Introducere in "Informatica" *de anul I*

SUMAR

- *Fundamentele programarii*
- *Structuri de date*
- *Algoritmi fundamentali*
- *Arhitectura sistemelor de calcul*
- *Programare orientata pe obiecte*

Fundamentele programarii

Sumarul disciplinei

Introducere in programarea procedurala

Operatori. Instructiuni.

Tablouri de date

Functii.

Pointeri.

Gestiunea dinamica a memoriei.

Structuri.

Lucrul cu fisiere.

Ce veti invata

Sa scrieti cod eficient

Sa intelegeti structurile de control a fluxului de executie a unui program

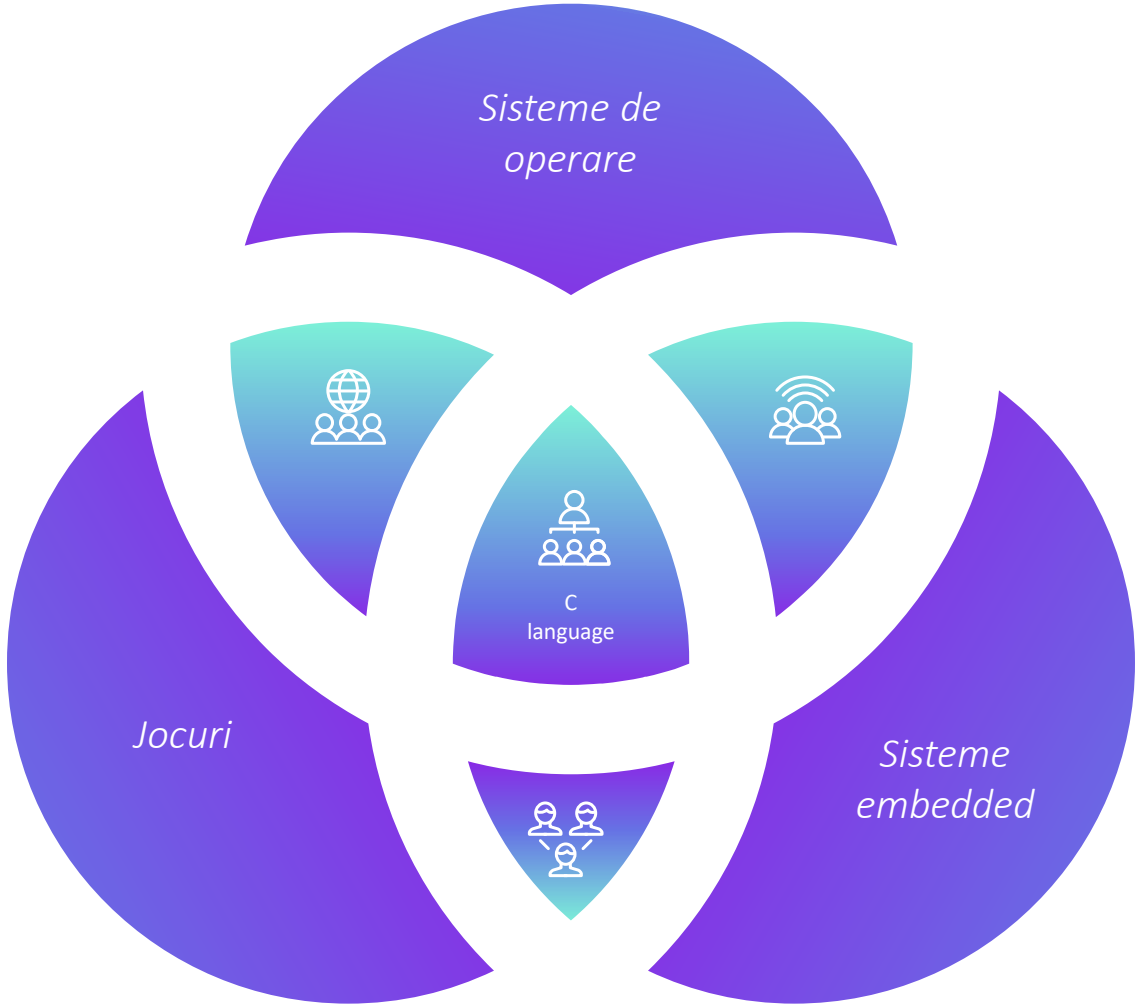
Sa lucrati cu functii.

