필기

권우석 샘의 사무자동화산업기사

[3. 프로그래밍 일반]

[P 1강]-운영체제, 프로그래밍 언어 종류

1. 운영체제 기능적 분류 > 제어 프로그램 ★★☆☆☆

: 시스템 전체의 작동 상태 감시, 작업의 순서 지정, 작업에 사용되는 데이터 관리 등의 역할을 수행하는 P/G

① 감시 프로그램 (Supervisor Program)
② 작업 제어 프로그램 (Job Control Program)
: 작업의 연속 처리를 위한 스케줄 및 시스템 자원 할당 등을 담당한다.
③ 데이터 관리 프로그램 (Data Management Program)
: 주기억장치와 보조기억장치 사이의 자료 전송, 파일의 조작 및 처리, 입/출력 자료와 프로그램간의 논리적 연결 등, 시스템에서 취급하는 파일과 데이터를 표준적인 방법으로 처리할 수 있도록 관리

2. 운영체제 기능적 분류 > 처리 프로그램 ★★☆☆☆

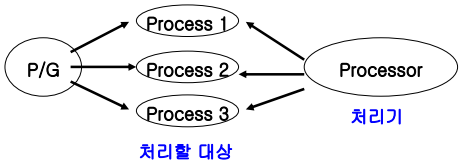
: 제어 프로그램의 지시를 받아 사용자가 요구한 문제를 해결하기 위한 프로그램

① 서비스 프로그램 (Service Program)
: 효율성을 위해 사용 빈도가 높은 P/G
② 문제 프로그램 (Problem Program)
: 특정 업무 해결을 위해 사용자가 작성한 P/G
③ 언어 번역 프로그램 (Language Translator Program) : 어셈블러, 컴파일러, 인터프리터

[P 1강]-운영체제, 프로그래밍 언어 종류

3. 프로세스(Process) 정의

- 주기억장치에 저장된 프로그램 (실행중인 프로그램)
- 운영체제가 관리하는 최소 단위의 작업
- 비동기적(비연속적) 행위를 일으키는 주체
- PCB를 가진 프로그램
- 프로세서가 할당되는 실체
- 디스크(보조기억장치)에 저장된 프로그램 (X)



4. 프로세스 제어 블록 : PCB (Process Control Block) ★☆☆☆☆

- 운영체제가 프로세스에 대한 중요한 정보를 저장해 놓은 곳 (프로세스 정보 리스트)
- 각 프로세스가 생성될 때마다 PCB가 생성되고, 완료되면 PCB는 제거
- 프로세스 이름 및 고유 식별자
- 프로세스 우선 순위
- 프로세스의 현재 상태

[P 1강]-운영체제, 프로그래밍 언어 종류

[07년5월][00년7월][00년10월][03년3월][05년3월]
1. 운영체제의 제어 프로그램(Control Program)에 해당하는 것은?
가. 언어번역(Language Translator) 프로그램
나. 서비스(Service) 프로그램
다. 자료관리(Data Management) 프로그램
라. 문제(Problem) 프로그램

[07년8월][07년3월][06년8월][03년8월][02년5월]
2. 운영체제를 기능상 분류했을 경우 제어프로그램에 해당하지 않는 것은?
가. 감시프로그램 나. 작업제어프로그램
다. 언어번역프로그램 라. 데이터관리프로그램

[06년5월][01년6월][03년5월][99년10월][05년3월][01년9월][02년3월][01년3월]
3. 운영체제를 기능상 분류했을 대 처리(processing) 프로그램에 해당하지 않는 것은?
가. language translation program
나. service program
다. problem program
라. supervisor program

[04년3월][03년3월][01년3월]
4. PCB(process control block)의 포함 정보가 아닌 것은?
가. 프로세스의 현재 상태
나. 프로세스의 생성율 및 부재율
다. 프로세스의 고유 식별자
라. 프로세스의 우선순위

[정답] 1.다 2.다 3.라 4.나

[P 1강]-운영체제, 프로그래밍 언어 종류

5. Interrupt 종류 ★★☆☆☆

1) 외부 인터럽트
- 시스템 타이머에서 일정한 시간이 만료된 경우나 오퍼레이터가 콘솔상의 인터럽트 키를 입력한 경우 발생

2) 입출력 인터럽트
- CPU에 채널이나 입, 출력 기기의 변화를 알리거나 데이터의 I/O종료, 오류 발생시 발생

3) SVC (Supervisor call interrupt)
- 프로그래머에 의해 발생하는 인터럽트로서, 보통 입/출력 수행, 기억장치 할당, 오퍼레이터와의 대화를 위해 발생 (운영체제 제어프로그램인 감시 프로그램 호출)

[04년5월]
1. 시스템 타이머에서 일정한 시간이 만료된 경우나 오퍼레이터가 콘솔상의 인터럽트 키를 입력한 경우 발생하는 인터럽트는?
가. 입/출력 인터럽트
나. 외부 인터럽트
다. SVC 인터럽트
라. 프로그램 검사 인터럽트

[정답] 1.나 2.라

[06년8월][06년3월][03년3월][01년3월][06년5월][01년6월]
2. 프로그래머에 의해 발생하는 인터럽트로서, 보통 입/출력 수행, 기억장치 할당, 오퍼레이터와의 대화를 위해 발생하는 것은?
가. Program check interrupt
나. External interrupt
다. I/O interrupt
라. Supervisor call interrupt

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 1강]-운영체제, 프로그래밍 언어 종류

6. 프로세스 스케줄링 (= CPU 스케줄링)

- 정의 : 컴퓨터 시스템의 성능을 높이기 위해 그 사용 순서를 결정하기 위한 정책

7. 프로세스 스케줄링 기법

1) 비선점 스케줄링 (Non Preemptive) : 비효율적, 비양보
- 프로세스에게 이미 할당된 CPU를 강제로 빼앗을 수 없고, 사용이 끝날 때까지 기다려야 하는 방법
- 일괄 처리(오버헤드 발생 X), 실시간 처리가 안되므로 중요한 작업이 기다리는 경우 발생
- 대표적인 스케줄링 : FIFO, SJF, HRN

2) 선점 스케줄링 (양보) : 효율적
- 우선 순위가 높은 다른 프로세스가 할당된 CPU를 강제로 빼앗을 수 있는 방법
- 실시간 처리, 대화식 시분할 처리(오버헤드 발생 O)
- 대표적인 스케줄링 : RR, SRT

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 1강]-운영체제, 프로그래밍 언어 종류

8. 페이지 교체(Replacement) 알고리즘

1) 정의
- 페이지 부재(page fault)가 발생하였을 경우, 가상기억장치의 필요한 페이지를 주기억장치의 어떤 페이지 프레임을 선택, 교체 해야 하는 가를 결정하는 기법

2) 종류
① OPT (OPTimal replacement, 최적교체)
- 앞으로 가장 오랫동안 사용하지 않을 페이지를 교체하는 기법 (실현 가능성X)

② FIFO (First In First Out)
- 가장 먼저 들여온 페이지를 먼저 교체시키는 방법 (주기억장치 내에 가장 오래 있었던 페이지를 교체)

③ LRU (Least Recently Used)
- 최근에 가장 오랫동안 사용하지 않은 페이지를 교체하는 기법

④ LFU (Least Frequently Used)
- 사용 횟수가 가장 적은 페이지를 교체하는 기법

⑤ NUR (Not Used Recently)
- 각 페이지 당 두 개의 하드웨어 비트를 두어서 가장 최근에 사용하지 않은 페이지를 교체하는 기법

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 1강]-운영체제, 프로그래밍 언어 종류

9. 프로그램 종류 ★★★☆☆

1) 디버깅(debugging, debugger)
- 프로그램 개발 과정에서 프로그램 안에 내재해 있는 논리적 오류를 발견하고 수정하는 작업

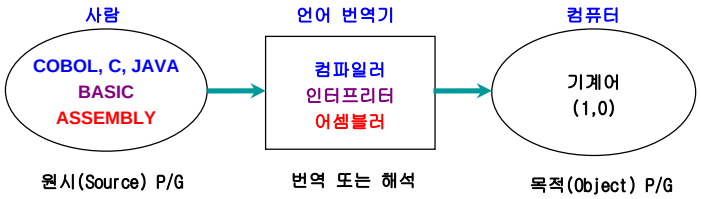
2) 응용 프로그램(application program)
- 특정한 작업을 수행할 수 있도록 사용자가 개발한 프로그램 (ex.계산기)

3) 펌웨어(firmware)
- 롬(ROM)에 기록된 하드웨어를 제어하는 마이크로프로그램의 집합
- 소프트웨어와 하드웨어의 특성을 모두 가지고 있다고 할 수 있다.

4) Simulation
- 실제의 실험이 불가능하거나 시간적, 경제적으로 어려움이 많은 경우 또는 해석적인 방법으로 해답을 구할 수 없는 경우에 대한 가상 모의실험

[P 1강]-운영체제, 프로그래밍 언어 종류

10. 프로그래밍 언어 ★★



1) 저급언어 → 컴퓨터가 이해하기 쉬운 언어
① 기계어
- 컴퓨터가 직접 이해할 수 있어 실행 속도가 빠르다 .
- 프로그램의 유지보수가 어려움
- 호환성이 없고 기계마다 언어가 다르다.
- 2진수를 사용하여 데이터를 표현한다.

