

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	<b>Universitatea Babeș-Bolyai din Cluj-Napoca</b>
1.2 Facultatea	<b>Facultatea de Matematică și Informatică</b>
1.3 Departamentul	<b>Departamentul de Informatică</b>
1.4 Domeniul de studii	<b>Informatică</b>
1.5 Ciclul de studii	<b>Licență</b>
1.6 Programul de studiu / Calificarea	<b>Informatică română</b>

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Fundamentele programării</b>						
2.2 Titularul activităților de curs	<b>Conf. dr. Istvan Gergely Czibula</b>						
2.3 Titularul activităților de seminar	<b>Conf. dr. Istvan Gergely Czibula</b>						
2.4 Anul de studiu	<b>1</b>	2.5 Semestrul	<b>1</b>	2.6. Tipul de evaluare	<b>E</b>	2.7 Regimul disciplinei	<b>Obligatorie</b>

### 3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	6	Din care: 3.2 curs	2	3.3 seminar/laborator	2 sem 2 lab
3.4 Total ore din planul de învățământ	84	Din care: 3.5 curs	28	3.6 seminar/laborator	56
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					12
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					14
Tutoriat					8
Examinări					18
Alte activități: .....					-
3.7 Total ore studiu individual		66			
3.8 Total ore pe semestru		150			
3.9 Numărul de credite		6			

### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	-
4.2 de competențe	-

### 5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none"> <li>Sală, plus proiector</li> </ul>
-------------------------------	--

5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> <li>• Laboratoare echipate cu Python</li> </ul>
--	--

## 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	<p>C1.1 Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic.</p> <p>C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază</p> <p>C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date</p> <p>C1.4 Testarea unor aplicații pe baza unor planuri de test</p> <p>C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente</p>
<b>Competențe transversale</b>	<p>CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională</p> <p>CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională</p>

## 7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>• Sa introduca conceptele de baza ale ingineriei software (proiectare, implementare si intretinere) si sa prezinte limbajul de programare Python.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• Sa introduca conceptele de baza ale programarii</li> <li>• Sa introduca conceptele de baza ale ingineriei software</li> <li>• Sa foloseasca instrumente de baza pentru construirea programelor</li> <li>• Sa prezinte limbajul Python si instrumente de dezvoltare pentru programarea, executia si depanarea programelor Python.</li> <li>• Sa promoveze un stil de programare conform celor mai bune recomandari practice.</li> </ul>

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
<p><b>1. Introducere în procese de dezvoltare software</b></p> <ul style="list-style-type: none"> <li>• Ce este programarea: algoritm, program, elemente de baza Python, interpretor Python, roluri în ingineria software</li> <li>• Cum scriem programe: enunț problema, cerințe, proces de dezvoltare dirijat de funcționalități (FDD)</li> <li>• Exemple: calculator</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
<p><b>2. Programare procedurală</b></p> <ul style="list-style-type: none"> <li>• Tipuri structurate: liste, tuple, dicționare</li> <li>• Funcții: cazuri de testare, definiție, variabile, apel</li> <li>• Transmiterea parametrilor</li> <li>• Funcții anonime</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactivă</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	