Sa lucrati cu memoria programului in mod eficient.

ETehnici esentiale de rezolvare a problemelor

Algoritmi de baza

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

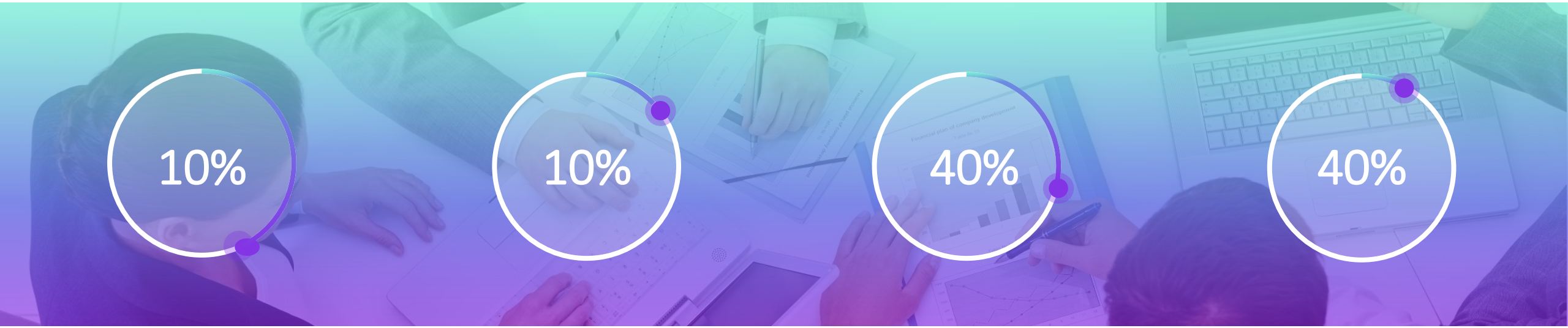


Limbajul C

- *Limbaj de programare fundamental pentru sisteme de operare, sisteme embedded, dezvoltarea jocurilor*
- *Eficient si rapid*
- *Ofera fundamente pentru invatarea altor limbaje de programare*

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Cunostinte anterioare necesare



Notiuni de aritmetica

Operatori aritmetici

Ordinea efectuarii operatiilor

Metode de rezolvare a problemelor

Logica matematica

Operatori logici

Conditii

Tenacitate

Perseverenta

Seriozitate

Indeplinirea obligatiilor impuse prin Fisa disciplinei

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Operatori aritmetici

Operatori aritmetici de baza

Adunare +, Scadere -,
Inmultire *, Impartire /
Modulo %

Exemplu x % y (restul impartirii lui x la y)

Utilizare in expresii

Impreuna cu date de tip numeric:
- Numere intregi (int)
- Numere rationale (float, double)

Operatorul modulo lucreaza **doar cu intregi.**
Exemplu: **5%2 = 1**

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Reguli de precedenta a operatorilor

Precedenta (prioritate)

Ordinea in care operatorii sunt evaluati intr-o expresie

Cea mai mare prioritate:

Inmultire *, Impartire /, Modulo %

Cea mai mica prioritate:

Adunare +, Scadere -

Asociativitate

Operatorii care au aceeasi prioritate, sunt evaluati pe baza *asociativitatii*

Majoritatea operatorilor aritmetici au asociativitate stanga-dreapta

a - b + c se evalueaza ca **(a - b) + c**

a + b * c se evalueaza ca **a + (b*c)**

Reguli de precedenta a operatorilor

Operatorul *paranteze ( )*

Are rolul de *grupare* a expresiilor, fortand o anumita ordine de evaluare

Suprascrie regulile de precedenta a operatorilor !!