② 어셈블리어 (Assembly) : 기계어와 가장 유사한 언어

2) 고급언어 → 사람이 이해하기 쉬운 언어
① FORTRAN : 수학, 과학, 공학 등과 같은 수리 계산 분야에 널리 사용되는 언어

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 1강]-운영체제, 프로그래밍 언어 종류

② COBOL : 사무용 자료처리 언어
- IDENTIFICATION DIVISION (식별부) : 프로그램 이름, 작성자 등
- ENVIRONMENT DIVISION (환경부) : 입/출력, 자료구조 등
- DATA DIVISION (데이터부) : 데이터를 저장할 변수 정의 등
- PROCEDURE DIVISION (절차부) : 프로그램의 알고리즘
* 반복문 : PERFORM 문 (C 언어의 FOR문에 해당)

③ SNOBOL : 문자열 대치, 복사, 치환 등과 같은 문자열의 조작을 편리하게 수행할 수 있도록 여러 가지 기능을 제공하며, 스트링 자료 활용의 예가 많은 언어

④ LISP : 인공지능 소프트웨어를 만들기 위하여 사용하는 프로그래밍 언어 (연결리스트 사용)
- 인터프리터, 해석, 선언문을 전혀 사용하지 않는 언어

⑤ BASIC : 퍼스널컴퓨터에서 이용되는 간단한 언어
- 인터프리터 언어

⑥ PROLOG : 비절차적 언어 (데이터베이스 관리 시스템에서 이용)

⑦ Ada : 군사용

⑧ C : 시스템 프로그래밍 언어

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 1강]-운영체제, 프로그래밍 언어 종류

[07년8월]
1. 선점형 스케줄링 방식에 해당하는 것은?
가. FIFO 나. SJF
다. Round-Robin 라. HRN

[07년3월][06년5월][05년8월]
2. 프로그램 개발 과정에서 프로그램 안에 내재해 있는 논리적 오류를 발견하고 수정하는 작업을 무엇이라고 하는가?
가. 링킹(linking) 나. 바인딩(binding)
다. 로딩(loading) 라. 디버깅(debugging)

[07년8월][00년10월][02년3월][03년3월][01년9월][06년8월]
3. 특정한 작업을 수행할 수 있도록 사용자가 개발한 프로그램을 일반적으로 무엇이라 하는가?
가. System program 나. operating system
다. application program 라. compiler

[03년3월][02년3월][00년7월][01년9월]
4. 어셈블리에서 16진수 상수를 정의한 명령어는?
가. DC CL3"A2" 나. DC XL3"A2"
다. DC BL3"111" 라. DC PL3"38"

[06년8월][99년6월][99년8월][02년3월][01년6월]
5. COBOL 언어의 PERFORM 문, C 언어의 FOR문에 해당되는 것은?
가. 반복문 나. 종료문
다. 입출력문 라. 선언문

[02년5월][02년3월][01년9월][00년7월][06년8월][03년8월][03년5월][04년5월][05년3월][01년3월][02년8월][00년10월]
6. 문자열 대치, 복사, 치환 등과 같은 문자열의 조작을 편리하게 수행할 수 있도록 여러 가지 기능을 제공하며, 스트링 자료 활용의 예가 많은 언어는?
가. SNOBOL 나. C
다. PL/1 라. ADA

[정답] 1.다 2.라 3.다 4.나 5.가 6.가

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 1강]-운영체제, 프로그래밍 언어 종류

[07년5월][06년8월][06년3월][06년5월][02년8월][05년3월][99년8월]
7. 인터프리터(Interpreter) 기법을 사용하는 언어는?
가. BASIC 나. C
다. FORTRAN 라. COBOL

[07년5월][06년8월][05년3월][04년8월][99년8월][06년5월][00년10월][02년8월][03년8월][03년5월][03년3월][02년5월]
8. 시스템 프로그래밍 언어로서 가장 적당한 것은?
가. C 나. COBOL
다. PASCAL 라. FORTRAN

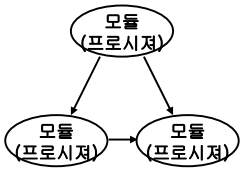
[정답] 7.가 8.가 9.라 10.가

[03년3월][00년10월]
9. 고급 프로그래밍언어에 관한 설명 중 옳지 않은 것은?
가. COBOL언어는 회사의 사무용 자료처리 언어로 개발되었으며, 기계 독립적인 부분과 기계 종속적인 부분을 분리하는데 성공한 언어이다.
나. PASCAL언어는 간결하면서도 강력한 언어로 손꼽히고 있으며, 교육용 언어로는 뛰어나다는 평가를 받고 있다.
다. FORTRAN은 과학 계산용 언어로서, 뛰어난 실행 효율성으로 성공한 언어이며, 번역기를 구현한 최초의 고급 언어로 평가된다.
라. C 언어는 고급 언어 프로그래밍과 저급 언어 프로그래밍도 가능한 언어이며, 인터프리트 방식의 대표적 언어이다.

[07년3월]
10. 고급 언어에 대한 설명으로 옳지 않은 것은?
가. 컴파일 과정 없이 실행 가능하다.
나. 저급 언어보다 배우기 쉽다.
다. 기종 간에 큰 차이가 없어 호환성이 높다.
라. COBOL은 고급 언어에 해당한다.

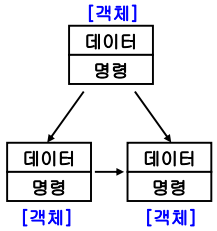
[P 1강]-운영체제, 프로그래밍 언어 종류

11. 절차적(구조적) 개발 VS 객체지향 개발 ★



[장점]
구조 단순 -> 이해 O, 수정 O, 정확 O (C언어)

[단점]
소프트웨어 재사용, 유지보수 어려움 → 소프트웨어 위기 해결 안됨

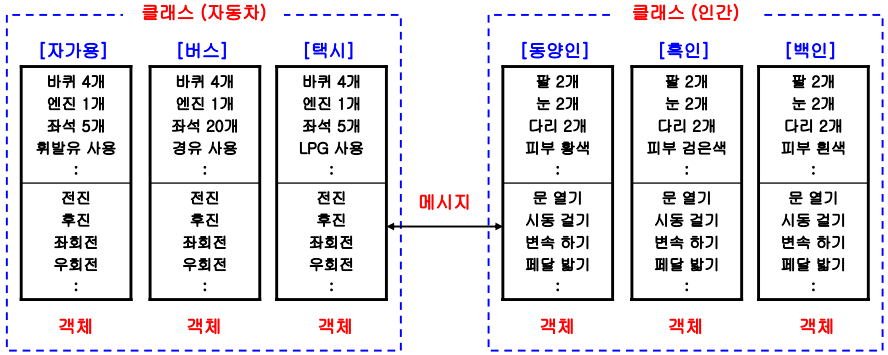


[장점]
현실 세계를 프로그램에 반영 (C++, Ada95, Smalltalk, Delphi)
소프트웨어 재사용, 유지보수 향상 → 소프트웨어 위기 해결 방안

[관련 용어]
기본 : 객체, 클래스, 메시지
원칙 : 캡슐화, 정보 은폐, 추상화, 상속성, 다형성
데이터 = 상태, 속성(Attribute), 변수, 자료구조
명령(연산) = 행위, 메소드(Method), 동작(Operation)

[P 1강]-운영체제, 프로그래밍 언어 종류

12. 객체, 클래스, 메시지 - 개념 이해하기



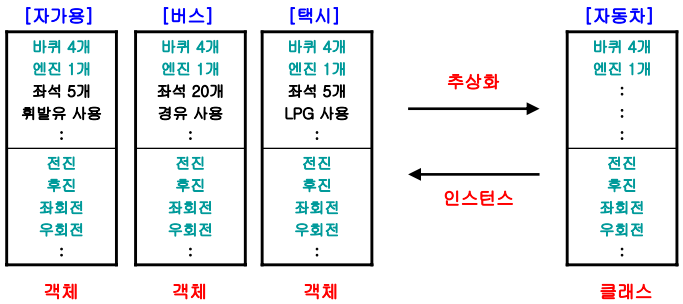
1) 객체 (Object)
- 현실 세계의 개체며 객체들 간의 상호작용은 메시지를 통해 이루어짐
① 데이터 : 객체가 가지고 있는 상태
② 연산 : 객체의 데이터를 처리하는 행위 (메소드, function)

[P 1강]-운영체제, 프로그래밍 언어 종류

2) 클래스 (Class)
- 하나 이상의 유사한 객체들을 묶어서 하나의 공통된 특성을 표현한 객체지향의 요소

3) 메시지 (Message)
- 객체들 간에 상호작용을 하는데 사용되는 수단
- 객체에서 객체로 메시지가 전달되면 메소드(행위)를 시작함

4) 메소드(method)
- 객체지향 개념에서 객체가 메시지를 받아 실행해야 할 객체의 구체적인 연산



[P 1강]-운영체제, 프로그래밍 언어 종류

13. 정보은폐 (Information Hiding)

- 객체는 다른 객체로부터 자신의 자료를 숨기고 자신의 연산만을 통하여 접근을 허용하는 것
- 왜? 고려되지 않은 영향들을 최소화하기 위해

14. 상속 (Inheritance)

- 상위 클래스의 메소드와 속성을 하위 클래스가 물려받는 것

15. 다형성 (Polymorphism)

- 한 메시지가 객체에 따라 다른 방법으로 응답할 수 있는 것
- 많은 상이한 클래스들이 동일한 메소드명을 이용하는 능력

[P 1강]-운영체제, 프로그래밍 언어 종류

[04년3월][00년7월]
1. 객체지향 프로그래밍 언어(Object-oriented programming language)가 절차지향 프로그래밍 언어(Procedure -oriented programming language)에 비해 특히 우수한 점은?
가. 구조화 프로그래밍(structured programming)이 가능하다.
나. 함수(function)를 자유자재로 사용할 수 있다.
다. 컴파일시 실행파일(executable file)의 속도가 향상된다.
라. 유지보수성(maintainability)과 재사용성(reusability) 이 좋다.