<ul style="list-style-type: none"> <li>Cum scriem functii: programare dirijata de teste, refactorizari</li> </ul>		
<b>3. Programare modulara</b> <ul style="list-style-type: none"> <li>Module, pachete: modul Python, domeniul variabilelor, pachete, module standard, distribuire module</li> <li>Organizarea aplicației pe module și pachete</li> <li>Medii de dezvoltare – IDE - Eclipse+PyDev</li> </ul>	<ul style="list-style-type: none"> <li>Expunere interactiva</li> <li>Explicatie</li> <li>Conversatie</li> <li>Exemple</li> <li>Demonstratie didactica</li> </ul>	
<b>4. Principii de organizarea aplicațiilor</b> <ul style="list-style-type: none"> <li>Principiul unei singure responsabilități (SRP)</li> <li>Principiu separării responsabilităților (SoC)</li> <li>Cuplare, Coeziune</li> <li>Arhitectură stratificată</li> <li>Tratarea erorilor - Excepții</li> </ul>	<ul style="list-style-type: none"> <li>Expunere interactiva</li> <li>Explicatie</li> <li>Conversatie</li> <li>Exemple</li> <li>Demonstratie didactica</li> </ul>	
<b>5. Tipuri definite de utilizator</b> <ul style="list-style-type: none"> <li>Programare orientată obiect - Clase, obiecte</li> <li>Principii de definire a tipurilor utilizator: Încapsulare, ascunderea detaliilor de implementare</li> <li>Tipuri abstracte de date: Interfața, Implementare, Specificații</li> </ul>	<ul style="list-style-type: none"> <li>Expunere interactiva</li> <li>Explicatie</li> <li>Conversatie</li> <li>Exemple</li> <li>Demonstratie didactica</li> </ul>	
<b>6. Principii de proiectare</b> <ul style="list-style-type: none"> <li>Problema: program cu operații CRUD pe entități de un tip dat</li> <li>Șabloane GRASP: Information Expert, Low Coupling, High Cohesion, Protected Variation, Single responsibility, Dependency Injection</li> <li>Arhitectura stratificata: UI, Domeniu, GRASP Controllere, Infrastructura</li> </ul>	<ul style="list-style-type: none"> <li>Expunere interactiva</li> <li>Explicatie</li> <li>Conversatie</li> <li>Exemple</li> <li>Demonstratie didactica</li> </ul>	
<b>7. Proiectarea programelor</b> <ul style="list-style-type: none"> <li>Șabloane DDD: entity, validator, repository, controller</li> <li>Diagrame UML</li> <li>Top down vs bottom up</li> <li>Dynamic Typing, Duck typing</li> <li>Reutilizare de cod prin moștenire</li> <li>Fișiere text in Python</li> </ul>	<ul style="list-style-type: none"> <li>Expunere interactiva</li> <li>Explicatie</li> <li>Conversatie</li> <li>Exemple</li> <li>Demonstratie didactica</li> </ul>	
<b>8. Testarea, inspectarea si depanarea programelor</b> <ul style="list-style-type: none"> <li>Black box vs white box testing</li> <li>Unit testing, integration testing</li> <li>Testare automata, PyUnit</li> <li>Depanare</li> <li>Stil de programare, factorizări</li> </ul>	<ul style="list-style-type: none"> <li>Expunere interactiva</li> <li>Explicatie</li> <li>Conversatie</li> <li>Exemple</li> <li>Demonstratie didactica</li> </ul>	
<b>9. Recursivitate</b> <ul style="list-style-type: none"> <li>Recursivitate directa si indirecta</li> <li>Exemple</li> </ul> <b>Complexitatea algoritmilor</b> <ul style="list-style-type: none"> <li>Notația asimptotica: big-o, little-o, big-omega, little-omega, theta</li> <li>Complexități uzuale, formule matematice (constant, logaritmic, liniar, quadratic, polinomial, exponențial)</li> <li>Comparații algoritmi, calcul complexități</li> </ul>	<ul style="list-style-type: none"> <li>Expunere interactiva</li> <li>Explicatie</li> <li>Conversatie</li> <li>Exemple</li> <li>Demonstratie didactica</li> </ul>	
<b>10. Algoritmi de căutare</b>	<ul style="list-style-type: none"> <li>Expunere interactiva</li> </ul>	

<ul style="list-style-type: none"> <li>• căutare secvențială, căutare binară</li> </ul> <b>Algoritmi de sortare</b> <ul style="list-style-type: none"> <li>• BubbleSort, SelectionSort, InsertionSort</li> <li>• QuickSort, MergeSort</li> <li>• Complexitatea algoritmilor</li> </ul>	<ul style="list-style-type: none"> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
<b>11. Backtracking</b> <ul style="list-style-type: none"> <li>• Algoritmul Backtracking</li> <li>• Extensii ale algoritmului</li> <li>• Exemple</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
<b>12. Divide et Impera</b> <ul style="list-style-type: none"> <li>• Descriere Metoda</li> <li>• Exemple</li> </ul> <b>Greedy</b> <ul style="list-style-type: none"> <li>• Descriere Metoda</li> <li>• Exemple</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
<b>13 Programare dinamica</b> <ul style="list-style-type: none"> <li>• Descriere Metoda</li> <li>• Exemple</li> </ul>	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
<b>14. Recapitulare</b>	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Conversatie</li> </ul>	
<b>Bibliografie</b> <ol style="list-style-type: none"> <li>1. Kent Beck. <i>Test Driven Development: By Example</i>. Addison-Wesley Longman, 2002. See also Test-driven development. <a href="http://en.wikipedia.org/wiki/Test-driven_development">http://en.wikipedia.org/wiki/Test-driven_development</a></li> <li>2. Martin Fowler. <i>Refactoring. Improving the Design of Existing Code</i>. Addison-Wesley, 1999. See also <a href="http://refactoring.com/catalog/index.html">http://refactoring.com/catalog/index.html</a></li> <li>3. Frentiu, M., H.F. Pop, Serban G., Programming Fundamentals, Cluj University Press, 2006</li> <li>4. <i>The Python language reference</i>. <a href="http://docs.python.org/py3k/reference/index.html">http://docs.python.org/py3k/reference/index.html</a></li> <li>5. <i>The Python standard library</i>. <a href="http://docs.python.org/py3k/library/index.html">http://docs.python.org/py3k/library/index.html</a></li> <li>6. <i>The Python tutorial</i>. <a href="http://docs.python.org/tutorial/index.html">http://docs.python.org/tutorial/index.html</a></li> </ol>		
8.2 Seminar / laborator	Metode de predare	Observații
1. Programe Python	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
2. Programare procedurala	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
3. Programare modulara	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie</li> </ul>	

