ECDL

SYLLABUS

MODUL ECDL
COMPUTING

Versiune Syllabus 1.0

Obiective

Aceasta reprezintă programa pentru modulul ECDL Computing. Ea descrie, prin rezultatele învăţării, cunoştinţele şi competenţele pe care un candidat ar trebui să le aibă. Programa prezintă, de asemenea, baza pentru testul teoretic şi proba practică a acestui modul.

Disclaimer

Chiar dacă în pregătirea acestei publicații au fost luate toate măsurile de precauție de către Fundația ECDL, aceasta nu poate oferi nicio garanție ca editor cu privire la complexitatea informațiilor conținute în ea. Fundația ECDL nu este responsabilă de eventualele erori, omisiuni, inexactități, pierderi sau distrugeri de informații și instrucțiuni conținute în această publicație. Fundația ECDL poate modifica această programă oricând, fără un aviz prealabil.

Versiunea oficială a Programei Analitice ECDL pentru **Modulul Computing** este publicată în secțiunea **Download** a website-ului **www.ecdl.ro**

Computing

Acest modul stabilește conceptele esențiale și competențele referitoare la abilitatea de a utiliza gândirea computațională și cunoștințele de programare pentru a crea aplicații simple de computer.

Obiectivele modulului

Candidații vor fi capabili să:

- Înțeleagă conceptele cheie cu privire la computing și la etapele de creare a unui program.
- Înțeleagă și să utilizeze tehnici de gândire computațională precum descompunerea, recunoașterea de șabloane, abstractizarea și utilizarea algoritmilor în analiza unei probleme și dezvoltarea unei soluții.
- Scrie, testeze și modifice algoritmi pentru un program, utilizând scheme logice și pseudocod.
- Înțeleagă conceptele cheie asociate programării, precum și importanța unui cod documentat și bine structurat.
- Înțeleagă și să utilizeze variabile, constante și expresii în programare.
- Îmbunătățească eficiența și funcționalitatea unui program prin utilizarea iterațiilor, structurilor condiționale, procedurilor și funcțiilor, evenimentelor și comenzilor.
- Testeze și depaneze un program și să se asigure că acesta respectă cerințele tehnice înainte de lansare.

CATEGORIE	SET APTITUDINI	REF.	TEMATICĂ
1 Noțiuni de bază	*1.1 Concepte cheie*	1.1.1	Definirea termenului de computing.
		1.1.2	Definirea termenului de gândire computațională.
		1.1.3	Definirea termenului de program.
		1.1.4	Definirea termenului de cod. Distingerea termenilor de cod sursă și cod mașină.
		1.1.5	Înțelegerea termenilor de descriere a unui program și specificații.
		1.1.6	Recunoașterea etapelor de creare a unui program: analiză, proiectare, programare, testare, îmbunătățire.
		1.1.7	Înțelegerea diferențelor dintre un limbaj formal și un limbaj natural.
2 Metode de gândire computațională	2.1 Analiza problemei	2.1.1	Identificarea metodelor utilizate în gândirea computațională: descompunerea, recunoașterea de șabloane, abstractizarea și utilizarea algoritmilor.
		2.1.2	Descompunerea unor date, procese sau a unei probleme complexe în elemente mai mici.
		2.1.3	Identificarea șabloanelor în cadrul problemelor mici, descompuse.
		2.1.4	Utilizarea abstractizării pentru a filtra detaliile inutile în analiza unei probleme.

CATEGORY	SKILL SET	REF.	TASK ITEM
		2.1.5	Înțelegerea modului în care algoritmii sunt utilizați în gândirea computațională.
	2.2 Algoritmi	2.2.1	Definirea termenului de secvență de instrucțiuni. Identificarea scopului utilizării secvențelor de instrucțiuni în crearea algoritmilor.
		2.2.2	Recunoașterea metodelor posibile de reprezentare a problemelor: scheme logice, pseudocod.
		2.2.3	Recunoașterea simbolurilor din schemele logice, precum: start/stop, proces, decizie, intrare/ieșire, conector, săgeată.
		2.2.4	Identificarea secvenței de operații reprezentate în cadrul unei scheme logice sau unui pseudocod.
		2.2.5	Scrierea unui algoritm corect pe baza unei descrieri, utilizând o schemă logică sau un pseudocod.
		2.2.6	Remedierea erorilor dintr-un algoritm precum: elemente de program lipsă, secvență incorectă de instrucțiuni, rezultat incorect.
3 Programare	*3.1 Noțiuni introductive*	3.1.1	Descrierea caracteristicilor unui cod bine structurat și documentat: indentare, comentarii adecvate, utilizarea numelor descriptive.
		3.1.2	Utilizarea operatorilor aritmetici simpli pentru efectuarea calculelor într-un program: +, -, /, *.
		3.1.3	Înțelegerea ordinii operatorilor și a ordinii de evaluare a expresiilor complexe. Înțelegerea modului de utilizare a parantezelor pentru structurarea expresiilor complexe.
		3.1.4	Înțelegerea termenului de parametru. Identificarea scopului unui parametru într-un program.
		3.1.5	Definirea termenului de comentariu. Identificarea scopului unui comentariu într-un program.
		3.1.6	Utilizarea comentariilor într-un program.
	3.2 Variabile și tipuri de date	3.2.1	Definirea termenului de variabilă. Identificarea scopului unei variabile într-un program.
		3.2.2	Definirea și inițializarea unei variabile.
		3.2.3	Atribuirea unei valori unei variabile.
		3.2.4	Utilizarea variabilelor denumite corespunzător în cadrul unui program pentru efectuarea calculelor și stocarea valorilor.