[06년5월][05년8월][04년5월][06년3월][99년8월][01년6월]
2. 하나 이상의 유사한 객체들을 묶어서 하나의 공통된 특성을 표현한 객체지향의 요소는?
가. 추상화 나. 객체
다. 메시지 라. 클래스

[정답] 1.라 2.라 3.다 4.나 5.라

[07년8월][03년5월][06년3월]
3. 객체의 외부적인 활동을 연산이라는 전제하에서 구현한 것은?
가. 속성 나. 메시지
다. 메소드 라. 추상화

[07년8월][03년3월][05년3월]
4. 객체지향 언어에서 객체(object)의 구성을 나타낸 것은?
가. object = program +operator
나. object = member function +data
다. object = class + class
라. object = class +member function

[02년3월][99년6월]
5. 객체지향언어(Object-Oriented Programming Language)에서 상위의 클래스가 정의한 기능과 특성을, 하위의 클래스가 이어 받는 것을 무엇이라 하는가?
가. 자료 추상화(data abstraction)
나. 다형성(polymorphism)
다. 은닉화(encapsulation)
라. 상속성(Inheritance)

[P 1강]-운영체제, 프로그래밍 언어 종류

[08년3월]
1. HRN 스케줄링 기법에서 우선순위를 구하는 방법은?
가. 대기시간/서비스를 받을 시간
나. 서비스를 받을 시간/대기시간
다. 서비스를 받을 시간/(대기시간+서비스를 받을 시간)
라. (대기시간+서비스를 받을 시간)/서비스를 받을 시간

비선점 > HRN (Highest response ratio Next)

- SJF 방식의 단점(긴 작업과 짧은 작업간의 지나친 불평등)을 보완하는 기법
- 우선순위 계산식 : (대기 시간+서비스 시간)/서비스 시간

문제) 우선 순위가 가장 높은 작업

작 업	대기시간	서비스시간
A	5	5
B	10	6
C	15	7
D	20	8

A : (5+5)/5 = 2
B : (10+6)/6 = 2.666
C : (15+7)/7 = 3.14
D : (20+8)/8 = 3.5

[정답] 1.라

[P 2강]-프로그램 수행순서

1. 프로그램 수행 순서 ★★☆☆☆

원시(source) P/G → 번역(컴파일러) → 목적(object) P/G 생성 → 링커 → 로더 → 실행

컴파일러, 인터프리터, 어셈블러

2. 링커 (linkage editor) ★☆☆☆☆

- 독자적으로 번역된 여러 개의 목적 프로그램과 프로그램에서 사용되는 내장 함수들을 하나로 모아서 컴퓨터에서 실행될 수 있는 실행 프로그램을 생성하는 프로그램
- 재배치 형태의 기계어로 된 여러 개의 프로그램을 묶어서 로드 모듈을 작성하는 것

3. 로더 ★★★☆☆

: 목적 P/G을 주기억장치에 적재하여 실행 가능하도록 해주는 시스템 프로그램

1) 기능 : 할당(Allocation), 연결(Link), 재배치(Relocation), 적재(Load)

2) 종류
- 절대(Absolute) 로더 : 적재 기능만 하는 간단한 로더 (할당,연결-프로그래머, 재배치-언어번역기)

[P 2강]-프로그램 수행순서

[07년3월][01년6월][00년10월][03년3월][99년10월][03년5월][99년8월][02년5월]
1. 프로그램 수행 순서가 옳은 것은?

① 링커	② 원시 프로그램	③ 로더
④ 컴파일러	⑤ 목적 프로그램	

가. ②→④→⑤→①→③ 나. ⑤→④→②→①→③
다. ②→③→④→①→⑤ 라. ④→①→③→②→⑤

[03년3월][02년8월][99년4월]
2. 재배치 형태의 기계어로 된 여러 개의 프로그램을 묶어서 로드 모듈을 작성하는 것은?
가. 로더(loader)
나. 운영체제(operating system)
다. 프리프로세서(preprocessor)
라. 링키지 에디터(linkage editor)

[06년5월][00년10월][05년3월]
3. 다음 프로그램 중 성격이 나머지 셋과 다른 것은?
가. Assembler 나. Compiler
다. Interpreter 라. Linker

[05년5월][03년5월][02년5월][05년3월][04년3월]
4. 절대로더의 기능별 행위 주체의 연결이 옳지 않은 것은?
가. 기억 장소 할당-프로그래머
나. 연결-로더
다. 재배치-어셈블러
라. 적재-로더

[07년5월][02년5월][06년5월][03년8월][04년5월][04년8월]
5. 로더의 기능이 아닌 것은?
가. Allocation 나. Linking
다. Compile 라. Relocation

[정답] 1.가 2.라 3.라 4.나 5.다

[P 2강]-프로그램 수행순서

4. 고급언어 번역기 (인터프리터, 컴파일러) ★★★★☆

구 분		컴파일러	인터프리터
공 통 점		고급 언어 → 기계어	
차이점	번역 단위	전체를 번역	줄 단위로 번역
	목적 프로그램 생성 여부	생성 O	생성 X
	실행속도	빠르다	느리다.

1) 컴파일러 : 목적코드 생성 → 반복 수행시 효율적

2) 인터프리터 : 줄 단위 번역 → 대화형식의 프로그래밍이 가능(융통성, 시뮬레이션) → 간단한 프로그램

[P 2강]-프로그램 수행순서

[04년3월][00년3월]
1. 번역기(Compiler)와 인터프리터(Interpreter)에 대한 설명으로 거리가 먼 것은?
가. 컴파일러는 원시어가 고급언어이다.
나. 인터프리터를 사용하면 대화형식의 프로그래밍이 가능하게 된다.
다. 실행 시간의 효율성을 중시하는 프로그래밍 언어는 대부분 인터프리터를 사용한다.
라. 컴파일러의 단점 중 하나는, 번역된 산출물인 목적코드가 큰 기억장치를 요한다는 것이다.

[07년8월][03년8월][00년3월]
2. 컴파일러와 인터프리터에 관한 설명으로 옳은 것은?
가. 포트란, 코볼은 컴파일러 언어에 해당한다.
나. 인터프리터는 원시프로그램을 번역하여 목적프로그램을 생성한다.
다. 인터프리터는 반복적으로 실행하는 프로그램에서 실행 시간이 빠르다.
라. 컴파일러는 원시프로그램을 번역하여 목적프로그램을 생성 하지 않는다.

[정답] 1.다 2.가 3.라 4.라 5.라

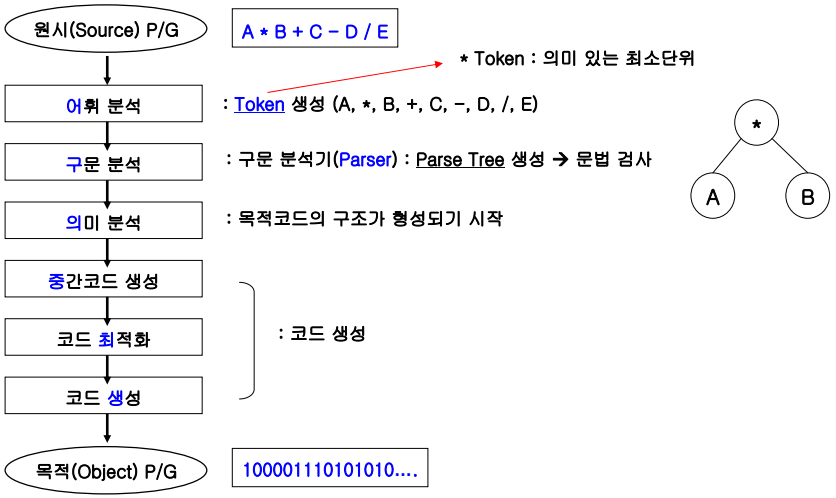
[99년4월]
3. 다음 중 성격이 다른 하나는?
가. 어셈블러 나. 인터프리터
다. 컴파일러 라. 프리프로세서

[07년5월][05년3월][01년6월][99년4월]
4. 인터프리터 기법을 사용하는 경우의 특징이 아닌 것은?
가. 사용상에 있어서 융통성(Flexibility)이 있다.
나. 기억장소가 추가로 필요하다.
다. 프로그램을 한 줄씩 번역하여 곧바로 실행시킨다.
라. 반복문이 많을 경우 컴파일 기법에 비하여 유리하다.

[05년8월]
5. 원시프로그램을 컴파일러가 수행되고 있는 컴퓨터의 기계어로 번역하는 것이 아니라, 다른 기종에 맞는 기계어로 번역하는 것은?
가. 프리프로세서 나. 인터프리터
다. 로더 라. 크로스 컴파일러

[P 2강]-프로그램 수행순서

5. 컴파일 과정



[P 2강]-프로그램 수행순서

6. 어휘 분석 (문장 → 문법적 단위) ★★★☆☆

- 원시 프로그램(source program)을 읽어 들여 토큰(token) 이라는 문법적 단위로 분석한다.
- 프로그래머가 프로그램의 설명을 위해 쓴 주석(comment)은 어휘 분석기에서 모두 처리된다.
- 어휘 분석기는 일명 스캐너(scanner)라고도 불리운다.
- 번역의 가장 기본적인 단계로 나열된 문자들을 기초적인 구성요소들인 식별자, 구분 문자, 연산기호, 핵심어, 주석 등으로 그룹화 하는 단계
- 컴파일 과정 중 원시 프로그램을 하나의 긴 스트링으로 보고 원시 프로그램을 문자 단위로 스캐닝하여 문법적으로 의미있는 일련의 문자(토큰)들로 분할해 내는 작업

```
'덧셈 코드
Private Sub cmd1_Click()
    txt3 = Val(txt1) + Val(txt2)
End Sub

'뺄셈 코드
Private Sub cmd2_Click()
    txt3 = Val(txt1) - Val(txt2)
End Sub

'곱셈 코드
Private Sub cmd3_Click()
    txt3 = Val(txt1) * Val(txt2)
End Sub

'나눗셈 코드
Private Sub cmd4_Click()
    txt3 = Val(txt1) / Val(txt2)
End Sub
```

구문 요소

① 주석 : 프로그래머 쓴 프로그램의 설명 (ex. '덧셈 코드)
② 식별자 : 유일한 이름을 가지는 변수 등 (ex. txt1, txt2, txt3)
③ 구분문자 : 구분 표시 (괄호, 반점, 공백)
④ 연산기호 : 연산자 (사칙연산)
⑤ 핵심어 : 특별한 의미를 갖는 고정된 부분 (ex. Private Sub)

[P 2강]-프로그램 수행순서

7. 구문 요소 ★★☆☆☆

[1에서100까지 합계 구하기]

```
Sub main()

'변수 선언
Dim i, Hap As Integer

'100까지 합 구하기
For i = 1 To 100
    Hap = Hap + i
Next i

'합 출력하기
MsgBox Hap

End Sub
```

① 주석 (ex. '100까지 합 구하기)
: 프로그램을 작성하는 과정에서 컴퓨터에 의하여 직접 실행되는 명령어들이 아니라, 프로그램을 읽어 이해하기에 도움이 되는 내용들을 기록한 부분으로 프로그램의 판독성을 향상시키는 요소

② 식별자 (ex. i, Hap)
: 유일한 이름을 가지는 변수 등

③ 구분문자 (괄호, 반점, 공백)
: 문장이나 식과 같은 구문적인 단위의 시작과 끝을 나타내기 위하여 사용되는 구문적 요소

④ 예약어 (ex. For, To, Next)
- 번역과정에서 속도를 높여준다.
- 프로그램의 신뢰성을 향상 시켜줄 수 있다.
- 프로그램을 번역할 때 예약어의 사용은 심볼 테이블 검색 시간을 단축시킨다.
- 예약어의 사용은 오류가 발생하였을 때 오류회복을 가능케 한다.
- 프로그래머가 변수 이름이나 다른 목적으로 사용할 수 없는 핵심어

[P 2강]-프로그램 수행순서

1. 컴파일러의 컴파일 단계로 옳은 것은?

