

제 1장 프로그래밍의 개념

1.1. 프로그래밍이란?

(중간점검)

1. 왜 계산기는 컴퓨터라고 할 수 없는가?

정답 : 계산기는 수행하는 작업이 고정되어 있으므로 컴퓨터라고 할 수 없다.

2. 컴퓨터가 가장 쉽게 이해하는 언어는 무엇인가?

정답 : 이진수로 되어 있는 기계어

3. 컴파일러는 어떤 역할을 하는가?

정답 : 고급 프로그래밍 언어를 기계어로 번역한다.

1.2. 프로그래밍 언어

(중간점검)

1. ()은 과학 기술 계산에 주로 사용된다.

2. ()은 컴퓨터 교육에 사용할 목적으로 제작되었다.

3. ()은 C에 객체지향 개념을 추가한 언어로서 C의 상위집합이다.

4. ()은 기업의 사무 처리에 적합하다.

5. ()은 인터넷 상에서 실행이 용이하도록 만들어진 언어이다.

6. C언어를 기초로 만들어진 대표적인 2가지 언어는 ()과 () 이다.

정답

1. FORTRAN

2. PASCAL

3. C++

4. COBAL

5. Java

6. C++, Java

1.3. C언어의 소개

(중간점검)

1. 임베디드 시스템이란 무엇인가?

정답 : 임베디드 시스템(embedded system)은 일종의 컴퓨터 시스템으로 한 가지 기능만 수행되도록 설계되었으며 하드웨어에 내장하여 특수한 기능만을 수행한다. 임베디드 시스템은 오늘날 일상 생활에 쓰이는 많은 장치들을 제어하고 있다.

2. C언어의 장점과 단점을 정리하여 보자

정답 : 장점 - 간결하고 효율적이며 이식성이 뛰어나다.

단점 - 배우기 어렵다. 객체 지향 프로그래밍이 힘들다.

1.4. 알고리즘이란?

(중간점검)

1. 친구에게 전화를 거는 알고리즘을 만들어 보자

정답 : 수화기를 든다. -> 상대방의 전화번호를 누른다. -> 통화를 한다.
-> 수화기를 내려놓는다.

2. 세탁기를 이용하여 세탁을 하는 알고리즘을 만들어 보자

정답 : 세탁기를 켜다. -> 세탁기의 뚜껑을 연다. -> 옷과 세제를 넣는다.
-> 세탁기의 뚜껑을 닫는다. -> 시작 버튼을 누른다. -> 전원을 차단한다.

1.5. 프로그램 개발 과정

(중간점검)

1. 프로그램 개발 과정을 순서대로 정리하여 보라.

정답 : 요구사항분석 -> 알고리즘 작성 -> 소스 작성 -> 컴파일과 링크 ->
프로그램 실행과 디버깅

2. 소스 파일의 이름으로 test.txt는 올바른가?

정답 : 올바르지 않다. 확장자로 .c를 붙여야 한다.

3. 소스 파일, 오브젝트 파일, 실행 파일의 차이점을 설명하라

정답 : 소스 파일 - 프로그래밍 언어로 작성한 원본 파일
오브젝트 파일 - 소스 파일을 기계어로 컴파일한 파일
실행 파일 - 오브젝트 파일들을 모아서 시동 코드와
라이브러리 코드를 붙인 파일

4. 소스 파일이 test.c라면 컴파일 과정을 거친 후에 생성되는 오브젝트 파일과 실행 파일의 이름은 어떻게 되는가?

정답 : 오브젝트 파일 - test.obj
실행파일 - test.exe

5. 컴파일과 링크 과정을 거쳐서 실행 파일을 만든 다음에 소스 파일과 오브젝트 파일을 보관해야 하는가, 아니면 삭제하여도 되는가?
그 이유를 말하라.

정답 : 소스 파일은 반드시 보관하여야 한다.
오브젝트 파일은 삭제하여도 된다.

6. 디버깅(debugging)이란 무엇인가?

정답 : 원하는 실행 결과가 나오지 않는 경우에 “소스 수정 -> 컴파일 -> 링크 -> 실행”의 단계를 다시 거치는 오류 수정 작업

7. 왜 소프트웨어도 유지 보수가 필요한가?

정답 : 오류가 남아있을 수도 있고 사용자의 요구조건이 변경될 수 있다.

1.6. 스크래치

(중간점검)

1. 고양이를 무한히 회전시키는 스크립트를 작성하고 실행하여 보자.

정답



2. <표 1-2>에 나오는 스크립트를 작성하여서 실행하여 보고 어떤 동작을 하는 지를 조사하여 보자. 필요하다면 변수도 생성한다.

정답



와 같은 입력을 한 다음 실행한다.

“조건부실행”과 “변수”에서는 변수 생성이 필요하다. 변수는 아직 학습하지 않았지만 값을 저장하고 있는 박스라고 생각할 수 있다.



변수는 에서 "Make a variable"을 클릭한다.