

TANTÁRGY ADATLAP

és tantárgykövetelmények

Cím:	Programozás I.
Tárgykód:	<i>IVB332MN</i>
Heti óraszám ¹ :	<i>1 ea, 2 gy</i>
Kreditpont:	<i>3</i>
Szak(ok)/ típus ² :	<i>K</i>
Tagozat ³ :	<i>N</i>
Követelmény ⁴ :	<i>f</i>
Meghirdetés féléve ⁵ :	<i>os</i>
Nyelve:	<i>Magyar</i>
Előzetes követelmény(ek):	-
Oktató tanszék(ek) ⁶ :	<i>Rendszer és Szoftvertchnológia Tanszék (100%)</i>
Tárgyfelelős:	<i>dr. Szabó Levente egyetemi adjunktus</i>
Tantárgy koordinátor, oktató	<i>dr. Szabó Levente egyetemi adjunktus</i>
Célkitűzése: A C programozási nyelv megismerésén és elsajátításán keresztül az alapvető programozási szerkezetek és technikák megtanulása.	
Rövid leírás: C nyelv alapjai, Függvények és változók, adattípusok, Szabványos I/O, Kifejezések, Vezérlő szerkezetek, Rekurzio, Tömbök és karakter tömbök, Mutatók, Mutatótömbök, Felhasználó által definiált adattípusok, Struktúrák, Struktúratömbök, C preprocessor, Dinamikus memóriakezelés, Aritmetikai műveletek precedenciája és asszociativitása, File-ok kezelése, Hibakeresés és hibakezelés, Magas szintű adatszerkezetek, Parancssori argumentumok, Szabványos ANSI C könyvtári függvények, Munka Gitlab-bel	
Oktatási módszer: Előadáson az elméleti alapok bemutatása, gyakorlatokon minél több, önálló kódolás.	
Követelmények a szorgalmi időszakban: A félév során, a gyakorlaton nyújtott teljesítmény értékelése félévközi jegyen keresztül történik, mely jegy értéke kettő dolgozat eredményének számtani átlaga alapján alakul ki. Ezek a dolgozatok a laborok számítógépein kerülnek megírásra a 7. és a 15. hét folyamán. Amennyiben a hallgató a dolgozat írásakor nem megengedett eszközöket használ, puskázik, csal, a dolgozatírást azonnal be kell fejeznie, és 0 ponttal értékelem a dolgozatát. Ilyen esetben a dolgozat nem is pótolható. A dolgozat alatt mobiltelefont vagy egyéb IPOD, stb eszközt használni nem lehet és még kikapcsolt állapotban sem lehet a padon. Jegyet csak az a hallgató kaphat, akinek esetében a dolgozatokra adott pontok számtani átlaga eléri az 51%-ot. Fontos tudni, hogy – a kettő ZH-val elért 51%-os számtani közép mellett – a 2. ZH-n el kell érni legalább 41%-ot! Azok a hallgatók, akiknek a féléves jegye legalább elégséges, lehetőséget kapnak jegyük javítására. Jegyet javítani saját ötlet alapján megfogalmazott feladat megoldásával, dokumentálásával és bemutatásával lehet. A feladat megoldását a dokumentációval együtt a bemutatás időpontja előtt 48 órával be kell adni. Az elkészített feladat bemutatása nyilvánosan, a hallgatótársak előtt történik. Időpontja a vizsgaidőszak első hetének végére várható, de ezt később még pontosítjuk. Ezzel, a megoldás minőségétől függően legfeljebb egy jegyet lehet javítani.	

¹ Tárgykurzus típusok: ea – előadás, gy – gyakorlat, lab – labor

² K – kötelező, KV – kötelezően választható, SZ – szabadon választható (fakultatív)

³ N – nappali, L – levelező, T – táv

⁴ a – aláírás, f – félévközi jegy, v – vizsga, s – szigorlat

⁵ os – őszi, ta – tavaszi

⁶ Több tanszék esetén zárójelbe a terhelés várható százalékos megoszlása

<p>A jegy kialakítása: 0-50% elégtelen (36-50% javíthat) 51-62% elégséges 63-74% közepes 75-86% jó 87-100% jeles</p>
<p>Követelmények a vizsgaidőszakban: Javítás a vizsgaidőszakban: Ha valakinek a félévközi teljesítménye a 36–50%–os (zárt) tartományba esett, akkor írhat egy javító ZH-t a félév teljes anyagából. A végleges gyakorlati jegy a két félévközi gyakorlati ZH átlagának és a javító ZH–nak az átlagából képződik az alábbi módon: $\text{gyakjegy} = ((zh1+zh2)/2 + \text{javitó})/2$ A javítási (jegypótlási) lehetőség időpontját, a vizsgaidőszak előtt egy héttel jelölöm ki. Pótlás a vizsgaidőszakban: A nem megírt dolgozat 0–s eredménnyel számít bele az átlagba. Ha valaki valamelyik ZH–t nagyon indokolt ok miatt nem tudja megírni, azt lehetőleg előre jeleznie kell, de legkésőbb az akadály elhárulása utáni gyakorlaton igazolnia kell. Ez esetben az utolsó héten egy külön időpontban pótolhatja az elmaradását. A pótló dolgozat a teljes félév anyagát tartalmazza, függetlenül attól, hogy mely dolgozatot nem írta meg a hallgató. Igazolást a hiányzást követő első órán való megjelenés időpontja után nem áll módomban elfogadni. Ez esetben az illető csak akkor élhet a jegypótlási lehetőséggel, ha az egyetlen megírt dolgozatának eredménye eléri a pótlási határ dupláját.</p>
<p>Pótlási lehetőségek: Lásd fentebb!</p>
<p>Konzultációs lehetőségek: Konzultáció biztosított minden fogadóórán.</p>
<p>Jegyzet, tankönyv, felhasználható irodalom: Pere László: Programozás C nyelven - UNIX – GNU/Linux (Kiskapu, 2003)</p>
<p>Ajánlott szakirodalom: Kernighan B.W., Ritchiw D.M.: A C programozási nyelv (Műszaki Könyvkiadó, 1985) Benkő T.-né, Benkő L., Tóth B.: Programozunk C nyelven! (ComputerBooks, 2000)</p>