① 어휘분석(lexical analysis)
② 구문분석(syntax analysis)
③ 중간코드 생성
④ 의미분석(sematic analysis)
⑤ 코드생성(code generation)
⑥ 코드 최적화(code optimizatim)

가. ①②④③⑥⑤ 나. ①②④⑤⑥③
다. ①④③⑤⑥② 라. ①②③④⑤⑥

[05년3월][01년6월][03년3월]
2. 어휘분석(Lexical Analysis) 단계에서 주로 하는 일은?
가. 구문 분석 나. 파싱
다. 기억장소 할당 라. 토큰 생성

[정답] 1.가 2.라 3.가 4.가 5.다

[05년5월][01년9월][03년8월][02년3월]
3. 번역의 가장 기본적인 단계로 나열된 문자들을 기초적인 구성요소들인 식별자, 구분 문자, 연산기호, 핵심어, 주석 등으로 그룹화 하는 단계는?
가. 어휘 분석 나. 구문 분석
다. 의미 분석 라. 코드 생성

[07년8월][04년8월]
4. 대부분의 고급 프로그래밍 언어에서 제공하는 예약어에 관한 설명으로 거리가 먼 것은?
가. 예약어의 사용은 프로그램의 판독성을 저해한다.
나. 프로그램을 번역할 때 예약어의 사용은 심볼 테이블 검색시간을 단축시킨다.
다. 예약어의 사용은 오류가 발생하였을 때 오류회복(error recovery)을 가능케 한다.
라. 프로그래머가 변수 이름이나 다른 목적으로 사용할 수 없는 핵심어(key word)이다.

[07년5월][05년3월][99년4월]
5. 특별한 정보는 갖고 있지 않으나, 판독성을 향상시키기 위하여 사용하는 구문 요소는?
가. 핵심어 나. 예약어
다. 잡음어 라. 연산식

[P 2강]-프로그램 수행순서

8. 구문 표기법 > BNF, EBNF, 구문도표 ★

1) BNF : 프로그래밍 언어의 구문형식을 정의하는데 가장 일반적인 표현방식 (Backus-Naur Form)
- 기호
① 정의 : ::=
② 선택, 택일 : | (키보드에서 Shift + ₩)
③ 비종단(non-terminal)표시 : < >

2) EBNF : 확장된 BNF
- 기호
① 반복 : { } (Ex. A ::= {a})

3) 구문도표 : BNF, EBNF 를 그래픽으로 표현
- 기호
① 비종단(non-terminal) : 사각형(▭)
② 종단(terminal) : 원/타원(○)
③ 흐름방향표시 : 화살표(→)

[P 2강]-프로그램 수행순서

[04년8월][99년8월][99년10월]
1. 프로그래밍 언어의 구문형식을 정의하는데 가장 일반적인 표현방식은?
가. Backus-Naur Form 나. Algorithm
다. DNF 라. HIPO

[06년8월][01년6월][00년7월][02년3월]
2. BNF에 사용되는 기호 중 선택의 의미를 갖는 것은?
가. ::= 나. < >
다. | 라. {}

[07년8월][06년5월][00년10월][01년3월][02년8월][02년5월][03년5월][05년5월][05년3월]
3. BNF 심볼에서 정의를 나타내는 것은?
가. ::= 나. <>
다. | 라. -->

[정답] 1.가 2.다 3.가 4.다 5.가

[07년3월][99년8월][05년3월]
4. EBNF에서 { }를 사용하는 이유는?
가. 블록(block)을 나타내기 위해 사용한다.
나. 생략 가능한 것을 나타내기 위해 사용한다.
다. 반복되는 부분을 나타내기 위해 사용한다.
라. 선택사항을 나타내기 위해 사용한다.

[00년5월]
5. PASCAL에서 subrange 형의 BNF로서 옳은 것은?
가. <subrange type> :: = <CONSTANT> ..<CONSTANT>
나. <subrange type> :: = <CONSTANT> ..<VARIABLE>
다. <subrange type> :: = <VARIABLE> ..<VARIABLE>
라. <subrange type> :: = <LETTER> ..<LETTER>

[P 2강]-프로그램 수행순서

9. 구문 분석 (토큰 → 파스 트리) ★

1) 정의 : 주어진 문장이 정의된 문법 구조에 따라 정당하게 하나의 문장으로서 합법적으로 사용될 수 있는가(문법에 맞나)를 확인하는 작업으로 토큰들을 문법에 따라 분석하는 작업을 수행하는 단계

2) 파스트리
- 고급 언어로 작성된 프로그램을 구문 분석하여 파서에 의하여 생성되는 결과물로서, 각각의 문장을 문법 구조에 따라 트리 형태로 구성한 것
- 구문 분석기가 처리한 문장에 대해 그 문장의 구조를 트리 형태로 표현한 것

3) 파싱의 분류
① 상향식 파싱 (Shift Reduce Parser) : 터미널 노드 → 루트(뿌리) 노드로 파스 트리 구성
② 하향식 파싱 (Recursive Descent Parser) : 루트(뿌리) 노드 → 터미널 노드로 파스 트리 구성

10. 의미 분석 (목적코드 구조 형성)

- 컴파일러에 의해 원시 프로그램이 분석될 때 구문 분석기에 의해 인식된 구문구조가 처리되고, 실행 가능한 목적코드의 구조가 형성되기 시작하는 단계

[P 2강]-프로그램 수행순서

[06년3월][03년8월]
1. Top-down Parser에 해당하는 것은?
가. Shift/Reduce Parser
나. LR Parser
다. Recursive Descent Parser
라. Precedence Parser

[03년5월][99년4월]
2. 구문 분석에는 하향식 파싱(Top-down parsing)과 상향식 파싱(Bottom-up parsing)이 있다. 하향식 파싱에 대한 설명으로 옳지 않은 것은?
가. 하향식 구문분석은 입력 문자열에 대한 좌측 유도(left most derivation) 과정으로 볼 수 있다.
나. 파싱할 수 있는 문법에 left recursion 이 없어야 하고 left factoring 을 해야 하므로 상향식 파서보다는 일반적이지 못하다.
다. 루트로부터 preorder 순으로 주어진 문자열에 대해 파스 트리를 구성한다.
라. 터미널 노드에서 뿌리 노드를 만들어 내는 과정으로 뿌리 노드, 즉 시작 기호가 만들어지면 올바른 문장이고 그렇지 않으면 틀린 문장이다.

[정답] 1.다 2.라 3.라 4.다 5.가

[07년5월][99년8월][00년5월][02년8월][02년3월][02년5월][05년5월]
3. 상향식(Bottom-Up) 파서에 해당하는 것은?
가. Predictive Parser
나. LL Parser
다. Recursive Descent Parser
라. Shift Reduce Parser

[07년3월][00년7월][02년3월][04년 5월][05년3월]
4. 구문 분석기가 처리한 문장에 대해 그 문장의 구조를 트리 형태로 표현한 것을 무엇이라 하는가?
가. 형태 트리 나. 문장 트리
다. 파스 트리 라. 표현 트리

[04년8월][02년5월][99년10월]
5. 작성된 표현식이 BNF의 정의에 의해 바르게 작성되었는지를 확인하기 위해 만들어진 tree의 명칭은?
가. parse tree 나. binary search tree
다. binary tree 라. skewed tree

[P 2강]-프로그램 수행순서

[08년3월]
1. 변수(Variable)에 대한 설명으로 옳지 않은 것은?
가. 프로그램 실행 과정에서 하나의 기억 장소를 차지한다.
나. 변수의 유형은 컴파일 시간에 한번 정해지면 일반적으로 그대로 유지한다.
다. 프로그램이 동작하는 동안 절대로 값이 바뀌지 않는 공간을 의미한다.
라. 변수는 이름, 값, 속성, 참조의 요소로 구성된다.

[08년7월]
2. 수명 시간 동안 고정된 하나의 값과 이름을 가지며, 프로그램이 동작하는 동안 절대로 값이 바뀌지 않은 것을 의미하는 것은?
가. 변수 나. 상수 다. 포인터 라. 블록

[정답] 1.다 2.나 3.다 4.다

[08년7월]
3. 기계어에 대한 설명으로 옳지 않은 것은?
가. 2진수 0과 1만 사용하여 명령어와 데이터를 나타낸다.
나. 컴퓨터가 직접 이해할 수 있어 실행 속도가 빠르다.
다. 모든 기계에서 공통으로 사용 가능하여 호환성이 높다.
라. 전문적인 지식이 없으면 이해하기 힘들다.

[08년5월]
4. 기계어에 대한 설명으로 옳지 않은 것은?
가. 0 또는 1로만 구성되어 있다.
나. 컴퓨터가 이해하는 언어이다.
다. 프로그램 작성이 용이하다.
라. 처리 속도가 빠르다.

