

Kursuseprogramm

Ainekood IFI6074	PROGRAMMEERIMISE ALUSED		
Maht 4 EAP	Kontakttundide maht: 56, iseseisva töö maht 48	Õppesemester: S	Eksam
Eesmärk:	Luu eeldused programmeerimise olemuse, baasmõistete ja põhimeetodite mõistmiseks. Luu eeldused protseduurseks programmeerimiseks sobiva keele kasutamiseks ning tüüpiliste keeletarindite mõistmiseks. Aidata kaasa algoritmimisoskuse ning lihtsamate tüüpalgoritmide kasutamisoskuse kujunemisele. Luu eeldused väikesemahulise tarkvara loomise elutsükli sammude omandamiseks Toetada baasi kujunemist, mis on vajalik teiste programmeerimisega seotud ainete läbimisel.		
Aine lühikirjeldus: (sh iseseisva töö sisu kirjeldus vastavuses iseseisva töö mahule)	<p>Arv- ja tekstandmete esitamine arvutis. Arvusüsteemid. Üldotstarbelise programmeerimiskeele süntaks ja semantika. Lihtsamad algoritmid: summeerimine, loendamine, suurim ja vähim väärtus. Tegevusskeem algoritmide esitamiseks. Muutujad. Lihtandmetüübid. Struktuursed andmetüübid. Aritmeetika- ja loogikaavaldised. Omistamine. Sisend. Väljund. Juhtstruktuurid: jada, valik ja kordus ning vastavad keeletarindid. Alamprogrammid ja parameetrite edastamine. Tekstifailide kasutamine. Programmi koostamise tüüpilised sammud, testimine ja silumine.</p> <p>Aine toimub praktikatundidena arvutiklassis. Iseseisva tööna tuleb õpitu kinnistamiseks igal nädalal iseseisvalt lõpetada kontakttundides pooleli jäänud ja lisaks lahendamiseks antud ülesanded ning lugeda kursuse materjali, sh vastavaid peatükke õpikust.</p>		
Õpiväljundid:	<p>Kursuse edukalt läbinud üliõpilane:</p> <p>Tunneb protseduurse programmeerimisega seonduvaid põhimõtteid, mõisteid ja keelt ning programmeerimise olemust. Analüüsib lihtsamaid probleeme ning koostab nende lahendamiseks algoritme, kasutades sealjuures tüüpalgoritme. Esitab algoritme tegevusskeemi abil ning „tõlgib“ algoritmi programmeerimiskeelde, jälgides kodeerimise reegleid ja häid tavasid.</p> <p>Oskab programmikoodi töötlemiseks kasutada sobivat keskkonda, programmikoodi siluda ja testida.</p>		
Hindamismeetodid:	<p>Eksam toimub eksamisessioonil.</p> <p>Hinne kujuneb kirjaliku testi (mõisted, keelekonstruktsioonid, koodi mõistmine) – 30% ning eksamitöö (probleemi</p>		

	lahendamine, programmi koostamine ja testimine) – 70%põhjal.
Õppejõud:	Inga Petuhhov, õpetaja
Ingliskeelne nimetus:	Programming Fundamentals
Eeldusaine:	Eeldusaine puudub. Eeldatakse elementaarset arvuti kasutamise oskust.
Kohustuslik kirjandus:	Materjalid kursuse veebilehel: http://www.cs.tlu.ee/~inga/progbaas/ A. Downey. Think Python. How to Think Like a Computer Scientist http://faculty.stedwards.edu/mikek/python/thinkpython.pdf (ptk 1., 2., 3., 5., 7., 8., 10., 14.)
Asenduskirjandus: (üliõpilase poolt läbi töötatava kirjanduse loetelu, mis katab ainekursuse loengulist osa)	Kuna kursusel puudub selgelt eristuv loenguline osa, ei ole võimalik eraldi välja tuua asenduskirjandust loengute jaoks. Lugeda tuleks samu materjale, mis on esitatud kohustusliku kirjandusena.
Õppetöös osalemise ja eksamile/arvestusele pääsemise nõuded	Eksamile pääsemise eeldusena peab teoreetilisi teadmisi kontrolliv test olema positiivselt sooritatud (st testi eest on kogutud vähemalt 51% punktidest). Võlgnevuse likvideerimiseks saab testi uuesti teha peale kontaktõppe lõppu (14.-18.12.2015 – aeg täpsustatakse).. Praktikatundides osalemine on kohustuslik. Puudumise korral tuleb iseseisvalt uus osa omandada ja tunnis antud ülesanded lahendada enne järgmist tundi, sest see tagab järgmise tunni parema mõistmise. Õppetöö jooksul toimub kaks kontrolltööd, mille eesmärgiks on anda üliõpilasele tagasisidet tema senisest edenemisest. Et tööd ei ole arvestuslikud, ei toimu ka järeltöid, kuid igäühel on võimalus hiljem oma tööd korrigeerida, lõpuni lahendada ning soovi korral uut tagasisidet saada.
Iseseisva töö nõuded	Iseseisvaks tööks on igal nädalal praktikatunnis pooleli jäänud ning lisaks lahendamiseks antud ülesannete lõpuni lahendamine ning kursuse materjalide lugemine viidatud peatükkide ulatuses. Järgnevas tunnis on alati võimalik küsida kodus teha jäänud ülesannete kohta. Järgnevates tundides tehtavad ülesanded baseeruvad osaliselt eelmiste tundide töödele, mistõttu

