

FIȘA DISCIPLINEI¹

1. Date despre program

| | |
|--|--|
| 1.1 Instituția de învățământ superior | Universitatea Politehnică Timișoara |
| 1.2 Facultatea ² / Departamentul ³ | Facultatea de Inginerie Hunedoara / Inginerie Electrică și Informatică Industrială |
| 1.3 Catedra | — |
| 1.4 Domeniul de studii (denumire/cod ⁴) | INGINERIE ELECTRICĂ / 90 |
| 1.5 Ciclul de studii | Licenta |
| 1.6 Programul de studii (denumire/cod/calificarea) | INGINERIE ELECTRICĂ ȘI CALCULATOARE / 60 / Inginer |

2. Date despre disciplină

| | | | | | | | |
|--|---|---------------|---|-----------------------|---|--------------------------------------|----|
| 2.1 Denumirea disciplinei/Categoria formativă ⁵ | Programarea calculatoarelor și limbaje de programare 1 / DF | | | | | | |
| 2.2 Titularul activităților de curs | Ș. L. Dr. Iordan Anca-Elena | | | | | | |
| 2.3 Titularul activităților aplicative ⁶ | Asist. Dr. Ing. Ghiormez Loredana | | | | | | |
| 2.4 Anul de studii ⁷ | I | 2.5 Semestrul | I | 2.6 Tipul de evaluare | E | 2.7 Regimul disciplinei ⁸ | DI |

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)⁹

| | | | | | |
|--|------------------|--|----|---------------------------------------|----|
| 3.1 Număr de ore asistate integral/săptămână | 5 , format din: | 3.2 ore curs | 2 | 3.3 ore seminar/laborator/proiect | 3 |
| 3.1* Număr total de ore asistate integral/sem. | 70 , format din: | 3.2* ore curs | 28 | 3.3* ore seminar/laborator/proiect | 42 |
| 3.4 Număr de ore asistate parțial/săptămână | , format din: | 3.5 ore practică | | 3.6 ore elaborare proiect de diplomă | |
| 3.4* Număr total de ore asistate parțial/semestru | , format din: | 3.5* ore practică | | 3.6* ore elaborare proiect de diplomă | |
| 3.7 Număr de ore activități neasistate/săptămână | 5 , format din: | ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren | | | 1 |
| | | ore studiu individual după manual, suport de curs, bibliografie și notițe | | | 2 |
| | | ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri | | | 2 |
| 3.7* Număr total de ore activități neasistate/semestru | 70 , format din: | ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren | | | 14 |
| | | ore studiu individual după manual, suport de curs, bibliografie și notițe | | | 28 |
| | | ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri | | | 28 |
| 3.8 Total ore/săptămână ¹⁰ | 10 | | | | |
| 3.8* Total ore/semestru | 140 | | | | |
| 3.9 Număr de credite | 5 | | | | |

4. Precondiții (acolo unde este cazul)

| | |
|-------------------|------------------------------------|
| 4.1 de curriculum | • Cunoștințe de bază de matematică |
| 4.2 de competențe | • |

¹ Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 și cerințelor Standardelor specifice ARACIS valabile începând cu 01.10.2017.

² Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina.

³ Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului.

⁴ Se înscrie codul prevăzut în HG nr.140/16.03.2017 sau în HG similare actualizate anual.

⁵ Disciplina se încadrează potrivit planului de învățământ în una dintre următoarele categorii formative: disciplină fundamentală (DF), disciplină de domeniu (DD), disciplină de specialitate (DS) sau disciplina complementară (DC).

⁶ Prin activități aplicative se înțeleg activitățile de: seminar (S) / laborator (L) / proiect (P) / practică (Pr).

⁷ Anul de studii în care este prevăzută disciplina în planul de învățământ.

⁸ Disciplina poate avea unul din următoarele regimuri: disciplină impusă (DI), disciplină opțională (DO) sau disciplină facultativă (Df).

⁹ Numărul de ore de la rubricile 3.1*, 3.2*,...,3.8* se obțin prin înmulțirea cu 14 (săptămâni) a numărului de ore din rubricile 3.1, 3.2,...., 3.8. Informațiile din rubricile 3.1, 3.4 și 3.7 sunt chei de verificare folosite de ARACIS sub forma: (3.1)+(3.4) ≥ 28 ore/săpt. și (3.8) ≤ 40 ore/săpt.

¹⁰ Numărul total de ore / săptămână se obține prin însumarea numărului de ore de la punctele 3.1, 3.4 și 3.7.

