



## INTRODUCERE ÎN PROGRAMAREA CALCULATOARELOR



**CURS**  
- CARE INCLUDE APLICAȚII -  
lacramioara.stoicu-tivadar@upt.ro

**Sala A111**  
(cabinet)

**APLICAȚII** **Sala B019**  
- CARE ÎNCLD CELE ÎNVĂȚATE LA CURS -  
Oana CHIRILA, Ștefan NICOLA, Cătălin PEȘCAN, Adela GĂBIU

**APLICAȚII**

## DE CE FACEM ACEST CURS?

- ▶ Toți, pentru a dobândi/aprofunda cunoștințe de bază legate de informatică și subiectele asociate
- ▶ Unii, pentru a dobândi/aprofunda cunoștințele de utilizare și programare a unui calculator - **programare structurată**, și
- ▶ Alții, pentru consolidarea, ordonarea și abordarea academică a unor cunoștințe pe care le au deja.

## DE CE FACEM ACEST CURS? ÎNAINTE ȘI DUPĂ



## CE FACEM LA ACEASTĂ DISCIPLINĂ?

- ▶ Noțiuni fundamentale asociate noțiunii de INFORMATICĂ
- ▶ Structura și funcționarea unui calculator
- ▶ Programele conexe unui calculator
- ▶ Noțiuni de Ingineria programării,
- ▶ Utilizarea calculatoarelor în diverse domenii,
- ▶ Prezentarea unei perspective asupra limbajelor și tehnologiilor utilizate în programare

## CE FACEM LA ACEASTĂ DISCIPLINĂ?

- ▶ Introducere în programare folosind limbajul C
- ▶ Tipuri standard, constante, variabile, expresii
- ▶ Funcții standard de citire/scriere, instrucțiuni simple
- ▶ Instrucțiuni selective
- ▶ Instrucțiuni ciclice
- ▶ Funcții și tipuri definite de utilizator, parametri
- ▶ Tipuri de date structurate, tipul tablou, tipul șir de caractere, structuri
- ▶ Pointeri

### Do you find C to be a fun programming language?

**John Joyce** · Follow  
OS Build Engineer at Apple (company) (2007-present) · 2y

Yes!

C is great fun because you can make things go really fast. You also get to deal with code and APIs that were written a long time ago which is interesting.

It's not just low-level hardware stuff, it's also all of the POSIX stuff in Unix and Unix-like systems.

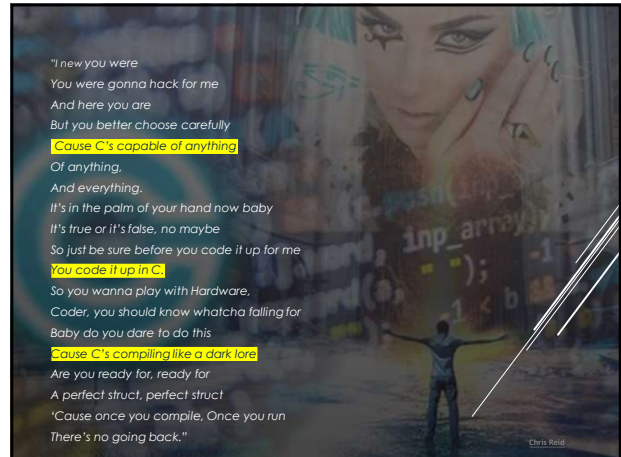
Every API is different and C is an incredibly flexible language that can do all sorts of things.

**David Rayna** · Follow  
Long-time programmer · 2y

Yes. Especially on small microcontrollers. You make it do what you want when you want it. And the tricks in short cuts you can take are a lot of fun like accessing memory and type casting pointers etc. For bigger things I prefer C++ for a few added things and the options for "objects" and derivation in the rare case I want them. Making new memory allocation strategies can be fun too.