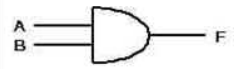
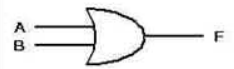
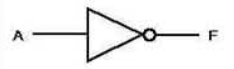
[변수]
: 프로그램 수행되는 과정에서 변할 수 있는 수

[상수]
: 프로그램 수행되는 과정에서 변하지 않는 고정된 수

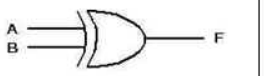
[P 3강]-연산자, 수식표기법, 자료형

1. 단항, 이항 연산자 ★★★★☆

1) 단항(Unary) : Not(Complement, 보수), MOVE, Shift 등
2) 이항(Binary) : AND, OR, XOR 등

게이트	기호	의미	진리표	논리식
AND	 (논리곱)	입력신호가 모두 1일 때 1 출력	A B F 0 0 0 0 1 0 1 0 0 1 1 1	F = A•B F = AB
OR	 (논리합)	입력신호 중 1개만 1이어도 1 출력	A B F 0 0 0 0 1 1 1 0 1 1 1 1	F = A + B
NOT		입력된 정보를 반대로 변환하여 출력	A F 0 1 1 0	F = A' F = $\overline{A}$

[P 3강]-연산자, 수식표기법, 자료형

게이트	기호	의미	진리표	논리식
XOR	 (exclusive-OR, 배타적 논리합)	입력되는 값이 모두 같으면 0, 한 개 라도 틀리면 1출력	A B F 0 0 0 0 1 1 1 0 1 1 1 0	F = A⊕B = A'B + AB'

2. 수식 표기법 변환 ★★★★☆

- PreFix (전위 표기법) : 연산자 → Left 피연산자 → Right 피연산자
- InFix (중위 표기법) : Left 피연산자 → 연산자 → Right 피연산자 (프로그래밍언어에서 가장 보편적으로 사용되는 표기법)
- PostFix (후위 표기법) : Left 피연산자 → Right 피연산자 → 연산자

* 피연산자 = 오퍼랜드

예)
- PreFix (전위 표기법) : + a b
- InFix (중위 표기법) : a + b
- PostFix (후위 표기법) : a b +

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 3강]-연산자, 수식표기법, 자료형

1) InFix → PostFix

[산술문] A / B ** C + D * E - A * C

① : BC**
② : A / ① → ABC**/
③ : DE*
④ : AC*
⑤ : ② + ③ → ABC**/DE*+
⑥ : ⑤ - ④ → ABC**/DE*+AC*-

2) InFix → PreFix

[산술문] A * B + C - D / E

① : *AB
② : /DE
③ : ① + C → +*ABC
④ : ③ - ② → -+*ABC/DE

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 3강]-연산자, 수식표기법, 자료형

주의) : 왼쪽에서 PreFix 표기인 연산자, 피연산자, 피연산자 구조를 찾는다.

3) PreFix → PostFix

[산술문] - / * A + B C D E

① : BC+
② : * A ① → ABC+*
③ : / ② D → ABC+*D/
④ : - ③ E → ABC+*D/E-

주의) : 왼쪽에서 PostFix 표기인 피연산자, 피연산자, 연산자 구조를 찾아서 괄호로 묶는다.

4) PostFix → InFix

[산술문] A B C - / D E F + * +

((A (B C -) /) (D (E F +) *) +)

① : (B-C)
② : (A ① /) → (A/(B-C))
③ : (E+F)
④ : (D ③ *) → (D*(E+F))
⑤ : (② ④ +) → ((A/(B-C))+(D*(E+F)))
→ 필요없는 괄호 없애기
: A/(B-C)+D*(E+F)

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 3강]-연산자, 수식표기법, 자료형

[05년3월][01년6월][07년5월]
1. 단항(unary) 연산에 해당하는 것은?
가. OR 나. AND
다. XOR 라. NOT

[04년8월][03년5월][02년5월][02년3월][01년3월][00년7월]
2. 이항(binary) 연산이 아닌 것은?
가. xor 나. or
다. and 라. complement

[07년8월][00년7월][04년8월][04년3월][03년8월][02년3월][01년6월][01년3월]
3. 프로그래밍언어에서 가장 보편적으로 사용되는 표기법은?
가. suffix 나. postfix
다. prefix 라. infix

[정답] 1.라 2.라 3.라 4.라 5.가 6.가 7.가

[07년8월]
4. 수학적 수식 "A+B*C-D"을 후위(Postfix) 표기법으로 표현한 것은?
가. A B C * D - + 나. A B + C * D -
다. A B C + * D - 라. A B C * + D -

[07년3월]
5. 다음 중위식(infix)을 후위식(postfix)으로 옳게 표현한 것은?
A - (D * K)
가. A D K * - 나. - A * D K
다. A D K - * 라. - * A D K

[04년8월][00년3월][03년8월]
6. 수식 " *+AB-CA " 에 사용된 표기법은?
가. Prefix 표기법 나. Postfix 표기법
다. Infix 표기법 라. Outfix 표기법

[05년5월]
7. A+(B*C) 를 PREFIX로 표현한 것은?
가. +A*BC 나. ABC*+
다. +*ABC 라. CBA*+

[P 3강]-연산자, 수식표기법, 자료형

3. 자료형 ★★★☆☆

1) 정수형, 실수형

	정수형 (고정소수점)	실수형 (부동소수점)
연산절차	간단	복잡 → 연산시간 많이 걸림 정규화 과정 필요
수 표현 범위		매우 크고, 작은 수 표현 가능

2) 부울형 : "TRUE" 혹은 "FALSE"라는 두 값 중에 하나를 나타내는 자료형
3) 포인터 : 다른 메모리 공간의 주소를 포함하고 있는 메모리 위치로 나타내는 자료 객체 (C언어 → 고급언어)
4) Array : 동일한 성격의 자료를 모아 놓은 것 (순차구조)

4. 자료객체 ★☆☆☆☆

: 파일, 변수, 상수 등 프로그램이나 시스템에서 정의한 것
- 변수 : 프로그램에서 하나의 값을 저장할 수 있는 기억 장소의 이름

5. 자료형 검사, 변환 ★☆☆☆☆

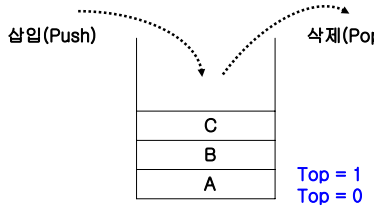
1) 동적 검사 : 번역 또는 실행시 자료형의 일관성을 동적으로 검사
- 프로그램 설계시 융통성을 준다. → 자료형 변경 가능 → 대화형언어 적합

2) widening (확장) : 정수형을 실수형으로 변환

[P 3강]-연산자, 수식표기법, 자료형

6. 자료저장구조 > 스택 (Stack) ★☆☆☆☆

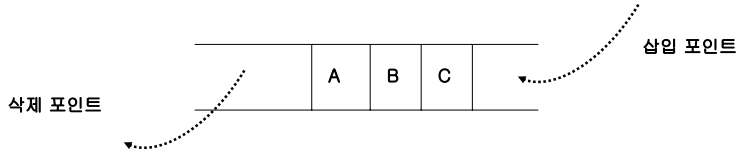
1) 삽입/삭제가 한 쪽에서 이루어지는 데이터 구조 (LIFO : Last In First Out) → 서브루틴 복귀번지 저장



* Top Point : 가장 최근에 삽입된 자료 또는 가장 먼저 삭제될 자료를 가리키는 스택 포인터
- 삽입 : Top 값 증가
- 삭제 : Top 값 감소

7. 자료저장구조 > 큐(Queue) ★☆☆☆☆

1) 노드의 삽입 작업은 선형 리스트의 한 쪽 끝에서, 제거 작업은 다른 쪽 끝에서 수행되는 자료 구조 (FIFO : First In First Out)



[P 3강]-연산자, 수식표기법, 자료형

[05년5월][04년3월]
1. 부동소수점(floating point) 연산에 대한 설명으로 옳지 않은 것은?
가. 고정소수점(fixed point) 연산에 비해 연산절차는 단순하다.
나. 매우 큰 수나 작은 수를 계산하기에 편리하다.
다. 고정소수점(fixed point) 연산에 비해 시간이 많이 걸린다.
라. 정규화(normalization) 과정이 필요하다.

[03년5월][02년8월]
2. 10진수 634를 BCD 코드로 표현한 것은?
가. 011000110100 나. 001100110100
다. 011000110011 라. 001100110011

[04년8월][02년3월][01년6월][99년8월]
3. Array 구조와 가장 밀접한 구조는?
가. 순차구조(Sequential structure)
나. 접속구조(Linked structure)
다. 리스트구조(List structure)
라. 환형접속구조(Circular linked structure)

[정답] 1.가 2.가 3.가 4.가 5.라

[07년5월][06년3월][04년8월][99년6월]
4. 포인터 자료형에 대한 설명으로 옳지 않은 것은?
가. 고급언어에서는 사용되지 않고 조급언어에서 주로 사용되는 기법이다.
나. 객체를 참조하기 위해 주소를 값으로 하는 형식이다.
다. 커다란 배열에 원소를 효율적으로 저장하고자 할 때 이용한다.
라. 하나의 자료에 동시에 많은 리스트의 연결이 가능하다.

[05년8월]
5. 동적(실행시간)형 검사에 대한 설명으로 옳지 않은 것은?
가. 프로그램 설계시 융통성을 준다.
나. 프로그램이 수행되는 과정내에 자료형을 변경할 수 있다.
다. 대화형언어에 적합하다.
라. 프로그램 수행 중에 형 정보를 유지할 필요가 없다.

[P 3강]-연산자, 수식표기법, 자료형

[05년3월]
6. 자료객체에 관한 설명으로 옳지 않은 것은?
가. 파일, 상수와 같이 프로그램이나 시스템에서 정의한 것이다.
나. 컴퓨터 저장소의 정적인 조직이다.
다. 프로그램 실행중에는 여러 형의 다른 객체가 존재한다.
라. 가상컴퓨터에서 실행시 하나 이상의 객체 묶음이다.

[00년10월]
7. 프로그래머가 프로그램 내에서 정의하고 이름을 줄 수 있는 자료 객체는?
가. 변수 나. 정수
다. 실수 라. 유리수

[07년3월][04년3월][02년8월][99년10월]
8. 요소 선택과 삭제는 한쪽에서, 삽입은 다른 쪽에서 일어나도록 제한하는 것은?
가. 큐 나. 스택
다. 트리 라. 방향 그래프

[04년5월][99년8월][02년5월]
9. 서브루틴 호출(subrutine call) 처리 작업시 복귀주소를 저장하고 조회하는 용도에 적합한 자료 구조는?
가. 데크 나. 큐
다. 스택 라. 연결리스트

[정답] 6.나 7.가 8.가 9.다

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

1. 구조화 프로그래밍(구조적 프로그램) 순서 제어

1) 구조화 프로그래밍 정의
: 컴퓨터 프로그램의 구조를 여러 갈래로 분기하여, 복잡하게 하지 않고, 순서대로, 선택적으로 반복 문장을 사용하는 제어구조만을 사용한 프로그램이다. → 이해하기 쉽다.

