

RTU studiju kurss "Datormācība (pamatkurss)"

33000 Datorzinātnes, informācijas tehnoloģijas un enerģētikas fakultāte

Vispārējā informācija

Kods	DE0201
Nosaukums	Datormācība (pamatkurss)
Studiju kursa statuss programmā	Obligātais/Ierobežotās izvēles
Atbildīgais mācītspēks	Artūrs Āboltiņš - Doktors, Profesors
Mācītspēks	Tatjana Solovjova - Docents (praktiskais)
Apjoms daļās un kredītpunktos	1 daļa, 5.0 kredītpunkti
Studiju kursa īstenošanas valodas	LV, EN
Anotācija	Studiju kurss iepazīstina bakalaura līmeņa studiju pirmā kursa studentus ar mūsdienu datoriem, populārām operētājsistēmām, elektronikas un telekomunikāciju nozarēs bieži izmantojamām programmēšanas valodām, ērtām programmu izstrādes vidēm un elementāriem skaitļošanas algoritmiem, kas ir noderīgi turpmākajos studijuursos un inženiera darbā. Tiek apskatīti datu iegūšanas, datu apstrādes un attēlošanas sistēmu matemātiskie, inženier-tehniskie un filozofiskie principi, kas kalpo profesionālas izpratnes un praktisku iemaņu veidošanai darbā ar profesionālām datorsistēmām, kas bāzētas uz atvērtā koda operētājsistēmu "Linux", konkrēti uz Ubuntu realizācijas. Studiju kursa klausītāji iepazīstas ar klasiskajām programmēšanas valodām C, C++ un moderno valodu Python. Tiek apgūtas binārās un sedecimālās skaitļošanas sistēmas. Padziļināti tiek apskatīti jautājumi, kas saistīti ar jēdzieniem "bits", "baits" un "datu tips". Praktiski veicamie programmēšanas vingrinājumi ir saistīti ar elementāriem fizikas, elektrotehnikas un mehānikas kursa uzdevumiem, kā arī ar augstākās matemātikas skaitļu virkņu tēmu, diferencēšanu un funkcionāli veidotu ģeometrisku figūru laukumu skaitļošanu. Ievērojot mācību profila specifiku, tiek pildīti vingrinājumi, kas ir saistīti ar kompleksiem skaitļiem, bitu operācijām un datu masīvu kārtošanu.
Mērķis un uzdevumi, izteikti kompetencēs un prasmēs	Studiju kursa mērķis ir sniegt pārskatu par C, C++ un Python programmēšanas valodām un brīvpieejas tehniskajiem līdzekļiem elementāru, bet profesionāli orientētu programmu veidošanai. Studiju kursa uzdevumi: 1. Iepazīstināt ar mūsdienīgām, profesionālām un efektīvām programmēšanas vidēm balstītām uz Linux operētājsistēmu, liekot lielu uzsvāri uz darbu komandringas režīmā. 2. Dot iespēju gūt praktisku pieredzi vienkāršu C++ un Python valodās rakstītu programmu veidošanā, algoritmizējot vienkāršus fizikas un matemātikas uzdevumus. 3. Attīstīt procedūrpogrammēšanas prasmes. 4. Iemācīt datu iegūšanas, apstrādes un attēlošanas vienkāršākos paņēmienus.
Patstāvīgais darbs, tā organizācija un uzdevumi	Studenti patstāvīgi analizē studiju kursa literatūru, veic mājasdarbus un izpilda programmēšanas uzdevumus, gatavojas laboratorijas darbiem, noformē laboratorijas darbu atskaites, gatavojas eksāmenam.
Literatūra	Obligātā/Obligatory: [1] Ziemelis J. Ievads algoritmu valodā C. Rīga, RTU, 2006. Papildu/Additional: [2] Stroustrup, B. The Programming Language C++. Addison-Wesley, 2009. [3] The C++ Resources Network. Available: http://www.cplusplus.com/ [4] Elkner, J., Downey, A.B., Meyers, C. How to Think Like a Computer Scientist, Learning with Python. 2nd ed. 2008. Available: http://www.openbookproject.net/thinkcs/python/english2e/ [5] Elkner, J., Downey, A.B., Meyers, C., McCane, B., Hewson, I., Meek, N. Practical Programming in Python. 2009. Available: http://www.cs.otago.ac.nz/staffpriv/mccane/Downloads/PracticalProgramming.pdf [6] Knuth, D. All questions answered. Notices of the American Mathematical Society, vol. 49(3), 2002, p.318–324. [67] Herrmann, D. C++ für Naturwissenschaftler: beispielorientierte Einführung. Addison-Wesley, 2001. [8] Shader, M., Kuhlins, S. Programmieren in C++: Einführung in den Sprachstandard. Springer, 1998. [9] Dausmann, M., Broeckl, U., Goll, J. C als erste programmiersprache. Teubner, 2005. [10] Zagorskis, V. Datormācība-pamatkurss. Laboratorijas darbu apraksti. Rīga: RTU, ETF, SC, 2010. [11] Zagorskis, V. Datormācības lasījumi. I. daļa. Rīga: RTU, ETF, SC, 2009.
Nepieciešamās priekšzināšanas	Pamatzināšanas informātikā, zināšanas matemātikā vidusskolas programmas līmenī, prasme apkopot un analizēt informāciju.