**Manuel Capel** · Follow  
2y

I like it yes, somehow fun, you can go full free style messing around whatever you want directly into the machine without intermediates, you feel a bit like a wizard when you code in C, pretty fun yes :)



## EVALUAREA

- ▶ **Test grilă:** 30 întrebări din TOT cursul/ **20 min** – **testul este eliminător** – nu intrați la testul pe calculator dacă nu aveți peste nota 5 (15 puncte).
- ▶ **Examen pe calculator:** 90 minute
  - ▶ **Programarea 1:** în sesiune (în primele 3 săptămâni)
  - ▶ **Programarea 2** (inclusiv testul grilă 2): în sesiune (în săptămâna de după vacanța de după prima sesiune)
  - ▶ **Programarea 3, cu plată** (inclusiv testul grilă 3): pe parcursul semestrului 2 (la sfârșitul lunii martie).

La Programarea 2 și 3 timpul de lucru se micșorează.

Cerințele, modul de notare și desfășurarea examenului vor fi la dispoziție pe Campus Virtual unde aveți și materialele de curs și laborator.

## EVALUAREA

Nota la examenul de IPC,  $NE_{IPC}$ , se va calcula astfel:

$$NE_{IPC} = (NT + 2 \cdot NP) / 3$$

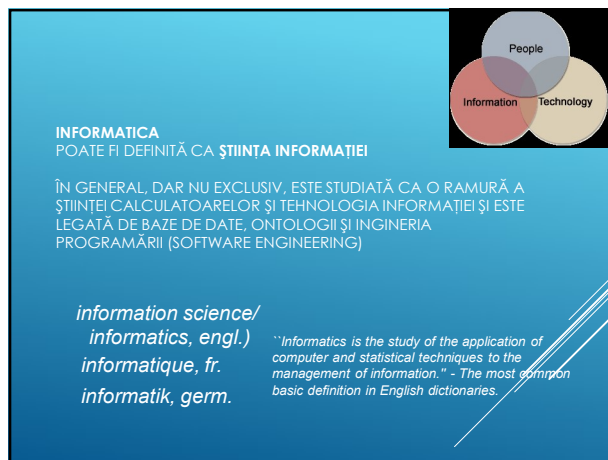
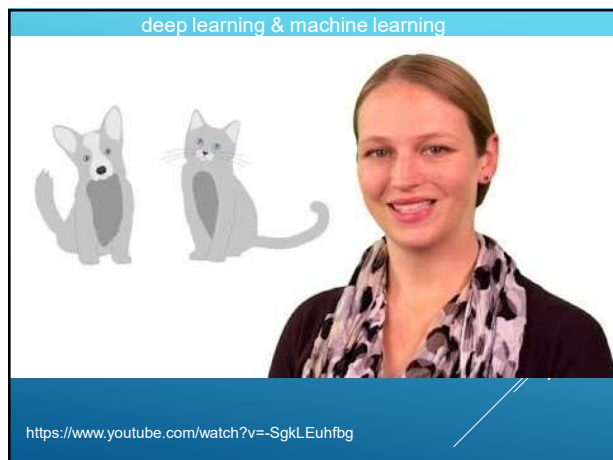
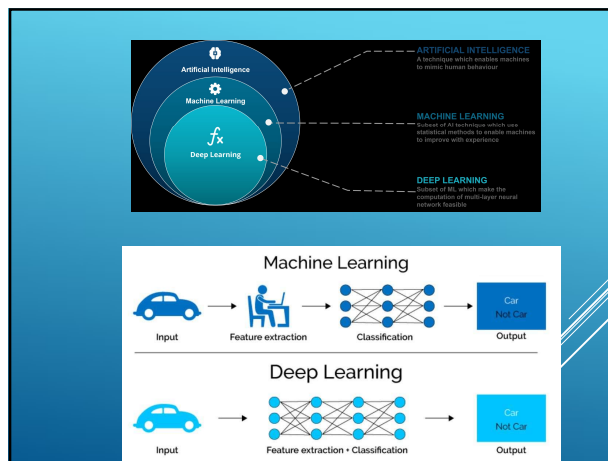
unde NT – Nota la testul grilă; NP – Nota la proba pe calculator.

**Nota finală pentru disciplina IPC va rezulta, conform regulamentului UPT**

$$1). \text{nota\_finală} = \text{parte\_intreaga\_din} \left[ \frac{\max(\text{col3a}), (\text{col3b}), (\text{col3c}) + k_2 \cdot (\text{col4}) + 0.5}{1} \right]$$

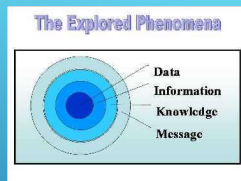
unde  $k_1 + k_2 = 1$ ,

Pentru IPC  $k_1 = 0.6$  și  $k_2 = 0.4$ .