	<p>iganädalane ülesannete lõpuni lahendamine on väga vajalik!</p> <p>Kohustusliku kirjanduse läbitöötamist kontrollitakse testi abil.</p>
<p>Eksami hindamiskriteeriumid või arvestuse sooritamiseks vajalik miinimumtase</p>	<p>Hindamiskriteeriumid, millest hindamisel lähtutakse (teoreetilised teadmised kontrollitakse kontrolltööga ning praktilised eksamitööga):</p> <p>1. Protseduurse programmeerimisega seotud mõistete tundmine</p> <p>A – Orienteerub mõistetes vabalt, seostab neid omavahel ja konkreetse keelega ning algoritmimisel ja kodeerimisel ettetulevate vajadustega.</p> <p>B – Seostab mõisteid omavahel ning oskab neid praktikas kasutada.</p> <p>C – Seostab mõisteid omavahel.</p> <p>D – Oskab tuua olulisemate mõistete kasutamise kohta näiteid.</p> <p>E – Oskab selgitada olulisemaid mõisteid.</p> <p>2. Protseduurse keele põhikonstruktsioonide ning andmetüüpide tundmine</p> <p>A – Tunneb protseduurse keele lauseid ja andmetüüpe, oskab neid õigel viisil kombineerituna kasutada, oskab selgitada nende toimimist. Tunneb keeletarindeid laiemalt ja oskab üldistada erinevatele keeltele. Saab aru võõrast lihtsamast programmist.</p> <p>B – Tunneb protseduurse keele lauseid. Suudab materjali abil tutvuda uute keelekonstruktsioonidega, funktsioonidega ja täiendavate andmetüüpidega ning neid sihipäraselt kasutada</p> <p>C - Oskab lugeda võõrast programmi ning selgitada selles kasutatavaid keelevahendeid. Valib muutujatele sobivad andmetüübid ja algoritmi jaoks sobilikud keelekonstruktsioonid</p> <p>D – Oskab muutujatele jaoks valida otstarbele vastavad andmetüübid. Kasutab programmis sihipäraselt peamisi keelekonstruktsioone.</p> <p>E – Ühe keele näitel tunneb protseduursetele keeltele omaseid tüüpilisi keelekonstruktsioone ning oskab selgitada nende toimimist. Tunneb peamisi andmetüüpe ja nende omadusi. Oskab neid kasutada jada, valiku ja tsüklit sisaldava algoritmi realiseerimiseks</p>

3. Programmikoodi kirjutamine. Lihtsamate tüüpalgoritmide (summa, loendamine, suurim ja vähim) kasutamine programmikoodis, nende sihipärane kasutamine probleemide lahendamisel.

A – Kombineerib programmikoodi kirjutamisel edukalt erinevaid algoritme, kasutamata sealjuures standardfunktsioone. Programm arvestab erinevate eri- ja veaolukordadega ning välistatud on täitmisaegsete vigade tekkimine kõigis programmi töö faasides.

B – Kombineerib programmikoodi kirjutamisel edukalt erinevaid algoritme, kasutamata sealjuures standardfunktsioone. Programm arvestab eri- ja veaolukordadega ning välistatud on olulisemate täitmisaegsete vigade tekkimine

C – Kasutab programmikoodi kirjutamisel mitut tüüpalgoritmi kombineeritult, kasutamata sealjuures standardfunktsioone. Lähtub lahendamisel erinevatest koodinäidetest.