Studiju kursa saturs

Saturs	Pilna un nepilna laika klātienē studijas		Nepilna laika neklātienē studijas	
	Kontakt stundas	Patstāv. darbs	Kontakt stundas	Patstāv. darbs
Cilvēks un Dators. Atšķirības uz līdzības. Mijiedarbības filozofija. Informācija. Dati. Bits. Baits.	1	1	0	0
Operētājsistēmas. Konceptija. Nepieciešamība. Uzbūve. Vēsture un šodienas	1	1	0	0
UNIX sistēmas. Laikā sadalītu uzdevumu izpilde. Vēsture un šodienas. Linux.	1	1	0	0

Operētājsistēmas kodols un čaulas. Datora fiziskās ierīces un to sadarbība ar sistēmas kodolu. Failu sistēmas.	1	1	0	0
Linux lietotāji. Termināls. Shell (čaulas). To veidi un komandas. Mainīgie un faili. Tiesības.	2	2	0	0
Elementāru datu iegūšana, apstrāde un uzglabāšana Linux vidē. Shell programmēšana. Shell scenāriji.	2	2	0	0
C un C++ programmēšanas valodu elementi. Sintakse. Programmas teksts. Operatori. Operācijas. Operandi.	2	2	0	0
Funkcija main(). Programmas teksta kompilēšana. Izpildāmu programmas failu iegūšana.	2	2	0	0
Izpildfaila izsaukšana. Rezultāta novērtēšana. Datu izvade. Izvades ierīču veidi. Standartizvade.	2	2	0	0
Iebūvētās C, C++ funkciju bibliotēkas. Viena baita datu tips Char. Mainīgā deklarēšana, deklarēšana, definēšana.	2	2	0	0
Skaitļu aplis. Naturāli un veseli skaitļi. Simboli. ASCII tabula. Aritmētiskās operācijas. Operāciju prioritāte.	1	1	0	0
Datu pārveidošana. Algoritmu jēdziens. Algoritms I. Kā decimālu skaitli konvertēt par bināru.	2	2	0	0
Veselu skaitļu datu tipi: short, int, long. Bibliotēkas fails limits.	2	2	0	0
Reālus skaitļus realizējošie datu tipi: float, double, long double. Datu masīvi.	2	2	0	0
Lietotāja definētas funkcijas. Funkcijas prototips. Programmas bloks.	2	2	0	0
Ciklisku aprēķinu veikšana. Operatori For un While. Cikla mainīgie. Cikla ķermenis.	2	2	0	0
Algoritms II. Faktoriāla aprēķins. Algoritms III. Kā bināru skaitli konvertēt par decimālu.	2	2	0	0
C, C++ iebūvēto matemātisko funkciju bibliotēku pārskats.	1	1	0	0
Datu tips Boolean. Programmas gaitu regulējošās struktūras - if, else (0.5) Bitu operācijas.	2	2	0	0
Datu masīvi. Norādes. Norāžu aritmētika. References. Lietotāja definēti datu tipi: struktūras, apvienības, u.c.	2	2	0	0
Formatēts datu ieraksts failā un nolasīšana no tā ar C un C++ palīdzību. Teksta un binārie faili.	2	2	0	0
Python programmēšanas valodas elementi. Aritmētikas operācijas. Datu izvade uz ekrāna. Datu ieraksts failā ar Python.	2	2	0	0
Python datu tipi. Veselie skaitļi, reālie skaitļi, kompleksie skaitļi. Sistēmas pārvaldes funkcijas aprēķins ar Python.	2	2	0	0
Saraksta (list) datu tips. Tuple datu tips. Cikli Python valodā. Datu nolasīšana no faila ar Python palīdzību.	2	2	0	0
Algoritms IV. Faktoriāla aprēķins. Algoritms V. Skaitļu virknes. Rekurence.	2	2	0	0
Python standarta bibliotēkas (sys, os, math, cmath, numpy, u.c.). Python lietotāja definētas funkcijas.	2	2	0	0
Algoritms VI. Funkcijas saknes atrašana ar dihotomijas metodi.	2	2	0	0
Programmas gaitu regulējošās struktūras - if, elif, else.	2	2	0	0
Algoritms VII. Datu kārtošanas algoritmi. Burbuļa metode.	2	2	0	0
Algoritms VIII. Funkcijas diferenciāla aprēķini.	2	2	0	0
Algoritms IX. Funkcionāli aprakstītu ģeometrisku figūru laukuma aprēķins. Trapeču metode.	2	2	0	0
Speciāla, zinātnisko skaitļojumu (funkciju) Python bibliotēka - numpy. Līdzeklis GNUPLOT.	2	2	0	0
Alternatīvas zinātniska rakstura programmpakas (Octave, Maxima, u.c.).	1	1	0	0
Elementārie statistiskie aprēķini.	1	1	0	0
Kopā:	60	60	0	0