5. Condiții (acolo unde este cazul)

| | |
|---|--|
| 5.1 de desfășurare a cursului | <ul style="list-style-type: none"> • Sală de curs echipată cu videoproiector și conexiune la Internet. • Studenții nu se vor prezenta la prelegeri cu telefoanele mobile deschise. |
| 5.2 de desfășurare a activităților practice | <ul style="list-style-type: none"> • Sală de laborator echipată cu cu videoproiector și computere. • Studenții nu se vor prezenta la activitățile practice cu telefoanele mobile deschise. |

6. Competențe la formarea cărora contribuie disciplina

| | |
|---|--|
| Competențe specifice | <p>C 2.</p> <p>C 2.1. Descrierea funcționării și structurii sistemelor de calcul și a aplicațiilor lor în ingineria electrică folosind cunoștințele referitoare la limbajele, mediile și tehnologiile de programare și la instrumente specifice (algoritmi, scheme, modele, protocoale etc.);</p> <p>C 2.2. Explicarea și interpretarea pachetelor de programe pentru proiectarea și optimizarea sistemelor electrice reprezentative;</p> <p>C 2.3. Rezolvarea de probleme uzuale din domeniul ingineriei electrice folosind pachete de programe dedicate și mijloace de proiectare asistată de calculator (CAD) adecvate;</p> <p>C 2.4. Evaluarea rezultatelor obținute în urma utilizării pachetelor de programe și a mijloacelor de proiectare asistată de calculator (CAD) în rezolvarea problemelor din domeniul ingineriei electrice;</p> <p>C 2.5. Transpunerea problemelor din ingineria electrică în programe de calculator.</p> <ul style="list-style-type: none"> • |
| Competențele profesionale în care se înscriu competențele specifice | <ul style="list-style-type: none"> • C 2. Operarea cu concepte fundamentale din știința calculatoarelor și tehnologia informației. |
| Competențele transversale în care se înscriu competențele specifice | <ul style="list-style-type: none"> • |

7. Obiectivele disciplinei (asociate competențelor de la punctul 6)

| | |
|---------------------------------------|--|
| 7.1 Obiectivul general al disciplinei | <ul style="list-style-type: none"> • Familiarizarea studenților cu principiile de bază ale algoritmilor, a proprietăților acestora și a modului de reprezentare al acestora. Însușirea conceptelor de bază ale limbajului de programare Python. |
| 7.2 Obiectivele specifice | <ul style="list-style-type: none"> • |

8. Conținuturi¹¹

| 8.1 Curs | Număr de ore | Metode de predare ¹² |
|--|--------------|---|
| <p>1. Teoria algoritmilor</p> <p>1.1. Definiție. Proprietăți</p> <p>1.2. Expresii</p> <p>1.3. Operațiile de bază realizate de un algoritm</p> <p>1.4. Reprezentarea algoritmilor</p> <p>1.4.1. Scheme logice</p> <p>1.4.2. Principiile de bază ale programării structurate</p> <p>1.4.3. Limbajul pseudocod</p> <p>1.5. Subalgoritmi</p> <p>1.6. Recursivitate</p> <p>1.7. Analiza algoritmilor</p> <p>1.7.1. Verificarea corectitudinii</p> | 12 | <p>Studenții au acces la curs în format electronic.</p> <p>Se vor utiliza atât prezentări interactive cât și tradiționale.</p> <p>Se vor folosi: problematizarea, studiu de caz, conversația.</p> |

¹¹ Se detaliază toate activitățile didactice prevăzute prin planul de învățământ (tematicile prelegerilor și ale seminariilor, lista lucrărilor de laborator, conținuturile etapelor de elaborare a proiectelor, tematica fiecărui stagi de practică). Titlurile lucrărilor de laborator care se efectuează pe standuri vor fi însoțite de notația „(*)”.

¹² Prezentarea metodelor de predare va include și folosirea noilor tehnologii (e-mail, pagină personalizată de web, resurse în format electronic etc.).

| | | |
|---|---------------------|--|
| 1.7.2. Calculul efectuat prin execuția unui algoritm 1.7.3. Analiza eficienței (complexității). Analiză asimptotică. | | |
| 2. Limbajul de programare Python 2.1. Evoluția limbajelor de programare 2.2. Cuvinte cheie 2.3. Tipuri de date și variabile 2.4. Operatori 2.5. Funcții de intrare/ieșire 2.6. Structura alternativă – instrucțiunea if 2.7. Instrucțiunile corespunzătoare structurii repetitive 2.8. Module Python și includerea acestora într-un program 2.9. Funcții. Tipuri de variabile 2.10. Colecții de date: tipuri secvențiale, mulțimi și dicționare 2.11. Prelucrarea fișierelor 2.11.1. Fișiere text: csv, xml; 2.11.2. Fișiere binare. | 16 | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| Bibliografie¹³ 1. Manuela Pănoiu, Ionel Muscalagiu, Caius Pănoiu, Utilizarea și programarea calculatoarelor, Editura Mirton, 2002, Timișoara 2. Cristian Giumale, Introducere în analiza algoritmilor, Editura Polirom, 2004, Iași 3. Rod Stephens, Essential Algorithms, John Wiley & Sons Ltd, 2013 4. Magnus Lie Hetland, Beginning Python From Novice to Professional, APress, 2005 5. Mark Lutz, Programming Python, O'Reilly Media, 2011 | | |
| 8.2 Activități aplicative¹⁴ | Număr de ore | Metode de predare |
| 1. Reprezentarea de algoritmi prin intermediul schemelor logice care utilizează structura secvențială, structura alternativă și structura repetitivă | 6 | Se va utiliza exercițiul la tablă. |
| 2. Reprezentarea de algoritmi prin intermediul limbajului Pseudocod utilizând cele trei structuri de control | 3 | |
| 3. Determinarea calculului efectuat la execuția unui algoritm | 3 | |
| 4. Rezolvarea problemelor utilizând subalgoritmi implementați iterativ și/sau recursiv | 3 | |
| 5. Implementarea în Python a programelor corespunzătoare algoritmilor care utilizează instrucțiunile specifice celor trei structuri de control | 6 | Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul. |
| 6. Utilizarea funcțiilor la implementarea programelor | 3 | Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul. |
| 7. Implementarea în Python a programelor corespunzătoare algoritmilor care utilizează tablouri unidimensionale și bidimensionale | 6 | Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul. |
| 8. Utilizarea tipurilor de date compuse la implementarea programelor | 6 | Se va utiliza exercițiul la tablă și implementarea |

