3. different c tokens with example

sol

1. Keywords
Ex : int, while

2. Identifiers
ex: main, total

3. Constants
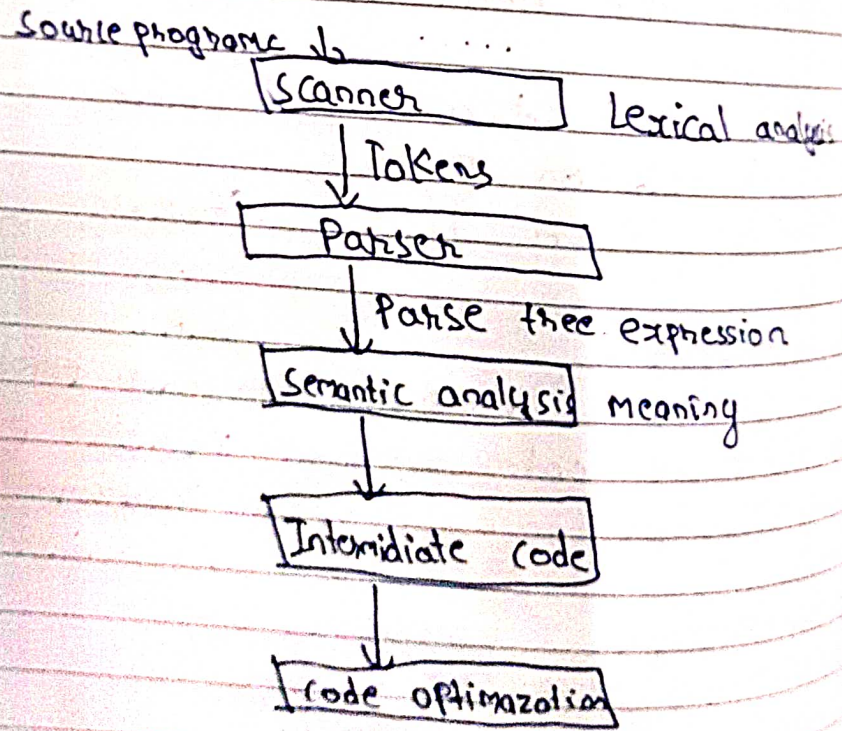Ex : 10, 20

4. strings |ex: "total", "hello")
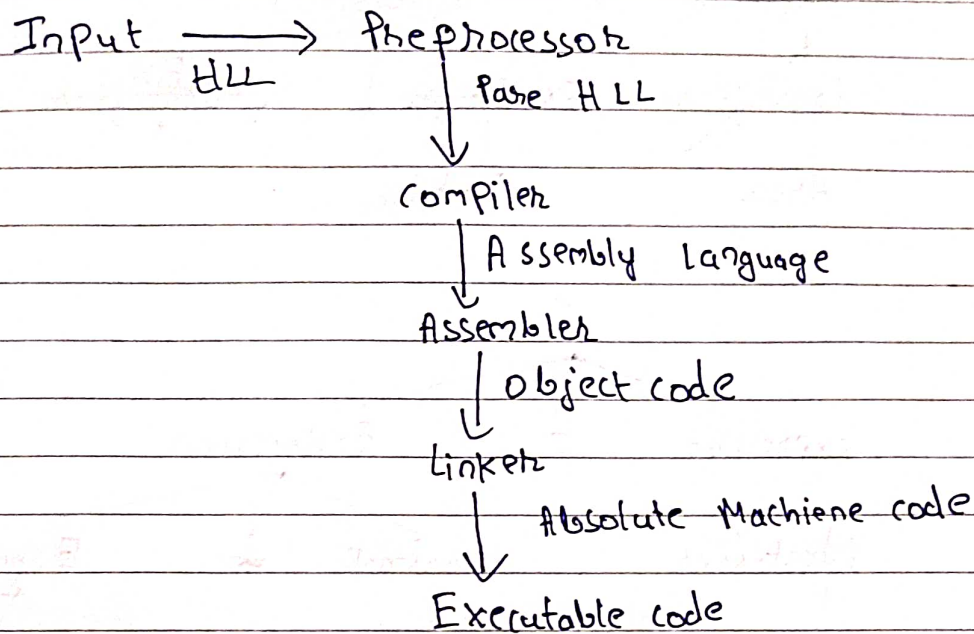5. special symbols (Ex: (), {})
6. Operators (Ex: +, /, -, &)

4. diagram of phases of compiler

Source programe

Scanner — Lexical analysis
↓ Tokens
Parser
↓ Parse tree expression
Semantic analysis meaning
↓
Intermidiate code
↓
code optimazation

