

# Programozás II.

## 1. gyakorlat

# Elérhetőségeim

- Antal Gábor
- Email: [antal@inf.u-szeged.hu](mailto:antal@inf.u-szeged.hu)
- Weblap: <http://inf.u-szeged.hu/~antal>
- Fogadóóra:
  - Hamarosan (előzetes egyeztetés alapján bármikor)
  - A gyakorlatok időpontjában
- Gyakorlattel kapcsolatos email-ek **csak** studios e-mail címről

# Követelmények

- 2 kötelező program (**Igen/Nem**)
  - Előre megadott listából kap majd mindenki feladatot
  - Első feladat
    - *Beadás:* október 15. 08:00, Coospace
  - Második feladat
    - *Beadás:* november 19. 08:00, Coospace
- Gyakorlati vizsga (**100 pont / 50 pont**)
  - Egy megadott program elkészítése (2 óra)
  - Utolsó két hétben legalább egyszer el kell menni
  - Egy javítási lehetőség, minimum pontszámért

# Követelmények

- Gyakorlatok látogatása **KÖTELEZŐ**
- Kettőnél több igazolatlan hiányzás esetén a kurzus nem teljesítettnek minősül
- Igazolásokat a hiányzást követő első órán kell bemutatni az oktatónak (vagy fogadóórán) – később bemutatott igazolás *nem* kerül elfogadásra
- A gyakorlaton szerezhető pluszpontok nem számítanak bele a minimum pontszámba

# Követelmények

- Előadásra járni kötelező
- Félév során 3 db kisdolgozat, az előadások végén
- Egy darabot lehet javítani/pótolni, az utolsó előadáson

	Dátum	Elérhető pont	Minimum
Első	október 02.	10	5
Második	október 16.	15	10
Harmadik	október 30.	15	10

# Javítás

- **Előadás kisdolgozat**
  - 1 javítható/pótolható, **teljes pontszámért**
  - További kérdés, kérés, probléma → előadó
  - Tavalyi pontok áthozása (aki érintett):
    - <https://goo.gl/rBfhSh>
- **Kötelező programok**
  - NEM javítható/pótolható
- **Gyakorlati vizsga**
  - egyszer javítható, minimum pontért

# Értékelés

- Gyakorlaton szerezhető pontok: 100
- Kollokviumon szerezhető pontok: 140

$\geq 85\%$	jeles
$\geq 75\%$	jó
$\geq 64\%$	közepes
$\geq 50\%$	elégséges
$< 50\%$	☹

# Irodalom

- Herbert Schildt: C/C++ Referenciakönyv Panem Kft Budapest (1998)
- Scott Meyers: Hatékony C++, ISBN: 9639193828, Scolar Kiadó Budapest, (2003)
- Bjarne Stroustrup: A C++ Programozási nyelv I-II., ISBN:9789639301191, Kiskapu (2001)
- <https://buckysroom.org/>
- [pub/Programozas-II/](https://buckysroom.org/pub/Programozas-II/)
- [https://www.tutorialspoint.com/c\\_standard\\_library/index.htm](https://www.tutorialspoint.com/c_standard_library/index.htm)



# Ismétlés

ANSI C, OOP

# Adattípusok

- signed/unsigned
- short, int, long, long long
- char
- float, double, long double
- pointerek és tömbök ( void\* )
- enum
- union, struct
- Programozás alapjai és Programozás 1

# Main függvény

```
int main(int argc, char *argv[], char *envp[])
```

- A kezdő eljárás minden C/C++ programnál
- Paraméterei:
  - argc: argument count, a parancssori paraméterek száma
  - argv: argument vector, a parancssori argumentumok (string) címeit tartalmazó tömb
    - indexelés 0-tól, utolsó argumentum után NULL
  - envp: környezeti változók címeit tartalmazó tömb

# stdio.h – printf/scanf

- `int printf(const char * format, ... );`

- Példa:

- ```
printf("Az %d annyi, mint öt", 5);
```

- Visszatérési értéke?

- `int scanf(const char * format, ... );`

- Példa:

- ```
int x=-1;
```

- ```
printf("Mennyi az x értéke?");
```

- ```
scanf("%d", &x);
```

- Visszatérési értéke?

# Konverziós karakterek

Karakter	Jelentése
d vagy i	Előjeles egész szám (integer)
f	Lebegőpontos szám (float)
c	Karakter (char)
s	Szöveg (string)
p	Pointer
e vagy E	Tudományos (mantissza/exponens)
%	Két % segítségével lehet megjeleníteni egy darab százalékjelet
.szám	Szám megjelenítésének pontossága

```
printf("Egy %s megevett %03d csirkét", "kutya", 12);  
      Egy kutya megevett 012 csirkét.
```

# stdio.h – fopen/fclose

**FILE \*fopen(const char \*filename, const char \*mode);**

- Megnyitási módok: r – olvasás, w – írás, a – append, b – bináris, t – szöveges, stb..

**int fclose(FILE \*stream);**

- Például:

```
FILE *file;
```

```
file = fopen("cseresznye.txt", "r");
```

```
fclose(file);
```

# stdio.h – fprintf/fscanf

**int fprintf ( FILE \* stream, const char \* format, ... )**

- Példa:

```
int x = 4;
```

```
fprintf( file, "%d cseresznye a fájlba.\n",x);
```

**int fscanf ( FILE \* stream, const char \* format, ... )**

- Példa:

```
int x;
```

```
fscanf( file, "%d", &x);
```

# String

- Karaktertömb, melynek végét `\0` jelzi
- **`#include <string.h>`**
- Ezek után használható függvények:
- **`size_t strlen( const char *s)`**  
a paraméterként kapott string méretét adja vissza (`\0` karakter előttig számolva)
- **`char *strcpy( char *s1, const char *s2)`**  
s2 string tartalmát az s1 karaktertömbbe másolja
- **`char *strncpy( char *s1, const char *s2, size_t n)`**  
s2 stringből (legfeljebb) n számú karaktert másol s1-be



# String

- **char \*strcat( char \*s1, const char \*s2)**  
s2 string tartalmát hozzáfűzi s1-hez (s1 végi \0-t törli, és a hozzáfűzött s2 végére teszi)
- **char \*strncat( char \*s1, const char \*s2, size\_t n)**  
n karaktert fűz s1 végéhez s2-ből
- **char \*strchr( const char \*s, int c)**  
A stringben a megadott c karakter legelső előfordulására mutató pointert ad vissza. (NULL, ha nem fordul elő benne).
- **char \*strrchr( const char \*s, int c)**  
A megadott c karakter utolsó előfordulására mutató pointert ad vissza. (NULL, ha nem fordul elő)

# String

- **int strcmp( const char \*s1, const char \*s2)**  
Összehasonlítja s1 és s2 stringet. Visszatérési értéke 0, ha a két string egyforma. <0, ha s1 < s2.  
>0, ha s1 > s2.
- **int strncmp( const char \*s1, const char \*s2, size\_t n)**  
strcmp-hez hasonló, az első n karaktert vizsgálja
- **char \*strstr( const char \*s1, const char \*s2)**  
Ha az s2 string előfordul s1-ben, az s2 kezdetére mutató pointert ad vissza. NULL különben
- **char \*strrev( const char \*s1)**  
Megfordítja a string tartalmát.