Hét	Gyakorlat
1.	Hello world program megírása. C programok lefordítása, linkelése és futtatása. A printf, scanf függvények használata, formátum karakterek, speciális karakterek. Változók és függvények. A legegyszerűbb függvény megírása és használata. A void függvény típus szerepe. A return kifejezés. Érték szerinti paraméter átadás szemléltetése. Blokk, függvény, file és globális érvényességi körök szemléltetése programokkal. A static kulcsszó szerepe.
2.	Programok írása a különböző vezérlő szerkezet szemléltetésére. Az if utasítás. Az else elágazás használata. Többszörös elágazás if és switch utasítással. A default és break szerepe a switch utasításban. A while és do-while vezérlőszerkezetek. A for ciklus. A break és continue használata cikluson belül. Innentől, minden gyakorlat, minden programját bontsuk függvényekre! Magyarazzuk el annak az okait is, hogy miért is írunk függvényeket!
3.	Ciklus, elágazás, szekvencia. Kombináció. Összetett vezérlési szerkezetek, összetett kifejezések. Egyszerű algoritmusok pszeudokód / C változatai. Algoritmizálás gyakorlása. Matematikai, relációs, logikai, növelő, csökkentő műveletek használata a precedencia-táblázat alapján. Értékadás, összehasonlítás, léptetés, stb. A zárójelezés szerepe. Implicit és explicit típuskonverziók szerepe a programokban. Elemi adattípusok, változók mérete (túlcsordulás).

4.	Rekurzió. További gyakorló feladatok az eddigiekre, elmélyítendő a megszerzett ismereteket.
5.	Tömbök és függvények használata, tömb átadása függvénynek. Sztringes feladatok. Tömbbel megoldható feladatok gyakorlása. Leendő zh-feladatok gyakorlása, függvényekkel való munka skill-jeinek mélyítése, programok több állományra való bontása.
6.	Szám tömb létrehozása. Számok beolvasása egy tömbbe. Tömb tartalmának kiírása. Indexelés. Egyszerű programok mutatókkal. A mutatókkal elkövethető hibák szemléltetése. Mutató követése (dereferencia), típuskényszerítés, a NULL pointer szerepe. Mutatók és tömbök. Változók tartalmának cseréje függvényvel.
7.	1. zárthelyi
8.	Őszi szünet
9.	1. zárthelyi feladatainak megoldása, felmerülő tanulságok levonása.
10.	Rekurzió, dinamikus adatszerkezetek. Karakterláncok. Szövegek létrehozása. Szabványos szövegkezelő függvények: strlen, strcpy, strcat, strcmp, stb. A szabványos szövegkezelő függvények megírása mutatókkal. Gyakorló feladatok, karakterláncok kezelésére.
11.	Struktúra definiálása, pl: személy. A struktúra feltöltése és tartalmának kiírása. Struktúrák egymásba ágyazása, pl. dátum is egy struktúra a személy struktúrában. Struktúrát jelölő mutatók. Új típusok létrehozása. Struktúratömbök, pl. több személy definiálása. Mutatót jelölő mutatók. Mutatótömbök.
12.	Makrók definiálása C programban. Konstansok, mint makrók. A malloc, calloc, realloc és free függvények használata tömböknél és struktúratömböknél.
13.	Szöveges file megnyitása és fájlba nyomtatás. A file visszaolvasása. Sorok formátum szerinti visszaolvasása fscanf függvényvel. Egyéb módszerek adatok visszaolvasására formátum szerint. Az stdin, stdout és stderr file-ok használata konzolos programoknál. Fájl átmásolása másik fájlba, stb.
14.	A main függvény paraméterei. Önmagukra hivatkozó adatszerkezetek (láncolt listák, veremek) kezelése.
15.	2. zárthelyi

A zárthelyik százalékos értékelése úgy van megkoreografálva, hogy csak az a hallgató kaphat elégségest a tárgyból, aki képes a második zh-n, 90 perc alatt, az alábbiakra:

Megír egy programot, mely program számára – azt parancssorból indítva – a main függvény paraméterein keresztül megadhatók a futás során feldolgozandó szöveges állományok nevei. A függvényekre tagolt program, az egyik állományból való beolvasás által feltölt egy dinamikus helyfoglalású struktúratömböt (listát), majd az ott található adatoknak a feladat szerinti feldolgozása után, a kapott eredmény(ek)e)t kiírja egy szöveges fájlba, ami lehet a beolvasott állomány is természetesen. Közepest, jót, jelest az kaphat, aki a felsoroltakon felül képes még az adott 90 percnyi időtartam keretén belül egyéb, magasabb szintű gondolkodási készséget feltételező algoritmizálási problémák kezelésére, s a kapott eredmények adott fájlhoz való hozzáfűzésére, illetve a program több állományra való szétbontására is.

2018.09.03.

dr. Szabó Levente
tantárgyfelelős