[1에서100까지 합계 구하기]

```
Sub main()

'변수 선언
Dim i, Hap As Integer

'100까지 합 구하기
For i = 1 To 100
    Hap = Hap + i
Next i

'합 출력하기
MsgBox Hap

End Sub
```

2) 구조화 프로그래밍 특징
- 프로그램의 이해가 쉽고 디버깅 작업이 쉽도록 한다.
- 한 개의 입구와 한 개의 출구 구조를 갖도록 한다.
- 계층적 설계를 한다.

* GOTO 문 사용 (X)
→ GOTO 문을 많이 사용하면 프로그램을 이해하기가 어렵다.

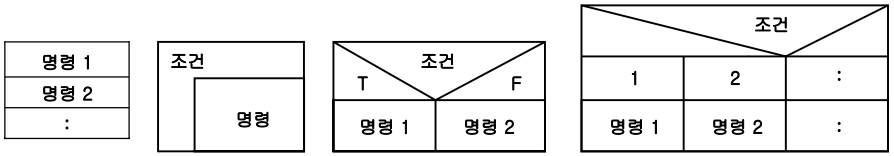
[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

3) 기본 순서 제어구조 (명령문의 순서를 제어하는 구조의 종류)

① 순차 구조 : 순차적으로 수행

② 반복 구조 : 조건을 만족할 때까지 반복 (FOR문)

③ 선택(조건, 다중 택일) 구조 : 두 가지 이상의 명령문 중에서 선택
- 두 가지의 수행 경로에 있는 일련의 문장들 중 하나가 선택 (IF 문)
- 두 가지 이상 중에서 선택 (Case 문, 계산형 GOTO 문, SWITCH 문)



순차 구조 반복 구조 선택 구조 다중 택일(Case) 구조

2. 일반적인 프로그래밍 순서제어 ★☆☆☆☆

① 묵시적 순서 제어 (ex. 사칙연산) : 프로그래머가 직접 제어 X
② 명시적 순서 제어 (ex. For 문) : 프로그래머가 직접 제어 O

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

[02년5월][01년9월][01년6월][00년3월]
1. 구조화 프로그램을 설계하기 위한 설명으로 옳지 않은 것은?
가. 프로그램의 이해가 쉽고 디버깅 작업이 쉽도록 한다.
나. 한 개의 입구와 한 개의 출구 구조를 갖도록 한다.
다. 실행시간의 단축을 위해 GOTO 문을 가급적 많이 사용한다.
라. 계층적 설계를 한다.

[07년8월][06년3월][03년5월][02년5월][02년8월][04년3월]
2. 구조적 프로그램의 기본 구조가 아닌 것은?
가. 순차(sequence)구조 나. 조건(condition)구조
다. 일괄(batch)구조 라. 반복(repetition)구조

[07년3월]
3. 구조적 프로그래밍과 거리가 먼 것은?
가. GOTO문 나. 순차실행문
다. 선택실행문 라. 반복실행문

[정답] 1.다 2.다 3.가 4.나 5.라 6.다

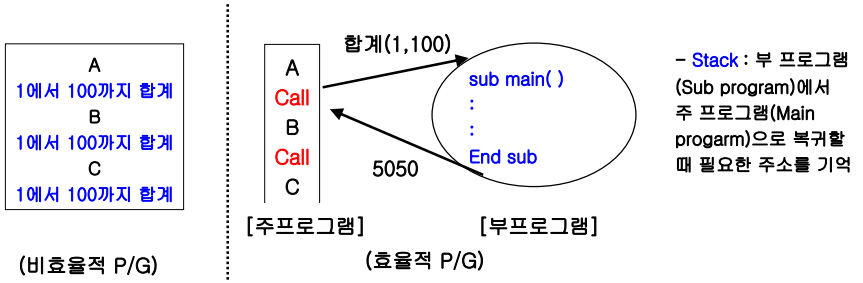
[07년5월][99년10월]
4. 구조화된(Structured) 순서 제어문과 가장 거리가 먼 것은?
가. IF문 나. GOTO문
다. CASE문 라. SWITCH문

[07년5월][99년6월][05년3월]
5. 일반적 프로그래밍 언어에서 다중 택일문에 해당하지 않는 것은?
가. 계산형 GOTO문 나. CASE문
다. SWITCH문 라. FOR문

[04년5월][03년3월][00년5월][99년4월][99년8월]
6. 프로그래머가 직접 제어를 표현하지 않았을 경우, 그 언어에서 미리 정해진 순서에 의해 제어가 이루어지는 순서제어는?
가. 구조적 나. 명시적
다. 묵시적 라. 문장 수준

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

3. 부프로그램 ★★★☆☆



(비효율적 P/G) [주프로그램] [부프로그램] (효율적 P/G)

- Stack : 부 프로그램(Sub program)에서 주 프로그램(Main progarm)으로 복귀할 때 필요한 주소를 기억

1) 부프로그램 선언 양식

합계 (1,100, integer)

이름 인자 = 매개변수 = 파라미터 (parameter) 유형

2) 매개변수 전달 방식
- call by value : 실제 값
- call by reference : 주소
- call by name : 이름

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

3) 특징
- 프로그램의 크기가 줄어든다 → 관리, 수정하기가 편리
- 프로그램의 처리 속도를 줄일 수 있다. (X)

4) 코루틴 : 두 모듈(부프로그램)이 같이 실행되면서 서로 호출하는 형태

5) 부프로그램(subprogram)과 매크로(macro) 비교
- 속도 : 부프로그램 < 매크로
- 프로그램 크기 : 부프로그램 < 매크로

6) 활성 레코드 : 프로그램 메인 루틴(주프로그램)의 수행시에 서브루틴(부프로그램)을 호출할 때 필요한 정보
① Parameter (파라미터, 매개변수)
② Local variable (지역,국부 변수) → 전역 변수 X
③ Return address (복귀,반환 주소)

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

[00년5월]
1. 부프로그램을 선언할 때 필요한 사항이 아닌 것은?
가. 부프로그램의 이름 나. 부프로그램의 존재
다. 부프로그램의 인자 라. 부프로그램의 위치

[00년5월][99년4월]
2. 주 프로그램의 매개변수(parameter)가 부 프로프램으로 넘어갈 때 실제 값이 전달되는 방식을 무엇이라 하는가?
가. call by value 나. call by reference
다. call by name 라. call by address

[정답] 1.라 2.가 3.다 4.다 5.라

[05년3월][00년5월][99년6월]
3. 프로그램 메인 루틴의 수행시에 서브루틴을 호출할 때는 활성 레코드(activation record)가 만들어진다. 이때 이 활성 레코드 안에 들어가는 정보가 아닌 것은?
가. 파라미터(parameter)
나. 국부 변수(local variable)
다. 실행 코드(execution code)
라. 복귀 주소(return address)

[99년4월]
4. 다음 중 활성 레코드(active record)를 구성하는 요소가 아닌 것은?
가. 지역 변수(Local variables)
나. 매개 변수(Parameters)
다. 전역 변수(Global variables)
라. 반환 주소(return address)

[04년3월][00년10월]
5. C 언어의 활성 레코드에 포함되는 사항이 아닌 것은?
가. 해당 함수의 지역변수 나. 반환 주소
다. 정적 링크 라. 전역 변수

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

4. 프로그램 언어의 유해한 특징 ★★☆☆☆

1) 부작용 현상 (Side Effect)

: 프로그램을 구성하는 함수에서 전역 변수를 사용하여 함수의 결과를 반환하는 경우, 함수에 전달되는 입력 파라미터의 값이 같아도 전역 변수의 상태에 따라 함수에서 반환되는 값이 달라질 수 있는 현상
→ 연산의 결과로 예상할 수 없을 정도로 다른 변수의 값이 변하는 경우를 의미한다.

2) 별명 (Alias)

: 하나의 기억장소(객체)에 둘 이상의 이름을 가질 수 있는 성질 (객체는 생존기간 중 여러 별명을 가질 수 있다.)
→ 일반적으로 별명은 프로그램의 이해를 매우 어렵게 한다.
→ 여러 가지 별명을 갖는 경우 프로그램의 무결점 검증이 어려워진다.
→ 같은 참조환경에서 다른 이름으로 같은 자료객체를 참조할 수 있는 언어의 경우 프로그래머에게 심각한 어려움을 줄 수 있다.

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

5. 바인딩 시간(Binding Time) ★★★★★

1) 바인딩 정의 : 어떤 변수의 명칭과 그 메모리 주소, 데이터형 또는 실제 값을 연결하는 것

2) 바인딩 시간 (Binding Time) 정의 : 프로그램에서 변수들이 갖는 속성이 완전히 결정되는 시간

3) 바인딩 시간의 종류

① 정적 바인딩 : 실행 시간 전에 바인딩이 일어나며, 실행 중에는 변하지 않음 (실행 전에 메모리 할당)
→ 명확하지만, 메모리 낭비 가능성 (효율성 우수) → 컴파일러 언어

- 번역 시간 : 원시 P/G → 목적 P/G
- 링크 시간 : 모듈 연결
- 언어정의 시간 : 프로그램의 자료구조, 택일문 등을 확정하는 바인딩 시간
- 언어구현 시간 : 언어를 컴퓨터 상에서 구현할 때 특성의 일부를 확정하는 바인딩 (ex. 정수 자릿수)

② 동적 바인딩 : 실행 시간 중에 바인딩이 일어나며, 실행 중에 변경 가능 (실행 중에 메모리 할당)
→ 메모리 낭비 없음 (융통성 우수) → 인터프리터 언어
- 프로그램 호출 시간
- 모듈의 기동 시간
- 실행시간 (실행시간 중 객체 사용시점)

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

[04년 5월][03년3월][00년7월][01년9월][06년5월][99년4월]
1. 프로그램을 구성하는 함수에서 전역 변수를 사용하여 함수의 결과를 반환하는 경우, 함수에 전달되는 입력 파라미터의 값이 같아도 전역 변수의 상태에 따라 함수에서 반환되는 값이 달라질 수 있는 현상을 무엇이라 하는가?
가. reference 나. side effect
다. aliasing 라. recursive