	didactica	
4. Tipuri definite de utilizator	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
5. Principii de proiectare	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
6. POO	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
7. Proiectare	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
8. Testare si inspectare	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
9. Recursivitate	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
10. Complexitatea algoritmilor	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
11. Backtracking	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> </ul>	

	<ul style="list-style-type: none"> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
12. Metoda injumatatirii. Algoritmi de cautare	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
13. Pregatirea examenului practic	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
14: Pregatirea examenului scris	<ul style="list-style-type: none"> <li>• Expunere interactiva</li> <li>• Explicatie</li> <li>• Conversatie</li> <li>• Exemple</li> <li>• Demonstratie didactica</li> </ul>	
<b>Bibliography</b>		
<ol style="list-style-type: none"> <li>1. Kent Beck. <i>Test Driven Development: By Example</i>. Addison-Wesley Longman, 2002. See also Test-driven development. <a href="http://en.wikipedia.org/wiki/Test-driven_development">http://en.wikipedia.org/wiki/Test-driven_development</a></li> <li>2. Martin Fowler. <i>Refactoring. Improving the Design of Existing Code</i>. Addison-Wesley, 1999. See also <a href="http://refactoring.com/catalog/index.html">http://refactoring.com/catalog/index.html</a></li> <li>3. Frentiu, M., H.F. Pop, Serban G., <i>Programming Fundamentals</i>, Cluj University Press, 2006</li> <li>4. <i>The Python language reference</i>. <a href="http://docs.python.org/py3k/reference/index.html">http://docs.python.org/py3k/reference/index.html</a></li> <li>5. <i>The Python standard library</i>. <a href="http://docs.python.org/py3k/library/index.html">http://docs.python.org/py3k/library/index.html</a></li> <li>6. <i>The Python tutorial</i>. <a href="http://docs.python.org/tutorial/index.html">http://docs.python.org/tutorial/index.html</a></li> </ol>		

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

<ul style="list-style-type: none"> <li>• Cursul respectă curricula IEEE și ACM pentru domeniul Informatică.</li> <li>• Cursul există în programele de studiu ale universităților importante din România și din străinătate.</li> <li>• Conținutul disciplinei este considerat de majoritatea companiilor software ca fiind deosebit de important pentru obținerea unor abilități medii de programare.</li> </ul>
--

### 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoștințele acumulate	Examen scris	40%
10.5 Seminar/laborator	Scrierea unui program	Examen practic	30%
	Programele scrise in timpul semestrului	Documentație	30%

#### 10.6 Standard minim de performanță

- Fiecare student trebuie să demonstreze că a atins un nivel acceptabil de cunoștințe și înțelegere a domeniului, că este capabil să prezinte aceste cunoștințe într-o manieră coerentă și că are abilitatea de a stabili anumite conexiuni și de a folosi aceste cunoștințe în rezolvarea diferitelor probleme în limbajul de programare Python.
- Pentru promovare, este obligatorie prezența la minim 10 seminarii și 12 laboratoare .
- Promovarea este condiționată de nota minimă 5 la fiecare – nota laborator, proba practică și examenul scris.

Data completării

20.04.2017

Semnătura titularului de curs

Conf. dr. Istvan Gergely Czibula

Semnătura titularului de seminar

Conf. dr. Istvan Gergely Czibula

Data avizării în departament

Semnătura directorului de departament

Prof. dr. Anca Andreica