

Magas szintű programozási nyelvek 1, INDK301

Félév: páros, **Típus:** Előadás/Labor, **Óraszám/hét:** 2+2, **Kredit:** 5, **Státusz:** Kötelező szakmai

Oktató: Dr. Bátfai Norbert

Oktatási módszer: Előadás/Labor, **Előfeltételek:** Bevezetés az informatikába

Vizsgáztatási módszer: írásbeli teszt vagy szóbeli vizsga, mindkettőn **bármilyen használható**, de a számítógép használata mindenképpen szükséges! Az írásbeli gép nélküli előadóban történik (a hallgató hoz gépet, a feladatsort elektronikusan kódolva, titkos szöveggént kapja, amelyet törnie kell, lásd a „hagyományainkat”) a szóbeli vizsga gépes laborban történik, alapja a tételhúzás, de itt is lehet a saját gépet használni természetesen. A mindenkori aktuális beugró- és tételsor¹ az UDPROG² évkönyvben található. Egy tétel beugró része gyakorlati feladat, fő része pedig a kötelező olvasmány tankönyv adott oldalszám tartománya.

Pontverseny: a hallgatóknak a félév során önálló feladatokat kell végezniük, akik adott határidőig a legtöbb pontot érik el, azok közül az első **20** (sorrendben 5 jeles, 7 jó, 8 közepes) hallgató kap jegymegajánlást, illetve a szerzett pontok a jegymegajánlásba be nem kerülő hallgatók szóbeli tételsorát is szabályozzák, lásd a részleteket az UDPROG évkönyvben (a szóbeli tételhúzáskor a megfelelő kedvezmények azt szabályozzák, hogy adott pont felett hány tételt lehet elkerülni) Az előadások látogatása (max. két hiányzás) 5%-ot ér az írásbeli vizsgán.

A pontverseny feladatai részben az UDPROG évkönyvben találhatóak, ebben van leírva a pontszerzés pontos szabályozása is. A hallgatóknak ebben az évkönyvben kell adminisztrálni a feladatok elvégzését, illetve egy a tárolóban lévő fájlban (book/points2017tavasz.xml) folyamatosan aktualizálni a pontjaik számát és az arról szóló nyilvános könyvelés URL címét, utóbbi azért fontos, mert a pontjait mindenki elosztott és **nem megbízható** módon magának számolja, ezért mindig számot kell tudni adnia arról, hogy **mikor-mire-mennyi** pontot kapott

A pontverseny feladatainak zöme az UDPROG évkönyvben található ebben van leírva a pontszerzés pontos szabályozása. Továbbá tematikusan két területet támogatunk kiemelten az önálló feladatoknál, ezek a következők:

- OOCWC (közlekedés szimulációs) platform, <https://github.com/nbatfai/robocar-emulator>
- Entrópia Samu (e-sport és gépi tanulás), <https://github.com/nbatfai/SamuEntropy>

A csoportmunka támogatása: a tutor rendszerben elszámolt pontokat a tutoriált 1.5x-es szorzóval számolhatja el, a tutor pedig a következő képlettel:

$$\sum_{i=1}^{\text{tutoriáltak száma}} \frac{1.5}{i} * a \text{ feladat pontszáma.}$$

A korábbi félévekről hozott pontok a pontszám -500 „képlettel” hozhatók át. Ha nyoma van (pl. Github-ba felkommitolt korábbi védés) akkor a korábbi sikeres védéseket nem kell megismételni, ha nincs nyoma, akkor a védést meg kell ismételni, de természetesen tartalmában az lehet a korábbi.

Kompetencia: A kurzus sikeres teljesítése esetén a hallgatók képesek lesznek C/C++ programok olvasására, írására.

Néhány széles spektrumon terített speciális feladaton keresztül (esetlegesen) rálátást kaphatnak olyan területekre, mint a Linux/Minix kernel, Qt, OpenMP, CUDA, Boost C++, Boost BGL, OpenStreetMap, libosmium, ágens programozás, neurális háló, backpropagation, deep Q-learning, gépi tanulás, Tensorflow vagy akár a humán genom feldolgozása.

¹ Az aktuális beugró- és tételsor vizsgaidőszak kezdete előtt pár héttel véglegesítődik.