[06년3월][04년8월][04년3월][00년10월]
2. 자료 객체의 별명(alias)에 관한 설명으로 옳지 않은 것은?
가. 자료 객체는 생존기간 중 여러 별명을 가질 수 있다.
나. 일반적으로 별명은 프로그램의 이해를 매우 어렵게 한다.
다. 자료 객체가 여러 가지 별명을 갖는 경우 프로그램의 무결점 검증이 쉬워진다.
라. 같은 참조환경에서 다른 이름으로 같은 자료객체를 참조할 수 있는 언어의 경우 프로그래머에게 심각한 어려움을 줄 수 있다.
[정답] 1.나 2.다 3.나 4.라 5.다

[07년5월][04년5월][03년3월][99년8월]
3. 프로그램에서 변수들이 갖는 속성이 완전히 결정되는 시간을 무엇이라 하는가?
가. 컴파일 시간(Compile Time)
나. 바인딩 시간(Binding Time)
다. 실행 시간(Run Time)
라. 로드 시간(Load Time)

[05년3월][99년10월][03년8월][02년3월]
4. 동적바인딩(Dynamic Binding)이 이루어지는 시간이 아닌것은?
가. 프로그램 호출 시간
나. 모듈의 기동 시간
다. 실행시간 중 객체 사용시점
라. 번역 시간

[07년8월][01년3월][00년7월][01년6월][02년8월][04년3월][05년3월][05년5월][03년5월][02년5월]
5. 정적바인딩에 해당하지 않는 것은?
가. 번역시간 나. 링크시간
다. 실행시간 라. 언어구현시간

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

6. 형식 문법 ★★☆☆☆

1) 정의 : 유한개의 규칙을 통해 어떤 문자열이 특정 언어에 포함되는지를 판단하거나, 그 문법으로부터 어떤 문자열을 생성해 낼지를 정한다.

2) 형식 문법 계층

① Type 0 : 형식에 제한이 없는 문법
- 인식기 : 튜링 기계 (Turing Machine)

② Type 1 : 복잡해서 프로그래밍 언어에 적용하지 않는 문법
- 인식기 : 선형 한계 오토마타 (Linear Bounded Automata)

③ Type 2 : Context-free 문법
- 인식기 : 스택 자동기계 (Push Down Automata)

④ Type 3 : 어휘구조(lexical-structure)를 표현하는데 사용하는 문법 (정규 표현, 정규 문법)
- 인식기 : 유한 오토마타 (Finite Automata) → 이산적인 입력과 출력에 유한 수의 내부상태를 가진 시스템의 수학적 모델

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

7. 정규 표현 (정규 언어) ★☆☆☆☆

1) 정의 : 정규 문법에 의해 생성된 언어

2) 특징
- 정규 표현은 정규 언어를 나타내는 수식이다.
- 정규 표현은 스트링 길이에 제한이 없다.
- 정규 표현은 상태 전이도로 나타낼 수 있다.
- 정규 집합(정규 표현을 위한 기호 집합)을 형성하는 기초가 된다.

8. 구역성 ★☆☆☆☆

1) 정의 : 프로세스가 실행되는 동안 일부 페이지만 집중적으로 참조하는 성질

① 시간 구역성 : 최근에 참조된 기억 장소가 가까운 장래에도 계속 참조될 가능성이 높음
예) 순환(looping), 부프로그램(subprogram), 집계(totaling) 등에 사용되는 변수

② 공간 구역성 : 하나의 기억 장소가 참조되면 그 근처의 기억 장소가 계속 참조될 가능성이 높음

[P 4강]-구조화프로그래밍, 부프로그램, 바인딩

[03년8월][00년10월][01년9월][99년8월]
1. 컴퓨터 프로그래밍 언어의 어휘구조(lexical-structure)를 표현하는데 사용하는 문법의 종류는?
가. Type 0 문법 나. Type 1 문법
다. Type 2 문법 라. Type 3 문법

[01년6월][02년3월][02년5월]
2. 이산적인 입력과 출력에 유한 수의 내부상태를 가진 시스템의 수학적 모델을 무엇이라 하는가?
가. 유한 오토마타 나. 정규문법
나. 정규언어 라. 컴파일러

[03년8월][00년3월][04년3월]
3. Context-free 문법으로 표현된 언어를 인식하는데 사용되는 automata는?
가. 유한 오토마타(Finite Automata)
나. 스택 자동기계(Push Down Automata)
다. 튜링 기계(Turing Machine)
라. 선형 한계 오토마타(Linear Bounded Automata)

[정답] 1.라 2.가 3.나 4.가 5.나 6.가

[05년3월][02년8월][00년5월]
4.정규표현(Regular Expression)을 받아들이는 효율적인 오토마타(automata)는?
가. 유한 상태 오토마타
나. 푸쉬다운 오토마타
다. 튜링 머쉰
라. 선형 제한 오토마타

[06년3월][00년7월][00년10월]
5. 정규표현(redular expression)에 대한 설명으로 옳지 않은 것은?
가. 정규 표현은 정규 언어를 나타내는 수식이다.
나. 정규 표현은 유한 길이의 스트링만 나타낼 수 있다.
다. 정규 표현은 상태 전이도로 나타낼 수 있다.
라. 정규 집합을 형성하는 기초가 된다.

[05년5월][02년5월]
6. 시간 구역성의 예가 아닌 것은?
가. 배열 순례(array traversal)
나. 순환(looping)
다. 부프로그램(subprogram)
라. 집계(totaling) 등에 사용되는 변수

[P 5강]-C언어

1. 기본구조

```
main()
{
    int A, B, C;
    A=1, B=2;
    C=A+B;
    printf("%d", C);
}
```

2. 특징 ★★★☆☆

- 이식성, 효율성이 높은 언어 → 시스템 프로그래밍 언어
- 구조적 프로그래밍이 가능
- 고급언어이면서 저급언어 프로그래밍도 가능
- 자료의 주소를 조작할 수 있는 포인터를 제공
- 컴파일러 방식의 언어
- 항상 main( )이라는 함수로부터 실행이 시작된다.
- 주석문은 컴파일러에 의해 번역되지 않는다.
- 영문자의 대문자와 소문자를 구별
- 문장을 끝마칠때 " ; " 이 필요하다.
- 숫자는 식별자(변수명)의 첫 번째 문자가 될 수 없다.
→ ex) 135 안됨

3. 데이터 유형 ★★★★★

의미	데이터 유형	크기(byte)
정수형	int	2
	long	4
실수형	float	4
	double	8
문자형	char	1

* 주의사항 :
integer, character 틀린 표기

[P 5강]-C언어

[05년3월][02년5월][01년6월][00년5월]
1. C 언어에 대한 설명으로 옳지 않은 것은?
가. 구조적 프로그래밍이 가능하다.
나. 시스템 소프트웨어를 작성하기에 편리하다.
다. 기계어에 해당한다.
라. 이식성이 높은 언어이다.

[04년8월][01년9월][03년8월]
2. C 언어에서 기본 자료 형에 해당되지 않는 것은?
가. 배열형(array) 나. 정수형(int)
다. 실수형(float) 라. 문자형(char)

[06년8월]
3. C언어에서 정수형 변수를 선언할 때 사용하는 자료형은?
가. char 나. int
다. float 라. double

[06년3월][99년10월][05년3월][06년5월][01년3월][02년8월][01년9월][03년3월][02년5월][02년3월][00년7월][00년5월][07년5월][07년8월][00년10월]
4. C언어의 자료형이 아닌 것은?
가. long 나. integer
다. float 라. double

[05년5월]
5. 다음의 C언어 데이터 유형 가운데 가장 메모리를 많이 차지하는 것은?
가. char 나. int
다. long 라. double

[05년8월]
6. C 언어에서 선언하는 자료형이 아닌 것은?
가. float 나. double
다. int 라. character

[정답] 1.다 2.가 3.나 4.나 5.라 6.라

[P 5강]-C언어

4. 기억 클래스 ★★★☆☆

1) 정의 : 변수는 데이타 유형 이외에 기억 클래스(storage class)라는 것이 있으며, 변수가 데이터 저장 장로로 메모리와 CPU의 레지스터 중 어느것에 기억되는 가를 결정하고, 변수의 유효 범위를 결정하는 것

2) 기억 클래스의 종류 (변수의 기억 장소와 기억 방식에 따른 분류)

① 자동 변수 (automatic variable)
- ex) auto int a;
- 자동 변수는 필요치 않을 때는 기억장소를 전혀차지 않으며, 어떤 함수에만 국한된 지역(local) 변수이기 때문에 다른 함수의 값을 변경시킬 수 없음.
- 저장 클래스를 명시하지 않은 변수는 기본적으로 auto로 인식됨

② 정적 변수 (static variable)
- 지역 변수와 유사한 역할

③ 외부 변수 (external variable)
- 전역 변수와 유사한 역할

④ 레지스터 변수 (register variable)

[P 5강]-C언어

5. 서술자 ★★☆☆☆

1) 정의 : C 언어의 데이터 형식을 규정

서술자	기 능
%o	octal : 8진수 정수
%d	decimal : 10진수 정수
%x	hexadecimal : 16진수 정수
%c	character : 문자
%s	string : 문자열

ex) printf("%-7d" , a)
→ 정수형 변수 a에 256이 저장되어 있을 경우, 7자리로 잡아 왼쪽으로 붙여 출력하므로 (- 가 없으면 오른쪽)

2 5 6 _ _ _ _

[P 5강]-C언어

[04년8월][02년8월][01년3월][06년5월]
1. C 언어의 기억 클래스에 해당하지 않는 것은?
가. 내부 변수(internal variable)
나. 자동 변수(automatic variable)
다. 레지스터 변수(register variable)
라. 정적 변수(static variable)

[05년8월][04년5월][00년3월]
2. C언어에서 저장클래스를 명시하지 않은 변수는 기본적으로 어떤 변수로 간주되는가?
가. global 나. extern
다. auto 라. local

[03년5월][02년3월][01년6월][00년5월][99년6월]
3. C 언어에서 사용하는 기억클래스에 해당하지 않는 것은?
가. auto 나. static
다. register 라. scope

[06년8월][05년5월][05년3월][01년3월][03년5월]
4. C 언어의 출력 문에서 데이터 형식을 규정하는 서술자로서 의미가 옳지 않은 것은?
가. %d : 8진 정수 나. %c : 문자
다. %s : 문자열 라. %x : 16진 정수

[01년9월][02년8월][99년10월]
5. C 언어에서 정수형 변수 a에 256이 저장되어 있다. 이를 7자리로 잡아 왼쪽으로 붙여 출력하려고 할 때, 적절한 printf() 내의 % 변화문자 사용은?
가. %7f 나. %7d
다. %-7d 라. %-7i

[05년8월]
6. C언어에서 16진 정수를 출력하기 위한 변환 문자의 사용으로 옳은 것은?
가. %x 나. %d
다. %s 라. %h

[정답] 1.가 2.다 3.라 4.가 5.다 6.가

[P 5강]-C언어

6. escape 문자 (이스케이프 시퀀스) ★☆☆☆☆

: 인쇄할 수 없거나 키보드로 표현할 수 없는 특별한 문자를 가리키며, 역슬러쉬(\ )와 한 개의 문자와 결합하여 작성한다.