## DATE

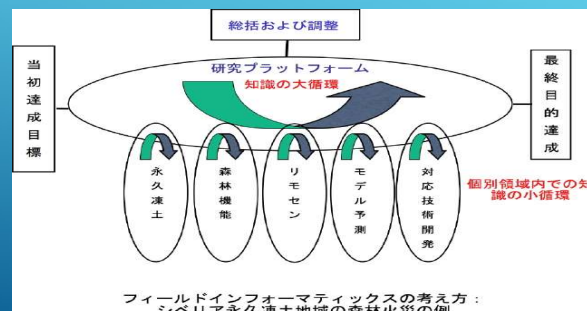
- ▶ Forma fundamentală a informației (un simbol, un semnal, etc.)
- ▶ Brut, nerafinat
- ▶ Numere/CNP
- ▶ Asocierea unui înțeles



Informație

Cunoaștere

??



フィールドインフォーマティックスの考え方：  
シベリア永久凍土地域の森林火災の例

## ȘTIINȚA INFORMAȚIEI INCLUDE

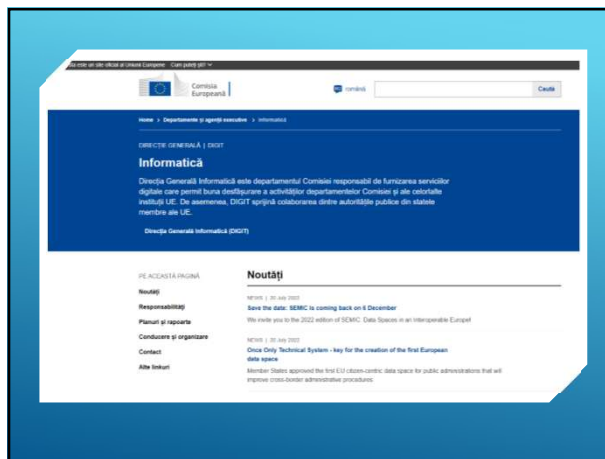
- ▶ Structurarea
- ▶ Crearea
- ▶ Organizarea
- ▶ Memorarea
- ▶ Obținerea
- ▶ Răspândirea
- ▶ Transferul informației

## INFORMATICA

- ▶ Se ocupă de studiul **reprezentării, prelucrării și comunicării informației**
- ▶ Conține aspecte legate de tehnologie, procesele cognitive și social.
- ▶ Transformarea informației – prin calcul sau comunicare

## EXEMPLE

- ▶ Bioinformatică (informația în sisteme biologice, genomică)
- ▶ Informatică medicală/în îngrijirea sănătății
- ▶ Chimie informatică (în chimie, industria medicamentelor)
- ▶ Ecoinformatică (în ecologie și științele mediului)
- ▶ Științe cognitive (mintea ca sistem de prelucrare a informațiilor)
- ▶ Informatică economică





- ▶ Dezvoltarea și integrarea sistemelor informatice din diverse domenii
- ▶ Dezvoltare și proiectare de baze de date
- ▶ Proiectarea interfețelor om-calculator
- ▶ Proiectarea rețelelor de calculatoare
- ▶ Proiectarea sistemelor informatice
- ▶ Implementare a soluțiilor în diverse limbaje
- ▶ Evaluare și testare
- ▶ Definirea specificațiilor și cerințelor pentru sisteme informatice
- ▶ Managementul resurselor
- ▶ Stăbilirea politicilor și a elementelor de etică legate de sistemele informatice



## ECHIPA IPC URMĂREȘTE SĂ VĂ TRANSMITĂ ȘI:

- ▶ Instrumente de dezvoltare a abilității de a **rezolva probleme**
- ▶ Posibilitatea de cunoaștere a rolului **abstractizării**
- ▶ Recunoașterea **soluțiilor alternative**
- ▶ Urmărirea subiectelor în **dinamica** reală
- ▶ Implicații **etice, juridice, sociale** (vandalism, drepturi de proprietate, profesionalism, securitate)
- ▶ **Abilități de comunicare** în limbaj profesionist