Sasniedzamie studiju rezultāti un to vērtēšana

Sasniedzamie studiju rezultāti	Rezultātu vērtēšanas metodes
Prot lietot profesionālās UNIX veida operētājsistēmas termināla režīmā. Ir iegūta pieredze darbā ar direktorijām un failiem. Prot strādāt ar teksta redaktoriem, izsaukt programmas un skriptus.	Mājasdarbi ar novērtējumu Testi. Eksāmens.
Spēj veidot elementāras C, C++ procedūrveida programmas, kas ietver ciklus un zarojumus. Ir zināšanas kā uzrakstīto programmu kompilēt un novērst sintakses kļūdas. Pazīst C valodas datu tipus.	Programmēšanas uzdevumi ar novērtējumu. Laboratorijas darbi ar novērtējumu. Testi. Eksāmens.
Prot veidot samērā vienkāršas Python programmas, kas spēj realizēt dažus vienkāršus skaitliskus algoritmus kā, piemēram, funkcijas saknes atrašana, diferencēšana un integrēšana.	Programmēšanas uzdevumi ar novērtējumu. Laboratorijas darbi ar novērtējumu. Testi. Eksāmens.
Prot strādāt ar programmu GNUPLOT, kas ļauj pētīt divu dimensiju funkcionālās sakarības.	Laboratorijas darbi ar novērtējumu. Eksāmens.

Studiju rezultātu vērtēšanas kritēriji

Kritērijs	% no kopējā vērtējuma
Apmeklējums	10
Testi	5
Programmēšanas uzdevumi	25
Mājas darbi	10
Laboratorijas darbi (kodi)	10
Laboratorijas darbi (aizstāvēšana)	15
Eksāmens	25
Kopā:	100

Studiju kursa plānojums

Daļa	KP	Stundas			Pārbaudījumi		
		Lekcijas	Prakt d.	Laborat	Ieskaite	Eksām.	Darbs
1.	5.0	30.0	0.0	30.0		*	