D – Kasutab programmis mitut tüüpalgoritmi. Oskab muuta koodinäiteid nii, et nad lahendaksid teisi sarnaseid probleeme.

E – Mõistab koodinäidetes esitatut. Oskab muuta koodinäidet nii, et see lahendaks natuke erinevat ülesannet.

4. Probleemi analüüsimine ja lahendusalgoritmi koostamine

A – Oskab koostada algoritmid keskmise keerukusega probleemide lahendamiseks, kus ei saa läheneda standartseid võtteid kasutades. Algoritmid on koostatud optimaalselt ja ei raiska asjatult mälu ning aega

B - Oskab koostada algoritmid keskmise keerukusega probleemide lahendamiseks, kasutades sealjuures nii tüüpalgoritme, kui nende kombinatsioone. Algoritmid on koostatud optimaalselt ja ei raiska asjatult mälu ning aega.

C - Oskab koostada algoritmi lihtsama probleemi lahendamiseks, kasutades sealjuures mitut tüüpalgoritmi ja viimaseid modifitseerides.

D – Oskab koostada algoritmi lihtsama probleemi lahendamiseks, kasutades sealjuures mitut tüüpalgoritmi.

E – Oskab koostada algoritmi lihtsama probleemi lahendamiseks, mis eeldab ühe-kahe tüüpalgoritmi kasutamist.

5. Programmikoodi korrektsus ja reeglitele vastavus

A – Programmi kood on korrektselt kommenteeritud, muutujate nimed vastavad nende otstarbele ja headele tavadele ning on valitud süsteemselt. Kood on trepitud tavapärasel viisil ning on väga hästi jälgitav ja loetav

B – Programmi kood järgib üldiselt soovitusi ja tavasid ning on loetav nii struktuurilt kui sisult

C – Enamus programmi kood vastab soovitudele ja on üldiselt loetav tänu kommentaaridele

D – Programmi kood vastab soovitudele, kuid on üle- või alakommenteeritud. Muutujate nimed on küll mnemoonilised, kuid valitud üsna juhuslikult

E – Programmi kood vastab üldiselt soovitudele, kuid esineb puudujääke muutujate nimedes, kommentaarides jms.

6. Programmikoodi silumine

A – Oskab programmi siluda nii väljatrükkide abil kui ka programmeerimissüsteemi silumisvahendeid kasutades. Oskab enamuse süntaksivigu läbinäha ja likvideerida juba enne kompileerimist.

B – Kasutab programmi koodi silumiseks muutujate väljatrükke, oskab seda teha sihipäraselt ja leiab sel teel vigu. Leiab osa süntaksivigu enne kompileerimist.

C – Kasutab programmi koodi silumiseks muutujate väljatrükke. Süntaksi vead leiab keskkonna tagasiside abil edukalt.

D – Saab hakkama oma programmi silumisega, kuid see ei toimu alati sihipäraselt. Juhuslikkust esineb ka süntaksiveateadete tõlgendamisel.

E – Saab hakkama oma programmi silumisega, kuid see toimub tihti „katse-eksituse“ meetodil – ehk silumine on läbi mõtlemata. Juhuslikkust esineb ka süntaksiveateadete tõlgendamisel.

7. Programmi testimine

A – Koostab testid enamuse olukordade kontrollimiseks, samuti programmi käitumise hindamiseks veaolukordades.

