



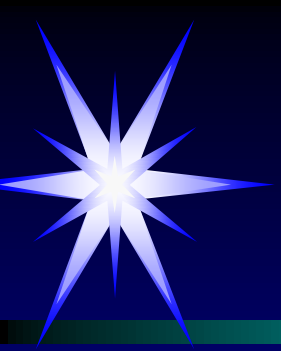
# 홈페이지 강의자료 다운로드

---

교수홈페이지

컴파일러 구성 학습정보 게시판

<http://cs.knou.ac.kr/~khkim/data3.asp>



# 컴파일러 관련과목

- ◆ 컴퓨터 프로그램 언어

- ◆ Visual BASIC, C, C++, Java, Delphi, .NET, FORTRAN, COBOL, PASCAL,

- ◆ 프로그래밍언어론

- ◆ 이산수학

시스템프로그래밍



# 컴파일러란?

언어를 번역한다.

번역기 프로그램

고급언어  
C, C++ JAVA



저급언어  
기계어  
010111000000



# 과목 개설 이유

◆컴파일러 구현의 어려움 -정확성

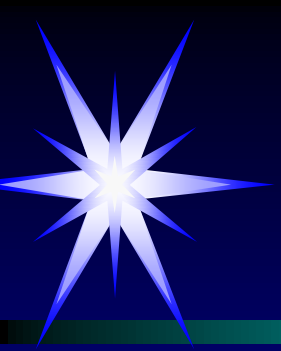
◆FORTRAN 컴파일러구현

→ 18 年人 (1957 backus)

◆발전과정

- 연산자 우선순위문법
- bottom up : LR(K) 소프트웨어
- top down : LL(K)