5. language processing system

Input $\xrightarrow[\text{HLL}]{}$ Preprocessor

Pure HLL

$\downarrow$

Compiler

Assembly language

$\downarrow$

Assembler

object code

$\downarrow$

Linker

Absolute Machiene code

$\downarrow$

Executable code

area of circle

area = Pi * r * r

$\downarrow$

| Scanner |

$\downarrow$

lexeme	token
area	Identifiers
=	operator
Pi	Identifier
*	operator
r	Identifier
*	operator
r	Identifier

$\downarrow$

| Parser |

Parse tree expression

S

expression = expression

identifier   EX identifier *   EX identifier

Area   identifier

S

Expression = Expression

identifier   constant *   Expression

Area   Pi   Expr   Exp

                 3.142   identifier   id

                                  r   r

semantic analysis

=

area   *

Float   3.142   *

                 t2   t2

            int to real   int to real

Intermidiate code

$t_1 = r$

$t_2 = t_1 * t_1$

$t_3 = Pi * t_2$

$area = t_3$

Code Optimization

$t_1 = r$

$t_2 = t_1 * t_1$

$Area = Pi * t_2$

MOV   R₁ r → MOV   R$_1$ r

MOV R_1 r
MUL R_1 R_1
MOV R_2 Pi
MUL R_1 R_2
MOV   Area R_1

Tutorial 2

Analyse the phases of instruction

$$area = 2 + Pi * r * r$$

↓

| scaner |

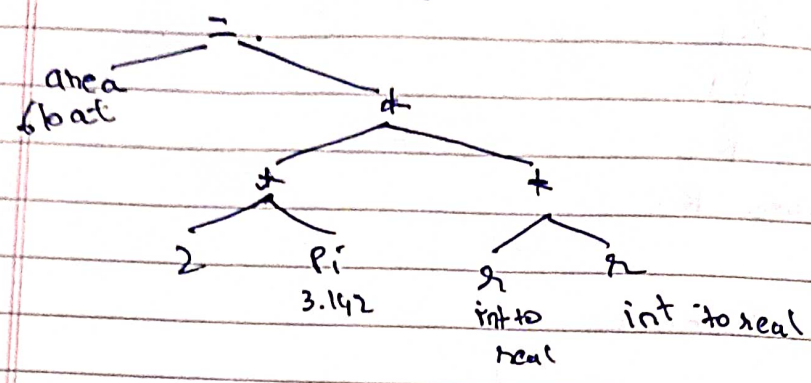| lexeme	token
area	Identifiers operator
2	Constants operator
Pi	Identifiers constant
r*	Operator
&r	Identifier
*	operator
r	Identifiers

↓

| Parser |

Parse tree expression
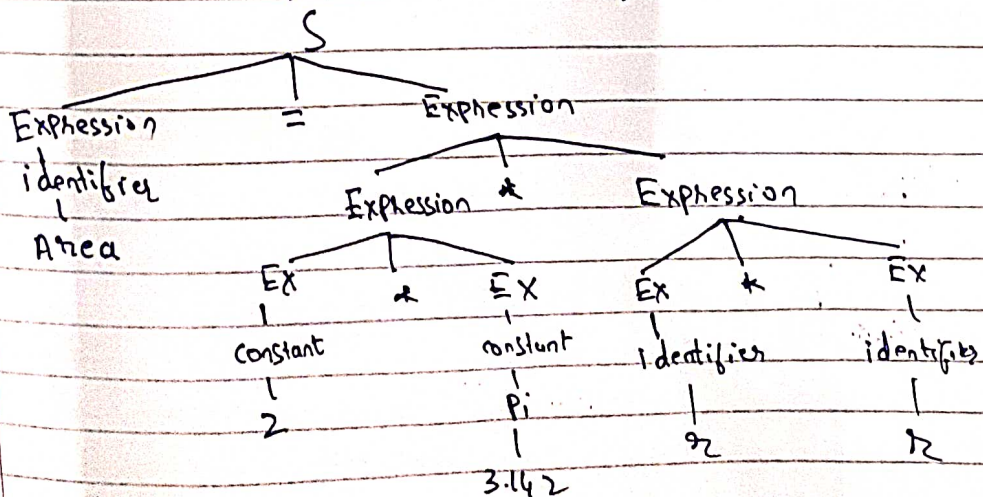
sematic analysis

Intermediate code

$t_1 = r$

$t_2 = r * r$

$t_3 = Pi * r * r$

$t_3 = 2 + t_3$

area $= t_3$

MOV	R0, PI		LDA	PI
MUL	R0, #2		MUL	#2
MOV	R1, r		MUL	id3
MUL	R1, R1		MUL	id4
MUL	R0, R1		STA	id1
MOV	Area, R0			