B – Koostab testid loogikavigade leidmiseks, mis võivad

	<p>avalduda harva ja vaid teatud sisendite puhul. Kasutab testide leidmiseks ekvivalentsiklasside meetodit</p> <p>C – Koostab testid olulisemate vigade leidmiseks. Testid on koostatud ennekõike programmikoodi ülesehitusest lähtudes.</p> <p>D – Koostab testid ja testib oma programmi olulisemate veaolukordade leidmiseks.</p> <p>E – Testib oma programmi juhuslike sisenditega, kuid siiski sellistega, mis tagavad programmi korrektse töö juhul, kui veaolukordi ei teki ning läbitakse tüüpiline töövoog.</p>
	Täiendav informatsioon kursuse sisu kohta, kursuse jaotumine teemade kaupa sh seminarivormis toimuvate kontaktundide ajad
1. nädal (3.9.15, kell 8:15)	Sissejuhatus. Üldine tutvustus. Programmeerimine. Algoritm. Algoritmi üleskirjutamine – tegevusskeem. Jada, kui algoritmi osa. Jada ja paralleelsus tegevusskeemil.
1. nädal (3.9.15, kell 10:15)	Sissejuhatus Pythonisse. IDLE kasutamine. Muutujad, andmetüübid, aritmeetikaavaldised. Sisend ja väljund.
2. nädal (10.9.15, kell 8:15)	Kompileerimine ja interpreteerimine. Erinevad vead programmis: süntaksivead, täitmisaegsed vead ja loogikavead.
2. nädal (10.9.15, kell 10:15)	Lihtsad programmid jadaga.
3. nädal (17.9.15, kell 8:15)	Valik kui algoritmi ehituskivi. Võrdlustehed. Ülesanded valikutega. Valik tegevusskeemil. Valikulause Pythonis.
3. nädal (17.9.15, kell 10:15)	Ülesannete lahendamine.
4. nädal (24.9.15, kell 8:15)	Tsükkel kui algoritmi ehituskivi. Lihtsad algoritmid summeerimiseks ja loendamiseks.
4. nädal (24.9.15, kell 10:15)	Loogikaavaldised. Plokid programmis ja programmikoodi treppimine.
5. nädal (1.10.15, kell 8:15)	Andmete esitamine arvutis: bitt, bait, sõna. Arvandmete esitamine. Positsioonilised arvusteemid. Tekstandmete digitaalne esitamine. Peamised andmetüübid. Erinevad kooditabelid.
5. nädal (1.10.15, kell 10:15)	Ülesannete lahendamine.

6. nädal (8.10.15, kell 8:15)	Testimisest põhjalikumalt. Testide koostamine juba lahendatud probleemidele.
6. nädal (8.10.15, kell 10:15)	1. kontrolltöö
7. nädal (15.10.15, kell 8:15)	Struktuursed andmetüübid: massiivid / listid. Stringide kasutamine.
7. nädal (15.10.15, kell 10:15)	Massiivide (listide) kasutamine andmete töötlemisel. Tüüpalgoritmide (summeerimine, loendamine, suurim, vähim) kasutamine massiivis
8. nädal (29.10.15, kell 8:15)	Ülesanded suuremate andmehulkade töötlemisega. Tüüpalgoritmide kasutamine.
8. nädal (29.10.15, kell 10:15)	Massiivid (listid) – ülesannete lahendamine.
9. nädal (5.11.15, kell 8:15)	Tekstifailide kasutamine andmete hoidmiseks..
9. nädal (5.11.15, kell 10:15)	Ülesannete lahendamine
10. nädal (12.11.15, kell 8:15)	Funktsioonide loomine.
10. nädal (12.11.15, kell 10:15)	2. kontrolltöö
11. nädal (19.11.15, kell 8:15)	Alamprogramm: veidi teooriat. AP kui protseduurse programmeerimise alus. Protseuur. Funktsioon. Andmevahetus alamprogrammidega. Globaalsed ja lokaalsed muutujad ja skoop. Moodulid.
11. nädal (19.11.15, kell 10:15)	Ettevalmistus „mitmemõõtmelisuseks“.
12. nädal (26.11.15, kell 8:15)	Mitmemõõtmeline massiiv (list listis).
12. nädal (26.11.15, kell 10:15)	Ülesannete lahendamine: kahemõõtmeline andmestik + tekstifail +funktsioonide kasutamine.
13. nädal (3.12.15, kell 8:15)	Ülesannete lahendamine: kahemõõtmeline andmestik + tekstifail +funktsioonide kasutamine
13. nädal (3.12.15, kell 10:15)	Test mõistetest ja keelekonstruktsioonidest

14. nädal (10.12.15, kell 8:15)	Ülesannete lahendamine
14. nädal (10.12.15, kell 10:15)	Ülesannete lahendamine
14.12.15 – 18.12.15	Testi järeltöö. Kuupäev ja kellaaeg täpsustatakse

Õppeainet kureeriv üksus:	Informaatika instituut/ Digitehnoloogiate instituut
Kursuseprogrammi koostaja	Inga Petuhhov
Allkiri:	
Kuupäev:	15.08.2015

Kursuseprogramm registreeritud akadeemilises üksuses

Kuupäev	22.08.2015
Õppeassistendi nimi	Liina Kirsipuu
Allkiri	