LEX,  
YACC



# 번역기의 종류

원시프로그램 → **번역기** → 목적프로그램

어셈블리어 → **어셈블러** → 기계어

고급언어 → **컴파일러** → 저급언어

고급언어 → **인터프리터** → 실행결과

고급언어 → **프리프로세서** → 고급언어

**C, C++, JAVA**

고급언어 high level 인간중심

저급언어 low level 기계중심

01010 11111

Load A 연산레지스터

어셈블리어 → 어셈블러 → 기계어

고급언어 → 컴파일러 → 저급언어

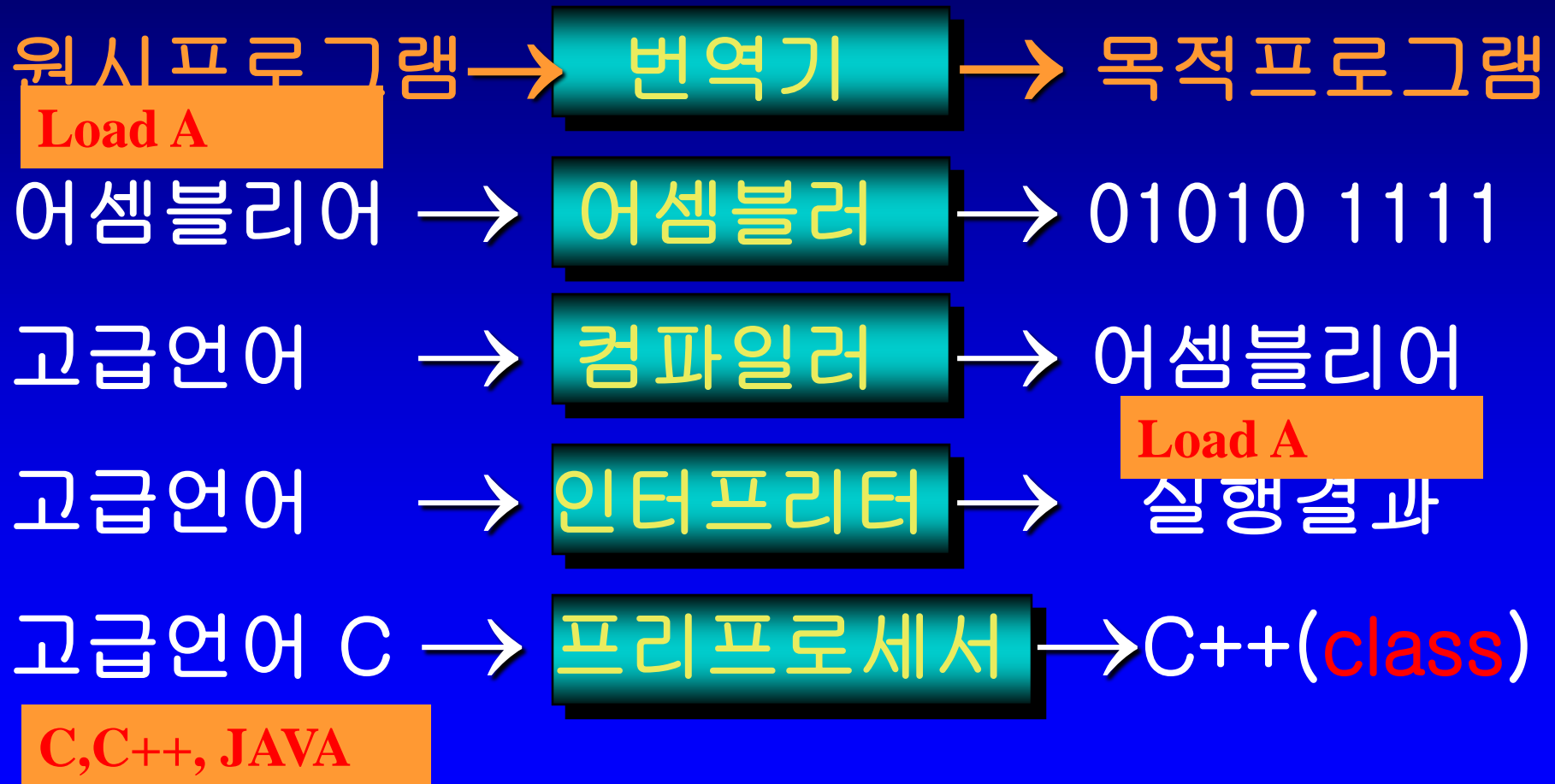
고급언어 → 인터프리터 → 실행결과


고급언어 → 프리프로세서 → 고급언어

C, C++, JAVA



# 번역기의 종류





```
while (time < tlimit)
{
    time = time + tstep;
    arrive = 0;
    random(&seed, &u);
    if (u < (prarr*tstep))
    { arrive = 1;
      queue = queue + arrive;
      totarr = totarr + 1.0;
    }
    if (tpump > 0.0)
    { tpump = tpump - tstep;
      if (tpump < 0.0) tpump = 0.0;
    }
    if ((tpump == 0) && (queue != 0))
    { queue = queue - 1;
      poissn(&nseed, &p);
      tpump = p;
    }
}
```




$$ABC := E * 3.14 + ABC / E;$$

100	LOAD	201
101	MULT	150
102	STORE	160
103	LOAD	200
104	DIV	201
105	ADD	160
106	STORE	200
:		

150	3.14
:	
160	(E*3.14)
161	
:	
200	ABC
201	E



**ABC := E \* 3.14 + ABC / E;**

**E**

<b>100</b>	<b>LOAD</b>	<b>201</b>
<b>101</b>	<b>MULT</b>	<b>150</b>
<b>102</b>	<b>STORE</b>	<b>160</b>
<b>103</b>	<b>LOAD</b>	<b>200</b>
<b>104</b>	<b>DIV</b>	<b>201</b>
<b>105</b>	<b>ADD</b>	<b>160</b>
<b>106</b>	<b>STORE</b>	<b>200</b>

