

ECDL MODULE

Computing

Syllabus Version 1.0

목적

이 문서는 ECDL Computing 에 대한 강의 내용을 상세히 설명합니다.

이 실라버스는 학습 결과를 통해 ECDL Computing 모듈 응시생이 알아야할 지식과 기술을 설명합니다.

이 실라버스는 이 모듈의 기초적인 이론과 실습 기반의 테스트를 제공합니다.

Copyright © 1997 - 2017 ECDL Foundation

All rights reserved. 이 문서의 어떤 부분도 ECDL 협회가 허용하는 경우를 제외하고는 복제할 수 없습니다. 자료 사용에 대한 승인 문의는 ECDL 협회로 보내 주십시오.

경고문

ECDL 협회는 본 발행물을 준비하는데 있어 모든 주의를 기울였으나 발행자로서 본 실라버스에 포함된 정보의 완벽성에 대해 어떠한 보증도 하지 않을 뿐 아니라, 오류, 누락, 부정확함 및 정보나 지침 또는 자문에 의해 발생하는 어떠한 종류의 손실이나 손해에 대해서도 책임이나 의무를 지지 않습니다. ECDL 협회는 언제든지 사전통지 없이 재량에 따라 내용을 변경할 수 있습니다.

COMPUTING

이 모듈은 단순한 컴퓨터 프로그램을 작성하기 위해 전산 사고와 코딩의 사용과 관련된 필수 개념과 기술을 설명합니다.

모듈 목표

이 모듈의 자격을 취득한 응시생은 다음을 수행할 수 있게 됩니다.

- 컴퓨팅과 관련된 핵심 개념과 프로그램 개발과 관련된 일반적인 활동을 이해할 수 있습니다.
- 문제를 분석하고 솔루션을 개발하기 위해 문제 분해(problem decomposition), 패턴인식(pattern recognition), 추상화(abstraction), 알고리즘(algorithm) 등과 같은 컴퓨팅 사고력을 이해하고 사용할 수 있습니다.
- 플로우차트(flowchart)나 의사코드(pseudocode)를 사용하여 프로그램에 대한 알고리즘을 작성, 테스트하고 수정할 수 있습니다.
- 코딩과 관련된 주요 원리나 용어, 체계적이고 문서화된 코드의 중요성을 이해할 수 있습니다.
- 프로그램에서 변수, 데이터 유형과 로직과 같은 프로그래밍 구조를 이해하고 사용할 수 있습니다.
- 반복, 조건문, 프로시저와 함수뿐만 아니라 프로그램의 이벤트, 명령을 사용하여 효율성과 기능을 향상시킬 수 있습니다.
- 릴리즈하기 전에 프로그램 테스트와 디버그를 하고 요구사항을 충족했는지 확인할 수 있습니다.

범주	지식 영역	참조 번호	지식 항목
1.1 컴퓨팅	1.1. 주요 개념	1.1.1.	컴퓨팅(computing)이라는 용어를 정의한다.
		1.1.2	전산적 사고(computational thinking)라는 용어를 정의한다.
		1.1.3	프로그램(program)이라는 용어를 정의한다.
		1.1.4	코드(code)를 정의한다. 소스 코드(source code)와 기계어 코드(machine code)를 구별한다.
		1.1.5	프로그램 기술(description)과 명세(specification)를 이해한다.
		1.1.6	분석, 설계, 프로그래밍, 테스트, 개선 등과 같은 프로그램 작성의 일반적인 활동을 인식한다.
		1.1.7	형식 언어(formal language)와 자연어(natural language)의 차이점을 이해한다.

범주	지식 영역	참조 번호	지식 항목
2. 전산적 사고방식	2.1 문제 분석	2.1.1	전산적 사고에 사용되는 전형적인 방법인 분해, 패턴 인식, 추상화, 알고리즘 등을 이해한다.
		2.1.2	데이터, 프로세스, 복잡한 문제 등을 작은 부분으로 나누기 위하여 문제를 분해한다.
		2.1.3	작고 분해된 문제들에서 패턴을 확인한다.
		2.1.4	문제를 분석할 때 불필요한 세부 사항을 필터링하기 위하여 추상화한다.
		2.1.5	전산적 사고에서 알고리즘이 어떻게 사용되는지 이해한다.
	2.2 알고리즘	2.2.1	순차(sequence)라는 용어를 정의한다. 알고리즘을 설계할 때 순차의 목적을 이해한다.
		2.2.2	플로우차트(flowcharts), 의사코드(pseudocode) 등과 같은 문제의 표현 방법을 이해한다.
		2.2.3	시작/중지, 프로세스, 판단, 입출력, 커넥터, 화살표 등과 같은 순서도 기호를 이해한다.
		2.2.4	플로우차트, 의사코드에 기술된 연산의 순서를 이해한다.
		2.2.5	플로우차트, 의사코드와 같은 기법에 근거하여 정확한 알고리즘을 작성한다.
		2.2.6	프로그램 요소 누락, 잘못된 순서, 잘못된 판단문 등과 같은 알고리즘 오류를 수정한다.
3. 코드 기초	3.1 시작하기	3.1.1	들여쓰기(indentation), 적절한 주석(comment), 의미있는 이름(descriptive naming) 등과 같은 구조적이며 문서화된 코드의 특성을 설명한다.
		3.1.2	프로그램에서 +, -, /, * 등과 같은 간단한 산술 연산자를 사용하여 계산한다.
		3.1.3	복잡한 표현에서 연산자 우선순위와 평가 순서를 이해한다. 복잡한 표현식을 구조화하기 위해 괄호를 사용하는 방법을 이해한다.
		3.1.4	파라미터(parameter, 매개 변수)라는 용어를 이해한다. 프로그램에서 파라미터의 목적을 설명한다.
		3.1.5	주석(comment)이라는 용어를 정의한다. 프로그램에서 주석의 목적을 이해한다.
		3.1.6	프로그램에서 주석을 사용한다.
	3.2 변수와 데이터 타입	3.2.1	변수(variable)라는 용어를 정의한다. 프로그램에서 변수의 목적을 이해한다.
		3.2.2	변수를 정의하고 초기화한다.
		3.2.3	변수에 값을 할당한다.
		3.2.4	프로그램에서 계산이나 값을 저장하기 위해 적절한 이름의 변수를 사용한다.

