri kuin Painotalo A:lla. Kirjoita Python-ohjelma, joka pyytää käyttäjältä painettavien kappaleiden määrän ja Painotalo B:n laskuttaman kappalehinnan. Ohjelma tulostaa, kumman painotalon tarjous on edullisempi ja mikä on tämän edullisemman vaihtoehdon kokonaishinta. (10 p.)

b) Kiinteistönvälittäjä saa palkkiota jokaisesta myymästään asunnosta 2 % asunnon myyntihinnasta, mutta kuitenkin vähintään yrityksen päättämän vähimmäispalkkion. Palkkio lasketaan erikseen jokaiselle myydylle asunnolle. Kirjoita Python-funktio `laske_kokonaispalkkio(myyntihinnat, vahimmaisipalkkio)`. Funktio saa ensimmäisenä parametrina listan, joka sisältää välittäjän myymien asuntojen myyntihinnat (kukin alkio on yhden asunnon myyntihinta). Funktion toinen parametri on yhden asunnon myynnistä tuleva vähimmäispalkkio. Funktio laskee ja palauttaa arvonaan myydyistä asunnoista välittäjälle yhteensä tulevan palkkion. Voit olettaa, että funktiolle on annettu järkevät parametrit. Tehtävässä ei tarvitse kirjoittaa muuta kuin pyydetty funktio. (20 p)

3. Eräällä kurssilla opiskelijoiden tiedot on tallennettu tekstitiedostoon seuraavasti: Yhdellä rivillä on ensin opiskelijan opiskelijanumero, sitten nimi, tämän jälkeen harjoituspisteet ja lopuksi tenttipisteet. Eri tiedot on erotettu toisistaan kaksoispisteellä. Tiedoston rivit voisivat näyttää esim. seuraavilta:  
 123456:Tiina Teekkari:1800:46  
 334455:Sampo Sahkolainen:1660:52

Voit olettaa, että yhden opiskelijan tiedot ovat tiedostossa vain yhteen kertaan. Kurssista pääsee läpi ne opiskelijat, jotka ovat saaneet harjoitustehtävistä vähintään 1500 pistettä ja tentistä vähintään 50 pistettä. Kirjoita Python-ohjelma, joka pyytää käyttäjältä tiedot sisältävän tiedoston nimen. Ohjelma lukee tiedot tästä tiedostosta ja tulostaa kaikkien läpipäässeiden opiskelijoiden nimet ja opiskelijanumerot.

Ohjelman on käsiteltävä seuraavat virhetilanteet:

- Annetun nimistä tiedostoa ei ole olemassa tai tiedoston lukeminen ei onnistu jostain muusta syystä
- Tiedoston jollain rivillä harjoitus- tai tenttipisteiden paikalla ei ole kokonaisluku .

Näissä tapauksissa ohjelma ilmoittaa käyttäjälle, millainen virhe on sattunut, ja lopettaa toimintansa. Ohjelman ei siis tarvitse jatkaa rivien lukemista virheellisen rivin jälkeen. Voit myös olettaa, että tiedoston jokaisella rivillä on täsmälleen neljä toisistaan kaksoispisteellä erotettua osaa. Ohjelman ei tarvitse osata käsitellä esimerkiksi sellaisia virhetilanteita, joissa rivi on tyhjä tai ei sisällä opiskelijanumeron ja nimen lisäksi muuta tekstiä. (20 p)

**VIIMEINEN TEHTÄVÄ SEURAAVALLA SIVULLA**

4. Kirjoita luokka `Asiakas` kaupan tiliasiakkaan kuvaamiseen. Asiakas voi tehdä ostoksia kaupassa kahdella tavalla: hän voi joko maksaa ostoksen heti (käteisellä) tai ostaa luotolla (tilille), jolloin ostossumma lisätään asiakkaan velkaan. Velan lisäksi asiakkaista pidetään kirjaa heidän suurimmasta ostoksesta jollain aikavälillä. Näin kaupan on helppo poimia asiakasrekisteristä sellaiset asiakkaat, jotka tekevät suuria kertaostoksia. Käteisellä maksetuista ostoksista ei pidetä muuten kirjaa, mutta ne vaikuttavat siihen, mikä on asiakkaan suurin kertaostos tarkastelulla aikavälillä.