- 주의사항 : ₩ 는 역슬러쉬(\ ) 이다.

escape 문자	기 능
₩n	new line : 커서를 다음 줄로 바꾼다.
₩r	carriage return : 커서를 그 줄의 맨 앞으로 이동
₩f	form feed : 한 페이지를 넘긴다.
₩b	backspace : 커서를 그 줄의 1문자만큼 앞으로 이동
₩t	tab : 커서를 그 줄의 tab 만큼 이동

[P 5강]-C언어

7. 블록 구조 ★☆☆☆☆

1) 블록 정의 : { } 로 묶인 부분 → 왜? 프로그램 구성을 단계적으로 세분화

2) 특징
- 변수를 사용할 프로그램의 문장 근처에서 선언하도록 하기 때문에 프로그램의 지역성(locality)을 높여준다.
- 프로그램의 변수명과 삽입되는 라이브러리 루틴의 변수명이 같더라도 문제점이 없게 된다.
- C언어는 함수 또는 프로시저와 같은 블록의 영역을 정의하는 기본 단위로 사용
- 프로그램의 블록 내포 관계를 기준으로 지역변수와 비지역 변수의 영역을 정의한다.
- 프로그램에서 사용하는 식별자 또는 변수의 자료형을 명시적으로 선언하여야 한다.

[P 5강]-C언어

[06년8월][03년8월][02년3월][01년3월]
1. C언어에서 이스케이프 시퀀스의 설명이 옳지 않은 것은?
가. ₩n : null character
나. ₩r : carriage return
다. ₩f : form feed
라. ₩b : backspace

[03년3월][04년3월][00년7월]
2. C 언어에서 사용되는 이스케이프 시퀀스(escape - sequence)와 그 의미의 연결이 옳지 않은 것은?
가. ₩n : new line 나. ₩b : null character
다. ₩t : tab 라. ₩r : carriage return

[05년3월][03년8월]
3. 블록 구조에 의한 영역 개념을 사용함으로써 얻어지는 장점으로 거리가 먼 것은?
가. 변수를 사용할 프로그램의 문장 근처에서 선언하도록 하기 때문에 프로그램의 지역성(locality)을 높여준다.
나. 프로그램 문장과 변수들의 지역성은 필요로 하는 기억장소의 크기를 작게 만들게 되며, 이는 운영체제의 working set을 크게 하는 장점이 있다.
다. 프로그램의 변수명과 삽입되는 라이브러리 루틴의 변수명이 같더라도 문제점이 없게 된다.
라. 프로그램의 구성을 단계적으로 세분화하는데 도움을 준다.

[정답] 1.가 2.나 3.나

[P 5강]-C언어

8. 연산자 → 연산 우선 순위 ★★★★★

1) 증가/감소 연산자

기호	예	의미
++	++A	A 를 1 증가 시킨 후 사용
--	--A	A 를 1 감소 시킨 후 사용

2) 산술 연산자

- 사칙 연산자 : *, /, +, -
- 나머지 연산자 : % → ex) y = a%b

3) 시프트 연산자

기호	예	의미
<<	A << B	A 를 B 비트만큼 왼쪽 shift
>>	A >> B	A 를 B 비트만큼 오른쪽 shift

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 5강]-C언어

4) 관계 연산자

기호	예	의미
>	A > B	A가 B보다 크다.
>=	A >= B	A가 B보다 크거나 같다.
<	A < B	A가 B보다 작다.
<=	A <= B	A가 B보다 작거나 같다.
==	A == B	A와 B는 같다.
!=	A != B	A와 B는 같지 않다.

5) 비트 단위 논리 연산자

기호	예	의미
&	A & B	A 와 B 를 비트 단위로 논리곱 → AND
|	A | B	A 와 B 를 비트 단위로 논리합 → OR
^	A ^ B	A 와 B 를 비트 단위로 배타적 논리합 → XOR
~	~ A	A 를 1 보수화

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 5강]-C언어

6) 할당 연산자

기호	예	의미
+=	A += B	A = A + B
-=	A -= B	A = A - B

* 기타 연산자

- 조건 연산자 : ?
- 논리 연산자 : &&, ||, !
- 대입 연산자 : =
- sizeof 연산자 : 기억장소 크기 반환
- 캐스트 연산자 : 어떤 수식을 다른 데이터 형으로 바꾸고 싶을 때 사용하는 연산자

사무자동화산업기사 필기 기사퍼스트(www.gisafirst.com)

[P 5강]-C언어

[07년8월][05년5월][05년8월][07년5월]
1. C언어에서 사용되는 관계 연산자 중 "A와 B가 같지 않다."의 의미를 갖는 것은?
가. A => B 나. A != B
다. A <= B 라. A & B

[04년5월][02년3월][00년5월][01년9월]
2. C 언어에서 비트 단위 논리 연산자의 종류에 해당되지 않는 것은?
가. ^ 나. ~
다. & 라. ?

[05년5월][01년3월]
3. C 언어에서 연산자 우선순위가 옳은 것은? (단, 오른쪽마지막 연산자가 가장 높은 우선순위를 가짐.)
가. +=, &, ==, <<, +, *, ++
나. +=, <<, &, ==, +, *, ++
다. +=, ==, &, <<, +, *, ++
라. +=, &, ==, +, *, <<, ++

[정답] 1.나 2.라 3.가 4.라 5.다 6.라 7.나

[03년3월][01년9월][01년6월][00년5월]
4. 다음의 C 언어 연산자 기호 중에서 우선순위가 가장 먼저인 것은?
가. && 나. ||
다. = 라. /

[05년8월][99년10월][03년8월]
5. C 언어에서 연산문의 표현이 옳지 않은 것은?
가. y=a%b 나. y+=a
다. y=a**2 라. y<<2

[05년8월][99년8월]
6. C 언어에서 어떤 수식을 다른 데이터 형으로 바꾸고 싶을 때 사용하는 연산자는?
가. 산술 연산자 나. 관계 연산자
다. 논리 연산자 라. 캐스트 연산자

[05년3월][00년3월]
7. C 언어의 연산자 중에서 오른쪽에서 왼쪽으로의 결합법칙을 따르지 않는 것은?
가. sizeof 나. <<
다. ! 라. ++

[P 5강]-C언어

9. 제어구조 (문장의 실행순서를 제어) ★★☆☆☆

1) 선택문
- if ~ else 문
- switch ~ case 문

2) 반복문
- for 문
- while 문 : 조건 검사 후 반복 실행 (실행 안될 수 있음)
- do ~ while 문 : 문장을 실행한 다음, 조건을 검사하여 반복 실행의 여부를 결정 (최소 한 번은 실행)

```
sum=0; i=1;
while(sum<20)
  { sum = sum+i; i=i+1; }
```

*수행 횟수 : 6 회

3) 강제 이동
- break 문
- continue 문
- goto 문

[P 5강]-C언어

[04년3월][99년10월]
1. C언어에서 문장의 실행순서를 제어하는 제어구조에 해당하는 문장으로 볼 수 없는 것은 ?
가. for 나. while
다. if 라. printf

[05년3월]
2. C 언어의 제어문 중 성격이 다른 것은?
가. break문 나. continue문
다. goto문 라. switch문

[07년8월][99년10월]
3. C 언어의 do ~ while 문에 대한 설명 중 틀린 것은?
가. 문의 조건이 거짓인 동안 루프처리를 반복한다.
나. 문의 조건이 처음부터 거짓일 때도 문을 최소 한번은 실행 한다.
다. 무조건 한 번은 실행하고 경우에 따라서는 여러 번 실행하는 처리에 사용하면 유용하다.
라. 맨 마지막에 ";"이 필요하다.

[05년8월]
4. C언어에서 사용되는 예약어가 아닌 것은?
가. case 나. switch
다. virtual 라. enum

[06년3월][00년3월]
5. 랜덤편성(Random organization)에 관한 설명으로 옳지 않은 것은?
가. 기억공간에 공백이 많아 효율적으로 사용하지 못한다.
나. 어떤 레코드에도 빠르게 액세스(access)하여 검색이 가능하다.
다. 키(key)변환이 운영체제(Operating System)에 의해 이루어 진다.
라. 입출력 매체의 종류에 영향을 받지 않는다.

[정답] 1.라 2.라 3.가 4.다 5.라

[P 5강]-C언어

[08년7월]
1. 사무자동화 자료처리를 위한 순차 파일의 장점으로 옳지 않은 것은?
가. 일괄처리 중심의 업무처리에 적합하다.
나. 파일 내에 필요 없는 레코드 삭제가 용이하다.
다. 어떤 매체라도 순차 편성 파일의 기록 매체가 될 수 있다.
라. 순차적으로 실제 데이터만 저장되므로 기억공간의 활용이 높다.

[정답] 1.나

* 순차 파일 : 목차 없는 책, 자기테이프
: 파일 내의 각 레코드를 논리적 순서에 따라 물리적으로 연속된 위치에 기록한 파일
- 기억장소의 낭비가 없다.
- 삽입, 삭제, 검색이 어렵다.

* 색인 순차 파일 : 목차 있는 책, 자기디스크
- 삽입, 삭제, 갱신, 검색 용이