범주	지식 영역	참조 번호	지식 항목
4. 코드 만들기	4.1 논리문	3.2.5	프로그램에서 string, character, integer, float, Boolean 등과 같은 데이터 타입을 사용한다.
		3.2.6	프로그램에서 array, list, tuple 등과 같은 집합체 데이터 타입을 사용한다.
		3.2.7	프로그램에서 입력문을 사용하여 사용자의 데이터를 입력한다.
		3.2.8	프로그램에서 출력문을 사용하여 화면에 데이터를 출력한다.
		4.1.1	논리 테스트라는 용어를 정의한다. 프로그램에서 논리 테스트의 목적을 이해한다.
		4.1.2	=, >, <, >=, <=, <>, !=, ==, AND, OR, NOT 등을 사용하는 부울 논리식의 결과는 true 또는 false 임을 이해한다.
		4.1.3	프로그램에서 부울 논리식을 사용한다.
	4.2 반복문	4.2.1	루프(loop)라는 용어를 정의한다. 프로그램에서 루프의 목적과 장점을 이해한다.
		4.2.2	for, while, repeat 등과 같은 루프 유형을 숙지한다.
		4.2.3	for, while, repeat 등과 같은 루프를 사용한다.
		4.2.4	무한 루프(infinite loop)라는 용어를 이해한다.
		4.2.5	재귀(recursion)라는 용어를 이해한다.
	4.3 조건문	4.3.1	조건문(conditional statement)이라는 용어를 정의한다. 프로그램에서 조건문의 목적을 이해한다.
		4.3.2	프로그램에서 IF ... THEN ... ELSE 조건문을 사용한다.
	4.4 프로시저와 함수	4.4.1	프로시저(procedure)라는 용어를 이해한다. 프로그램에서 프로시저의 목적을 이해한다.
		4.4.2	프로그램에서 프로시저를 작성하고 프로시저 이름을 만든다.
		4.4.3	함수(function)라는 용어를 이해한다. 프로그램에서 함수의 목적을 이해한다.
		4.4.4	프로그램에서 함수를 작성하고 함수 이름을 만든다.
	4.5 이벤트와 커맨드	4.5.1	이벤트(event)라는 용어를 이해한다. 프로그램에서 이벤트의 목적을 이해한다.
		4.5.2	마우스 클릭, 키보드 입력, 버튼 클릭, 타이머와 같은 이벤트 핸들러(event handler)를 사용한다.
		4.5.3	수학함수, 난수함수, 시간함수 등과 같은 유용한 라이브러리를 사용한다.

범주	지식 영역	참조 번호	지식 항목
5. 테스트, 디버그 및 릴리스	5.1 실행, 테스트 및 디버그	5.1.1	프로그램 오류를 해결하기 위한 테스트 및 디버깅의 장점을 이해한다.
		5.1.2	구문 오류, 논리오류 등과 같은 프로그램의 오류 유형을 이해한다.
		5.1.3	프로그램을 실행한다.
		5.1.4	프로그램에서 맞춤법 오류, 구두점 누락 등과 같은 구문 오류를 식별하고 수정한다.(mbps), 초당 기가비트 (gbps).
		5.1.5	프로그램에서 잘못된 부울 표현식, 잘못된 데이터 타입 등과 같은 논리 오류를 식별하고 수정한다.
	5.2 릴리스	5.2.1	초기 요구 사항과 프로그램이 일치하는지 확인한다.
		5.2.2	완성된 프로그램, 커뮤니케이션 의도와 값을 설명한다.
		5.2.3	추가적이거나 관련된 요구 사항을 충족시킬 수 있도록 프로그램에서 개선 및 향상시킬 수 있는 부분을 확인한다.