**:**

<b>150</b>	<b>3.14</b>
------------	-------------

**:**

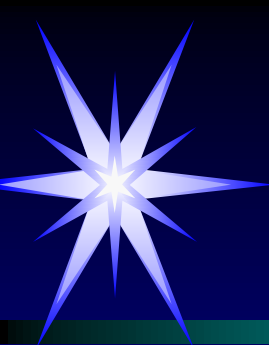
<b>160</b>
------------

<b>161</b>
------------

**:**

<b>200</b>	<b>ABC</b>
------------	------------

<b>201</b>	<b>E</b>
------------	----------


$$ABC := E * 3.14 + ABC / E;$$
$$E * 3.14$$

100	LOAD	201
-----	------	-----

101	MULT	150
-----	------	-----

102	STORE	160
-----	-------	-----

103	LOAD	200
-----	------	-----

104	DIV	201
-----	-----	-----

105	ADD	160
-----	-----	-----

106	STORE	200
-----	-------	-----

:
---

150	3.14
-----	------

:
---

160
-----

161
-----

:
---

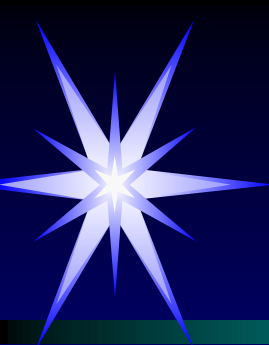
200	ABC
-----	-----

201	E
-----	---


$$ABC := E * 3.14 + ABC / E;$$
$$E * 3.14$$

100	LOAD	201
101	MULT	150
102	STORE	160
103	LOAD	200
104	DIV	201
105	ADD	160
106	STORE	200
:		

150	3.14
:	
160	(E*3.14)
161	
:	
200	ABC
201	E


$$ABC := E * 3.14 + ABC / E;$$

ABC

100	LOAD	201
101	MULT	150
102	STORE	160
103	LOAD	200
104	DIV	201
105	ADD	160
106	STORE	200

:

150	3.14
:	
160	(E*3.14)
161	
:	
200	ABC
201	E

**ABC := E \* 3.14 + ABC / E;**

**ABC/E**

100	LOAD	201
101	MULT	150
102	STORE	160
103	LOAD	200
104	DIV	201
105	ADD	160
106	STORE	200
:		

150	3.14
:	
160	(E*3.14)
161	
:	
200	ABC
201	E

**ABC := E \* 3.14 + ABC / E;**

**ABC/E + (E\*3.14)**

100	LOAD	201
101	MULT	150
102	STORE	160
103	LOAD	200
104	DIV	201
105	ADD	160
106	STORE	200
:		

150	3.14
:	
160	(E*3.14)
161	
:	
200	ABC
201	E


$$ABC := E * 3.14 + ABC / E;$$
$$ABC/E + (E*3.14)$$

100	LOAD	201
101	MULT	150
102	STORE	160
103	LOAD	200
104	DIV	201
105	ADD	160
106	STORE	200
:		

150	3.14
:	
160	(E*3.14)
161	
:	
200	ABC/E + (E*3.14)
201	E





# 컴파일러의 논리적 구조

◆ 프로그램

$ABC := E * 3.14 + ABC / E;$

◆ 어휘분석

ABC := E \* 3.14 + ABC / E ;

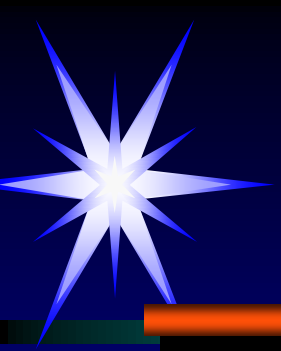
◆ 구문분석

식별자:=식별자\*숫자+식별자/식별자;

◆ 의미분석

기억장소 ← 곱셈+나눗셈 결과

◆ 목적코드 생성



# 논리적 구조 6단계

◆ 프로그램

$ABC := E * 3.14 + ABC / E;$

◆ 어휘분석

◆ 구문분석

◆ 의미분석

◆ 목적코드 생성



# 논리적 구조 6단계

◆ 프로그램

$ABC := E * 3.14 + ABC / E;$

◆ 어휘분석

◆ 구문분석

◆ 의미분석

◆ 중간코드 : 최적화를 위한 코드

◆ 최적화 : 수행시간, 기억공간 최소화

◆ 목적코드 생성



# 컴파일러 교재구성

고급언어 → 컴파일러 → 저급언어

어휘분석	2,3	LEX
구문분석	4,5	YACC
의미분석	6	
중간코드	7	
최적화	8	
코드생성	9	



# 컴파일러 교재구성

---

- ◆ 2장 : 형식언어와 오토마타(출석수업)
- ◆ 3장 : 어휘분석 (출석수업)
- ◆ 4장 : Context-free 언어와  
푸시다운 오토마타
- ◆ 5장 : 구문분석