Asiakas-oliolla on oltava seuraavat kentät:

- `__nimi` asiakkaan nimi
- `__luottoraja` määrä (euroina), jonka asiakas voi olla korkeintaan velkaa kaupalle
- `__velka` asiakkaan velka kaupalle euroina (positiivinen, jos asiakas on velkaa)
- `__suurin_ostos` asiakkaan suurimman kertaostoksen arvo euroina tarkasteltavana aikavälillä.

Määrittele luokkaan seuraavat metodit. (Jos metodin kuvauksessa ei ole kerrottu mitään metodin palauttamasta arvosta, metodin ei tarvitse palauttaa mitään.)

- `__init__(self, nimi1, luottoraja1)` luo uuden `Asiakas`-olion. Asiakkaan nimi ja luottoraja annetaan parametreina. Uuden asiakkaan velka ja suurin ostos ovat 0.0.
- `kerro_nimi(self)` palauttaa asiakkaan nimen.
- `kerro_luottoraja(self)` palauttaa asiakkaan luottorajan.
- `kerro_velka(self)` palauttaa asiakkaan velan määrän.
- `kerro_suurin_ostos(self)` palauttaa asiakkaan suurimman ostoksen arvon.
- `muuta_luottoraja(self, uusi_raja)` asettaa asiakkaan uudeksi luottorajaksi parametrina annetun arvon, jos se on mahdollista. Uusi luottoraja ei voi olla negatiivinen eikä myöskään pienempi kuin asiakkaan velka tällä hetkellä. Jos parametri ei täytä näitä ehtoja, metodi ei muuta luottorajaa ja se palauttaa arvon `False`. Jos luottorajan muutos onnistuu, metodi palauttaa arvon `True`.
- `lisaa_ostos(self, arvo, onko_velaksi)` lisää tarvittavat tiedot asiakkaan uudesta kertaostoksesta. Ostoksen arvo annetaan parametrina. Viimeisen parametrin arvo kertoo sen, onko ostos maksettu käteisellä vai luotolla. Parametrin arvo on `True`, jos ostos on maksettu luotolla, ja muuten `False`. Metodi päivittää tarvittaessa velan ja suurimman kertaostoksen arvoa. Asiakas voi tehdä ostoksen velaksi vain siinä tapauksessa, että velka ei ostoksen jälkeen ylitä asiakkaan luottorajaa. Asiakas ei voi myöskään koskaan tehdä arvoltaan negatiivista ostosta. Jos ostoksen teko ei onnistu, metodi palauttaa arvon `False` eikä muuta minkään kentän arvoa. Jos ostoksen teko onnistuu, metodi tekee tarvittavat päivitykset ja palauttaa arvon `True`.
- `lyhenna_velkaa(self, lyhennys)` lyhentää (pienentää) asiakkaan velkaa parametrina annetulla summalla. Parametrin täytyy olla positiivinen. Jos parametri on negatiivinen tai nolla, metodi ei tee mitään. Lyhennys saa olla suurempi kuin asiakkaan velka ennen lyhennystä. Tällöin velka muuttuu negatiiviseksi (jolloin kauppa on velkaa asiakkaalle).
- `poista_suurin_ostos(self)` poistaa asiakkaan tiedoista suurimman ostoksen (eli asettaa sen arvon nollassi). Tätä metodia voidaan käyttää halutun tarkasteluajavälin päätyttyä sen jälkeen, kun tiedot eri asiakkaiden suurimmista ostoksista edellisellä aikavälillä on kerätty talteen.
- `__str__(self)` palauttaa merkkijonon, joka sisältää asiakkaan nimen, luottorajan, velan määrän sekä suurimman ostoksen arvon tarkasteltavana aikavälillä.

Kirjoita lisäksi pääohjelma, joka luo kaksi `Asiakas`-oliota ja sen jälkeen muuttaa niistä toisen luottorajaa. Sitten pääohjelma lisää ostoksen kummallekin asiakkaalle ja tulostaa, onnistuiko ostoksen lisäys. Tämän jälkeen ohjelma lyhentää ensiksi luodun asiakkaan velkaa ja tulostaa hänen velkansa lyhennyksen jälkeen. Lopuksi ohjelman pitää tulostaa molempien asiakkaiden tiedot (nimi, luottoraja, velan määrä ja suurimman ostoksen arvo). Voit päättää asiakkaiden ja ostosten tiedot itse. Pääohjelman ei siis tarvitse kysyä mitään käyttäjältä. (25 p)