²<http://sourceforge.net/projects/udprog/>, generáld le a DocBook XML-ből a mindenkori aktuálisat!

Tárgyleírás:

A kurzus elsődleges szervezése a Facebook-on történik: ez a hallgatókkal való közös munka legfontosabb eszköze a kurzusban, itt találhatóak szakmai és szervezési beszélgetések vagy akár a hallgatói kérdések/válaszok, hasonló informális szerepet tölt be a FB csoportunk: <https://www.facebook.com/groups/udprog>

A kurzus szervezése másodlagosan a LinkedIn-en történik a The Yearbook of the Programmers of University of Debrecen című szakmai csoportban: <http://www.linkedin.com/groups/Yearbook-Programmers-University-Debrecen-7446358>

~~A hallgatók a laborközösségben binomokba dolgoznak, így dolgozzák fel a hallgatói laborkártyákat. Továbbá létezik a laborkártya intézmény, azaz egy ilyen kérdés nem tudása egy fél hiányzást jelent a laborkatalógusban és -30 pontot a pontversenyben. Immár -200 pont a tároló „elrontása”!~~

+3000 pont annak, aki bekerül a Google Summer of Code 2016-ba! (**+800** pont annak, aki bead proposal-t)

Előzetesen várható tartalmi témák:

1. C bevezetés

1. (logikailag) Előadás (az érdeklődés felkeltése)
 - a) A kurzus teljesítésének feltételei, szabályai
 - b) Általános kép adása a programozásról
 - c) C nyelvi bevezetés: a C nyelv, típusok, vezérlési szerkezetek, mutatók, deklarációk, kifejezések, függvények, paraméterátadás.

Labor (saját munkastílus kialakítása)

- a) szövegszerkesztő használata, forráskód indentálása
- b) gcc, g++, cmakehasználata parancssorból
- c) a manuál lapok használata (Linux programmer's Guide alapok)
- d) a PP25-31. oldal példáinak letöltése, kipróbálása, megbeszélése a laborvezetővel.

Laborkártyák

- a) Forrás és deklarációs kártyák

Otthoni opcionális feladat

- a) GNU/Linux rendszer telepítése, s a japán világbajnok HELIOS csapat szoftvereinek otthoni installálása (rcssserver, rcssmonitorstb.)

2. C tárgyalás

- a) Kolmogorov bonyolultság, véletlen számsorozat
- b) Élettartam és hatáskör (érvényességi tartomány)
- c) Mutatók és több dimenziós tömbök, mutatóaritmetika, dinamikus tárkezelés.

Labor

- a) Egy saját PageRankimplementáció
- b) EXOR törés
- c) Hatáskör „megbolondítása” a forkrendszerhívással
- d) A crashme.cforrásának a laborvezető által celebrált átnézése (csak a jelkezelés szempontjából lényegi működés)
- e) A PP 36-40 oldal példáinak letöltése, kipróbálása, megbeszélése a laborvezetővel (forkrendszerhívás és társai).

Laborkártyák

- a) Forrás, mutatós és operátoros kártyák

Otthoni opcionális feladat

- a) A japán világbajnok HELIOS csapat szoftvereinek otthoni

installálása (librsc, agent2D, soccervindowstb.)

3. C befejezés

- a) Struktúrák, önhivatkozó struktúrák. Állománykezelés.
- b) Bináris fák kezelése, Lempel-Ziv-Welch(LZW) algoritmus
- c) GNU/Linux PCB, listakezelés

Labor

- a) Binárisból karakteres „dump” írása
- b) LZW fa építése
- c) „Saját top” parancs megírása, PP 89
- d) A PP 173-oldal példáinak megbeszélése a laborvezetővel (kernel modulok, rendszerhívások).
- e) Szálak, jelek bevezetése

4. C a gyakorlatban

- a) IPC („azt mondja az egyik program a másiknak...”)
- b) Klasszikus IPC problémák és megoldásukEbédelő filozofokTermelő és fogyasztóOlvasók és írók
- c) Dijkstra-féleszemaforok

Labor

- a) System V és POSIX szemaforok, üzenetsorok,osztott memória, Lokális, anonim és TCP/IP socketek, csővezetékek bevezetése

Labor

- a) Példás kártyák

Otthoni opcionális feladat

- a) A japán világbajnok HELIOS csapat szoftvereinek otthoni tanulmányozása.