Exemplu: **a+b*c** diferit de **(a+b)*c**

Paranteze imbricate pentru expresii complexe

Exemplu: **(a+b*c)/(a+b)**

Apel de functii

Parantezele se mai folosesc pentru apeluri de functii

Exemplu: sqrt(64)

Conditii in limbajul C

Conditii

Permit luarea de decizii pe baza unor criteri
Conditiile se evalueaza la true (1) sau false (0)

Operatori relationali
Folositi pentru a compara valori
== != < > <= >=

Exemplu

```
if (x > y) {
        printf("x is greater than y");
}
```

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Operatori logici in limbajul C

Operatorii logici

Folositi pentru a combina mai multe conditi

AND (&&)

True (adevarat) daca ambele conditii sunt True

Exemplu:

```
if(a>0 && b>0)
    printf("ambele pozitive");
else
    printf("macar unul nu e pozitiv");
```

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Operatori logici in limbajul C

Operatorii logici

Folositi pentru a combina mai multe conditi

OR (||)

True (adevarat) daca **macar una dintre conditii** este True

Exemplu:

```
if(a>0 || b>0)
    printf("cel putin unul pozitiv");
else
    printf("nici unul pozitiv");
```

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Operatori logici in limbajul C

Exemplu

Exprimam o conditie pentru o persoana apta de munca in Romania: varsta trebuie sa fie intre [18, 65] ani si sa fie apt de munca, dpdv medical

Exemplu:

```
    if ( (varsta >= 18 && varsta <= 65)   && (apt_de_munca)) {
            printf("Poate munci.");
}
```

Operator de negatie

Inverseaza valoarea de adevar a unei expresii

Exemplu: if (! a) // adevarat daca a este fals sau 0

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Legile lui DeMorgan

Prima lege

! (A && B) este echivalent cu !A || !B

Exemplu:

```
    if (!(age >= 18 && age <= 65)) {
    printf("Nu poate sa lucreze.");
} else {
    printf("Poate sa lucreze.");
}
//Echivalent cu
if (!(age >= 18) || !(age <= 65)) {
    printf("Nu poate sa lucreze.");
} else {
    printf("Poate sa lucreze.");
}
```

Legile lui DeMorgan

A doua lege

! (A || B) este echivalent cu !A && !B

Exemplu:

Sa zicem ca primirea unei recompense este conditionata de obtinerea unei note de 5, la unul dintre examenele FP sau SD.

```
    if (!(nota_FP>= 5 || nota_SD >= 5)) {
    printf("Nu primesti recompensa");
} else {
    printf("Primesti recompensa.");
}
//Echivalent cu
if (!(nota_FP >= 5) &&  !(nota_SD >=5)) {
    printf("Nu primesti recompensa.");
} else {
    printf("Primesti recompensa.");
}
```

Structuri de date

Sumarul disciplinei

- Date si informatii
- Structuri de date I
- Tablouri (vectori, matrice)
- Liste
- Grafuri
- Arbori
- Tabele de dispersie
- Date structurate si nestructurate

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Ce veti invata

Formarea unei gandiri algoritmice

Crearea de deprinderi de organizare riguroasa a activitatii de programare

AProfundarea limbajului de programare C

Algoritmi fundamentali

Sumarul disciplinei

Notiunea de algoritm. Pseudocod.

Notiuni de complexitate

Algoritmi elementari

Algoritmi de cautare

Algoritmi de sortare

Algoritmi recursivi

Metode de elaborare a algoritmilor: Divide et Impera, Greedy.

Algoritmi de compresie

Ce veti invata

Formarea unei gandiri algoritmice

Crearea de deprinderi de organizare riguroasa a activitatii de programare

AProfundarea limbajului de programare C

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Arhitectura sistemelor de calcul

Sumarul disciplinei

Arhitectura stratificata a sistemului de calcul

Descrierea componentelor hardware.

Nivelul microarhitecturii

Microprocesoare CISC

Limbajul de asamblare

Performanta sistemului de calcul

Arhitecturi RISC

Sisteme multiprocesor

Ce veti invata

Baze teoretice ale informaticii

Serviciile oferite de sistemele de calcul

Cunostinte despre componentele hardware

Programare in limbaj de asamblare

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Programare Orientata pe Obiecte

Sumarul disciplinei

Elemente de baza programare Java.

Obiecte. Constructori.

Mostenire.

Polimorfism.

Interfete.

Exceptii.

Interfete grafice.

Evenimente.

Colectii.

Ce veti invata

Paradigma programarii orientate spre obiecte

Mecanismele limbajului Java

Dezvoltarea abilitatilor proprii de programare

Testarea unitara

Dezvoltarea unui proiect software si a documentatiei aferente

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

La final ...

Veti avea o* altfel de *imagine despre "informatica"!

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro

Va multumesc!

Va doresc un an universitar cu sanatate si reusite!

Conf. dr. Elena Băutu, elena.bautu@365.univ-ovidius.ro