¹³ Cel puțin un titlu trebuie să aparțină colectivului disciplinei iar cel puțin un titlu trebuie să se refere la o lucrare de referință pentru disciplină, de circulație națională și internațională, existentă în biblioteca UPT.

¹⁴ Tipurile de activități aplicative sunt cele precizate în nota de subsol 5. Dacă disciplina conține mai multe tipuri de activități aplicative atunci ele se trec consecutiv în liniile tabelului de mai jos. Tipul activității se va înscrice într-o linie distinctă sub forma: „Seminar:”, „Laborator:”, „Proiect:” și/sau „Practică:”.

| | | |
|---|---|--|
| | | rea programului utilizând calculatorul. |
| 9. Implementarea în Python a programelor ce utilizează fișiere text și binare | 6 | Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul. |

Bibliografie¹⁵

1. Anca-Elena Iordan, Daniela Cristea, Adela Berdie, Programarea calculatoarelor, Editura Mirton, Timișoara, 2004
2. Robert Sedgewick, Kevin Wayne, Algorithms, Pearson Education, 2011
3. Magnus Lie Hetland, Python Algorithms Mastering Basic Algorithms in the Python Language, APress, 2010
4. David Beazley, Python Essential Reference, Sams Publishing, 2006
5. <https://pythonprogramming.net/python-fundamental-tutorials>

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Disciplina vine în întâmpinarea așteptărilor angajatorilor reprezentativi din domeniul aferent programului prin conținutul orelor de curs și laborator.

10. Evaluare

| Tip activitate | 10.1 Criterii de evaluare ¹⁶ | 10.2 Metode de evaluare | 10.3 Pondere din nota finală |
|--|--|---|------------------------------|
| 10.4 Curs | Cunoștințe teoretice | Scris - subiecte teoretice și aplicații | 0,66 |
| 10.5 Activități aplicative | S: | | |
| | L: Abilități în analiza, proiectarea și implementarea aplicațiilor de laborator | Oral – aplicații utilizând calculatorul | 0,34 |
| | P¹⁷: | | |
| | Pr: | | |
| 10.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁸) | | | |
| <ul style="list-style-type: none"> • La finalul cursului, respectiv al laboratorului, studentul trebuie să aibă cunoștințe solide despre analiza și reprezentarea unui algoritm, precum și de implementare a acestuia în limbajul de programare Python. | | | |

Data completării

04.09.2017

**Titular de curs
(semnătura)**

.....

**Titular activități aplicative
(semnătura)**

.....

**Director de departament
(semnătura)**

.....

Data avizării în Consiliul Facultății¹⁹

06.09.2017

**Decan
(semnătura)**

.....

¹⁵ Cel puțin un titlu trebuie să aparțină colectivului disciplinei.

¹⁶ Fișele disciplinelor trebuie să conțină procedura de evaluare a disciplinei cu precizarea criteriilor, a metodelor și a formelor de evaluare, precum și cu precizarea ponderilor atribuite acestora în nota finală. Criteriile de evaluare se formulează în mod distinct pentru fiecare activitate prevăzută în planul de învățământ (curs, seminar, laborator, proiect). Ele se vor referi și la formele de verificare pe parcurs (teme de casă, referate ș.a.)

¹⁷ În cazul când proiectul nu este o disciplină distinctă, în această rubrică se va preciza și modul în care rezultatul evaluării proiectului condiționează admiterea studentului la evaluarea finală din cadrul disciplinei.

¹⁸ Nu se va explica cum se acorda nota de promovare.

¹⁹ Avizarea este precedată de discutarea punctului de vedere al board-ului de care aparține programul de studii cu privire la fișa disciplinei.