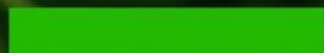




LAND OF THE CURIOUS



LUT CS1 LAPPEENRANTA OHJELMOINNIN PERUSTEET

Uolevi.Nikula@lut.fi

OHJELMOINNIN PERUSTEET @LUT LAPPEENRANTA

- » Normaali 6ects ohjelmoinnin perusteet –kurssi suomen kielellä ja Pythonilla
 - » Lahdessa englannin kielinen kurssi, noin 700 ilmoittautunutta, Pythonilla
- » Osallistujia kaikista koulutusohjelmista noin 500
 - » Pari vuotta sitten ilmoittautuneita oli 600 eli ilmoittautuneiden määrä laskenut
 - » Ilmoittautuneista 90% aloittaa kurssin, 10% poistetaan 3 viikon jälkeen suoritusten puutteessa
- » Kurssin perusrakenne
 - » Viikkotehtävät: teoria eli väittämät (ohjelmointisäännöt) ja viikoittaiset ohjelmointitehtävät
 - » Harjoitustyö: valittavana 3 eri tasoa – minimi, perus ja tavoitetaso (120, 190 ja 330 riviä malliratkaisussa)
 - » Tentti EXAMissa: Python, IDLE, tee toimiva ohjelma – jos ei toimi, arvosana on 0
- » Kurssilla oma Python-ohjelmointiopas, vastaa luentorakennetta, n. 210 sivua
 - » Opas on julkaistu viime vuosina luku per viikko -tahdilla



KURSSIN OSAAMISTAVOITTEET L1-7

1. Johdanto, syöttö, tulostus, tietotyypit, muuttujat
2. Perusohjelma Pythonilla, muuttujien roolit, tyyppimuunnokset, laskentaa, merkkijonot, tulostus osa 2
3. Valintarakenne, ehdollinen koodi, haarautuminen, koodilohko, loogiset operaattorit, valikkopohjainen ohjelma
4. Toistorakenteet, alkuehtoinen ja askeltava
5. Ohjelmien rakenne, aliohjelmien määrittely ja kutsuminen, tiedonvälitys, nimiavaruudet
6. Tiedostonkäsittely, lukeminen, kirjoittaminen jäsenfunktiot, muotoiltu tulostus/format
7. Rakenteiset tietorakenteet, lista, luokka ja olio, oliolista




KURSSIN OSAAMISTAVOITTEET L8-14

8. Uudelleenkäyttö, kirjastot, hyötyjä ja haasteita
9. Laadunvarmistus, testaaminen, poikkeukset, virheet, debuggaus
10. Data-analytiikka, sanakirja, lajittelu, matriisi
11. Ohjelmien tehokkuus, algoritmi, pseudokoodi, rekursio, kääntäjä ja tulkki
12. Tiedon esitysmuodoista, ASCII ja muut merkkitaulukot, kantaluvut ja niiden muunnokset
13. Käyttöliittymät ja dokumentointi, kuvaustekniikat, graafinen ja komentorivikäyttöliittymä
14. Lopetus, jatkoa



KURSSIN TOTEUTUKSESTA

- » Kurssin perusidea on, että tekemällä oppii eli kurssilla on 60 henkilökohtaista viikkotehtävää, noin 5 viikossa
- » Tehtävät palautetaan CodeGradeen
 - » Rakennetesteja eli semgrep, tarkistetaan ohjelman toteutusta eli “sääntöjen mukaan”
 - » Tulosteiden tarkastus, toimii ohjeiden mukaan
- » Plagiarismi
 - » Kaikki tehtävät henkilökohtaisia, saa palauttaa vain itse tekemäänsä koodia, ts. ei netistä, kaverilta, AI:lta
 - » Viikkotehtävien ratkaisut rupesivat kiertämään, joten nykyään tehtävistä suurin osa on variantteja, ts. vastaava tehtävä, mutta aiempi ratkaisu ei toimi oikein (tavoite 6 varianttia per tehtävä)
 - » Harjoitustöissä samat teemat ja vastaavat algoritmit sekä tietorakenteet, mutta tehtävät variantteja
 - » CodeGradessa plagiarismin tarkastus, tällä hetkellä noin 10-20 työtä vuosittain on niin samanlaista, että ne johtavat tarkistukseen eli “osaamisen osoittamiseen” EXAMissa

TEKOÄLY TÄLLÄ KURSSILLA

- »» Tekoälyn käyttö kurssilla on kielletty
 - »» Perustehtävät on tehty ohjelmoinnin perusrakenteiden opettelua varten, ei tekoälyä varten
 - »» Ohjelmoinnin perusteita ei opi muuten kuin tekemällä itse
 - »» Tekoälyn tekemien ohjelmien arviointi ja korjaaminen ei onnistu, jos ei itse osaa ohjelmoida (paremmin kuin tekoäly)
 - »» Ohjelmoinnin perusteiden opiskelu on aiemmin vaatinut 6op ja tämä ei ole muuttunut
- »» Jatkossa
 - »» Sanamuotoja pitää tarkentaa, koska opiskelijat käyttävät tekoälyä niin laajasti muuten
 - »» Arvosana tulee todennäköisesti 70% tenttiarvosanasta - EXAM-tentissä ei ole tekoälyä käytössä
- »» Huom
 - »» Tekoäly näyttää toimivan generointiin – generoidaan uusia/parempia ohjelmia
 - »» Ymmärrettävien ja ylläpidettävien ohjelmien tekeminen, ja ylläpito, on toinen asia (vrt. 1970-luvun koodi)