CATEGORY	SKILL SET	REF.	TASK ITEM
		3.2.5	Utilizarea tipurilor de date într-un program: șir de caractere (string), caracter (character), număr întreg (integer), număr real (float), boolean.
		3.2.6	Utilizarea unui tip de date agregat precum: tablou, listă, tuplu.
		3.2.7	Utilizarea datelor de intrare într-un program.
		3.2.8	Utilizarea datelor de ieșire într-un program.
4 Construirea codurilor	*4.1 Expresii logice*	4.1.1	Definirea termenului de test logic. Identificarea scopului unui test logic într-un program.
		4.1.2	Recunoașterea tipurilor de expresii logice booleene pentru a genera o valoare adevărată sau falsă: =, >, <, >=, <=, <>, !=, ==, AND, OR, NOT.
		4.1.3	Utilizarea expresiilor logice booleene în cadrul unui program.
	4.2 Iterații	4.2.1	Definirea termenului de buclă (loop). Identificarea scopului și beneficiilor utilizării unei bucle într-un program.
		4.2.2	Recunoașterea tipurilor de bucle utilizate în cadrul iterațiilor: for, while, repeat.
		4.2.3	Utilizarea iterațiilor într-un program: for, while, repeat.
		4.2.4	Înțelegerea termenului de buclă infinită.
		4.2.5	Înțelegerea termenului de recursivitate.
	4.3 Condiționalități	4.3.1	Definirea termenului de instrucțiune condițională. Identificarea scopului unei instrucțiuni condiționale în cadrul unui program.
		4.3.2	Utilizarea instrucțiunii condiționale IF…THEN...ELSE într-un program.
	4.4 Proceduri și funcții	4.4.1	Înțelegerea termenului de procedură. Identificarea scopului unei proceduri într-un program.
		4.4.2	Scrierea și definirea unei proceduri într-un program.
		4.4.3	Înțelegerea termenului de funcție. Identificarea scopului unei funcții într-un program.
		4.4.4	Scrierea și definirea unei funcții într-un program.
	4.5 Evenimente și comenzi	4.5.1	Înțelegerea termenului de eveniment. Identificarea scopului unui eveniment într-un program.

CATEGORY	SKILL SET	REF.	TASK ITEM
		4.5.2	Utilizarea evenimentelor apărute la acționarea butoanelor mouse-ului, la apăsarea diverselor taste de pe tastatură, la apăsarea unui buton de comandă, temporizator.
		4.5.3	Utilizarea bibliotecilor, precum: math, random, time.
5 Testare, depanare și lansare	*5.1 Rulare, Testare și Depanare*	5.1.1	Înțelegerea beneficiilor testării și depanării unui program în scopul rezolvării erorilor.
		5.1.2	Înțelegerea tipurilor de erori dintr-un program: erori de sintaxă, erori logice.
		5.1.3	Rularea unui program.
		5.1.4	Identificarea și corectarea unei erori de sintaxă într-un program: scriere incorectă, semne de punctuație lipsă.
		5.1.5	Identificarea și corectarea unei erori logice într-un program: expresie Booleeană incorectă, tip de date incorect.
	5.2 Lansare	5.2.1	Verificarea faptului că programul respectă cerințele din descrierea inițială.
		5.2.2	Descrierea programului final și a scopului acestuia.
		5.2.3	Identificarea îmbunătățirilor ce pot fi aduse programului astfel încât să satisfacă nevoi suplimentare.