OHJ-1400 Olio-ohjelmoinnin peruskurssi

Tentissä ei saa käyttää ylimääräistä kirjallista materiaalia, laskimia, tietokoneita tai muita lunttausvälineitä.

Muutama sana tenttivastauksen kirjoittamisesta:

1. Vastauksessa oleta vastaavasi sellaisen ihmisen kysymykseen, joka tuntee kohtalaisen hyvin ohjelmistotekniikan aihealuetta muutoin paitsi tämän kysymyksen osalta. Muista että vastauksesi tarkoitus on vakuuttaa tarkastaja osaamisestasi.
2. Mieti etukäteen esim. ranskalaisilla viivoilla vastauksesi pääkohdat ja lajittele ne johdonmukaiseen järjestykseen — älä kirjoita yhteen pötköön kaikkea mieleen tulevaa.
3. Muista vastata kaikkiin tehtävän kysymyslauseisiin, sillä täysiä pisteitä ei voi saada jos kaikkiin kysyttyihin asioihin ei ole vastattu.

Palauta kaikki *nimetyt* vastauspaperit *omiin pinoihinsa*!

..... Tehtävät 1. & 2. **omalle paperilleen**! Nimi paperiin! .....

1. Alla on selitetty joitain C++:n ja olio-ohjelmoinnin termejä. Mistä termeistä on kysymys? Jos lisäksi selityksessä on jotain vialla, korjaa selitys.
   a) Jäsenfunktiot (ja -muuttujat), jotka näkyvät vain luokasta periytetyille luokille.
   b) Luokan rakentaja, joka ei saa parametreja ja joka luodaan jokaiselle C++:n luokalle automaattisesti.
   c) C++:n ilmaisu, jonka jälkeen annettuun luokkaan voi tehdä osoittimia, mutta luokan muu käyttäminen ei onnistu.
   d) Jäsenfunktion kutsuhetkellä ohjelma päättelee ajoaikana kutsussa käytetyn osoittimen tyypin perusteella, mihin jäsenfunktion toteutukseen siirrytään.
   e) UML:n kaavio, josta käy ilmi ohjelman ajoaikainen käyttäytyminen kokonaisuudessaan.
   f) Luokka, joka kuvaa joidenkin toisten luokkien yhteisiä ominaisuuksia ja toimii pohjana näille luokille.
2. Seuraavassa on muutama pikkuessee. Pyri vastaamaan kuhunkin kaikki olennaiset asiat mukaan ottaen.
   a) C++:n `const`-mekanismi ja olio-ohjelmointi.
   b) Mitä hyötyä C++:n *purkajista* on dynaamisessa muistinhallinnassa? Entä tuovatko ne uusia vaaroja dynaamiseen muistinhallintaan?
   c) Millä eri tavoin oikeanlainen periytymisen käyttö lisää koodin yleiskäyttöisyyttä ja vähentää tarvetta saman koodinpätkän moneen kertaan kirjoittamiseen?

..... *KÄÄNNÄ!* .....

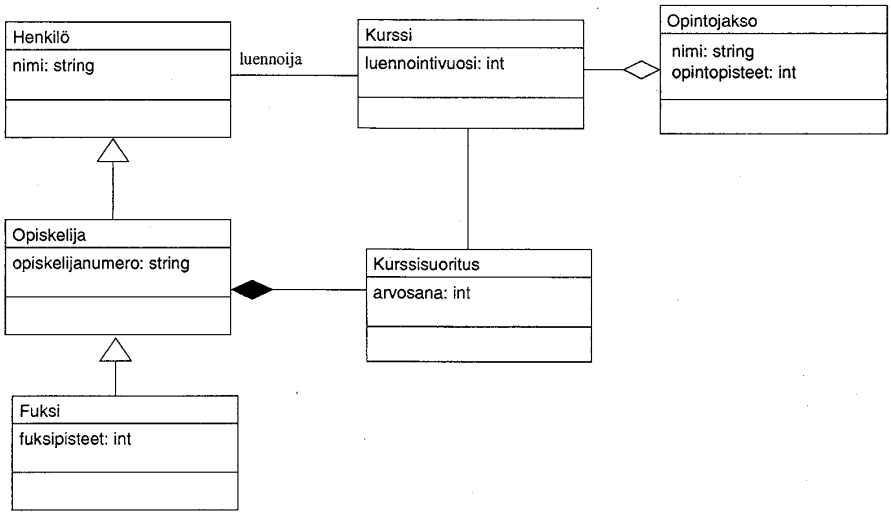
..... Tehtävät 3. & 4. **omalle paperilleen**! Nimi paperiin! .....

3. Selitä (n. max. 6-7 riviä/kohta) seuraavat olio-ohjelmoinnin käsitteet ja mitä hyötyä niistä saadaan olio-ohjelmoinnissa. ***Älä*** selitä niistä pelkkää syntaksia tms. vaan kerro etupäässä, mitä ko. käsitteet *tarkoittavat*.

   a) Kapselointi *(encapsulation)*
   b) Abstraktio *(abstraction)*
   c) Luokan vastuualue *(responsibility)*
   d) Koostumissuhde *(aggregation)*
   e) CRC-kortti *(CRC card)*
   f) Abstrakti kantaluokka *(abstract base class)*

4. Alla on osa TTKK:n kursseja ja niiden suorituksia kuvaavaa luokkakaaviota (rajapinnat on yksinkertaisuuden vuoksi jätetty pois). Vastaa sen perusteella seuraaviin kysymyksiin:

   a) Mitä tarkoittaa Opintojakson ja Kurssin välinen "avoin salmiakki"?
   b) Mahdollistaako kaavio sen, että jonkin kurssin opiskelija voisi toimia myös luennoijana? Perustele vastauksesi.
   c) Kopioi kaavio vastauspaperille ja lisää luokkien välisiin suhteisiin lukumäärämerkinnät ja tarvittaessa suunnat. Perustele lyhyesti valintasi.
   d) Kaavion alla on hahmotelma kaavion Kurssi-luokasta C++:lla. Kirjoita private-puolelle kaavion perusteella järkevät jäsenmuuttujien esittelyt. Perustele valintasi.
   e) Lisää puuttuvat const-sanat luokan esittelyyn (riittää, että kopioit vastauspaperille rivinumeroineen ne rivit, joihin tulee muutoksia).
   f) Onko luokan esittelyssä virheitä tai kyseenalaisuuksia? Korjaa ne.




```
1  #include <string>
   #include <iostream>
3  using namespace std;

5  class Kurssi : public Opintojakso
   {
7    Kurssi();
     int milloin_luennoitu();
9    string kuka_luennoi();
     void vaihda_luennoija(Henkilo uusi_luennoija);
11   string anna_kurssin_nimi();
     int anna_opintopisteet();
13   void lisaa_suoritus(Kurssisuoritus* suoritus);
     // ...
15 private:
     // ?
17 };
```