OHJELMOINTINSÄÄNNÖT ELI TYYLIOHJEET

- » Kurssin tavoitteena on ymmärrettävä ja ylläpidettävä koodi
 - » Lähtökohtaisesti opiskelijat tavoittelevat toimivaa koodia eikä ymmärrettävyys kiinnosta
 - » Yrity maailmassa on sama ongelma, ja siellä asiaa ratkotaan tyyliohjeilla tai säännöillä/standardeilla
- » Ohjelman rakenteen arviointi edellyttää selkeää dokumentointia
 - » Kurssilla käytetään kurssilla opetettuja rakenteita – Lahdessa CodeGrade / semgrep tarkistuksia
 - » Rakenteita pitää käyttää oppaan ja sääntöjen mukaisesti – LPR, ASPA ja CodeGrade/semgrep/opettaja
- » Ohjelmointisääntöjen eli tyyliohjeiden käyttöönotossa on useita haasteita
 - » Ohjeiden noudattaminen on monille opiskelijoille vaikeaa
 - » Useimmat opiskelijat myöntävät, että jos pomo maksaa palkkaa, niin silloin kannattaa tehdä sellaisia ohjelmia ja sillä tavalla kuin pomo haluaa
 - » Tyyliohje-käsitettä ei hyväksytty, mutta ohjelmointisääntö-käsite menee läpi liikennesääntö-analogialla



KURSSILLA KÄYTETTÄVÄT TYÖKALUT 1

- » Python ja IDLE, paljon sivuaineopiskelijoita eli lähtökohta on yksinkertaisuus
- » Ohjelman toiminnan automaattitarkastus: CodeGrade
- » Ohjelman rakenteen tarkastus
 - » ASPA eli abstrakti syntaksipuu –pohjainen rakenneanalysointityökalu käytössä sekä opiskelijoilla että opettajilla, verkossa oleva palvelin
 - » CodeGrade ja semgrep
- » Arvioinnissa arviointityökalu
 - » Idea on käydä ohjelma läpi ja merkitä kaikki havainnot tarkistuslistaan
 - » Havainnoista saadaan virhetilastoja, laskettua arvosana ja annettua palautetta opiskelijalle
 - » Auttaa useiden assareiden palautteiden harmonisoinnissa, kurssilla paljon tuntiopettajia
 - » Uusi versio työn alla nyt dippana



KURSSILLA KÄYTETTÄVÄT TYÖKALUT 2

- » Tehtävien hallintaan Procasma
 - » Viikkotehtäviä 60 kpl, variantteja 6kpl, ts. noin 300 erilaista tehtävää, joista tehdään jokaiselle kurssille 60 tehtävän kokonaisuuksia
 - » Tehtävistä pitää olla tehtävänanto, malliajoja, esimerkkiratkaisu jne. sekä CodeGradeen testitapauksia
 - » Näiden hallinta on työlästä ja Procasma on tehty helpottamaan tätä




KESKITETTY TIETOKANTA TEKEILLÄ

- » Kurssin tietoja on useissa paikoissa: Moodle, EXAM, CodeGrade, ASPA, tilahallinto, arviointityökalu
- » Tietokannan käyttötarkoituksia
 - » Kerätä kaikki arviointiin vaikuttavat tiedot yhteen paikkaan
 - » Lähettää opiskelijoilla viikoittain palautetta edellisen viikon tehtävistä
 - » Saada dataa (pitkittäis)tutkimusta yms. varten
- » Opetukseen liittyvän tutkimuksen haasteita
 - » Tietosuoja- ja tietoturvanäkökulmat selvitettävänä
 - » Tietosuoja tahtoo kieltää kaiken julkaisemisen varmuuden vuoksi
 - » Julkaisukanavat vaativat lisääntyvässä määrin eettisiä yms. lausuntoja organisaatiolta jne.



MUUTA

- » Learning analytics – massakurssit hyviä tilastolliseen käsittelyyn
- » Palautteen antaminen
 - » Opiskelijat kaipaavat henkilökohtaista palautetta ja esim. luennon tai oppaan ohjeet eivät riitä
 - » Harjoitustyöstä saa palautetta, mutta tämä on käytännössä liian myöhään
 - » Palautetta pitäisi antaa viikoittain eli analysointi pitää automatisoida
 - » Samoin palautteen antaminen pitää automatisoida
 - » Käytännössä tarvitaan työkaluja kuten ASPA, arviointityökalu ja palautetyökalu sekä keskitetty tietokanta



LUT
University