5. C++ bevezetés

- a) Referenciatípus
- b) OO alapelvek
- c) Dinamikus tárkezelés

Labor

- a) BogoMIPS
- b) Qt (fraktálok, genom ábrázolások, életjáték)

Labor

- a) Példás kártyák

Otthoni opcionális feladat

- a) A japán világbajnok HELIOS csapat szoftvereinek otthoni tanulmányozása.

6. C++ tárgyalás

- a) Osztályok, objektumok
- b) Másoló konstruktor, -értékdadás, baráti, beágyazott osztályok
- c) Öröklődés, a Liskov-féle helyettesítési elv
- d) Virtuális függvények
- e) C és C++ összehasonlítás

Labor

- a) Qt slot-signal mechanizmus
- b) Visszatekintés: hálózati állatorvosi és a deriváló ló

Labor

- a) Példás kártyák

Otthoni opcionális feladat

- a) A japán világbajnok HELIOS csapat szoftvereinek otthoni tanulmányozása.

7. C++ a gyakorlatban

8. C++ befejezés

9. Java platform bevezetés

A további kifejtés és maguk a fóliák a fészes csoportban kerülnek megosztásra.

A témák finomítását, hangsúlyainak eltolását tervezzük, például a C++11/C++14 szabvány középpontba emelésével.

Labor teljesítés: katalógus hiányzás ≥ 3.5 (laborkártya miatt tört) esetén nincs aláírás. A labor teljesítésének további szükséges feltétele ~~három~~ **két** „saját” program sikeres bemutatása és megvédése a laborközösség előtt. A feladatok kötöttek: a humán genomos elemzés vagy robotautó VB és robotautó VB vagy valamely Samu projekt; a védendő feladatok pontos kiírását lásd majd a kurzusban!

A labor teljesítésének további szükséges feltétele legalább **500** pont összegyűjtése a pontversenyben

Etikai elvárások: A vizsgákon bármi használható, de a (verbális, elektronikus) kommunikáció tilos. A hallgatókkal kapcsolatos etikai normákra A DEBRECENI EGYETEM ETIKAI KÓDEXE az irányadó lsd.: <http://www.unideb.hu/portal/hu/node/47> :A Debreceni Egyetem Etikai Kódexe (Vizsgakódex).

Az etikai normákat megsértők:

- Automatikusan elégtelent kapnak a tárgyból
- A hallgató neve az IK oktatói között nyilvánosságra kerül.

Fogadó óra: szerda 14-15, kedd 11-12, Helye: I228.

Irodalom:

A kötelező olvasmány tankönyv:

- Stroustrup: **A C++ programozási nyelv**, Bp., Kiskapu, 2001
<http://webpac.lib.unideb.hu/WebPac/CorvinaWeb?action=onelong&showtype=longlong&recnum=255516&pos=6>
- Stroustrup: **The C++ Programming Language** (4th Edition)
<http://www.stroustrup.com/4th.html>

Szintén kötelező:

- Kernighan, Brian W.: **A C programozási nyelv** : az ANSI szerint szabványosított változat / Brian W. Kernighan, Dennis M. Ritchie : [ford. Molnár Ervin] 5. kiad. Budapest : Műszaki Kvk., 2006

Ajánlott:

- Benedek Zoltán: **Szoftverfejlesztés C++ nyelven** / Benedek Zoltán, Levendovszky Tihamér Budapest : Szak K., 2007

Néhány példa itt van kidolgozva:

- Bátfai Norbert: **Párhuzamos programozás GNU/Linux környezetben: SysV IPC, P-szálak, OpenMP**, Typotex Kiadó,
http://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0063_01_parhuzamos_prog_linux/adatok.html
- Bátfai Norbert: **Programozó Páternosztér újratöltve**, Debreceni Egyetem
<http://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011->

- [0103_23_programozoi_paternoszter_ujratolte/adatok.html](#)
- UDPROG repó: <https://sourceforge.net/p/udprog/code/ci/master/tree/>

Debrecen, 2017. február 23.