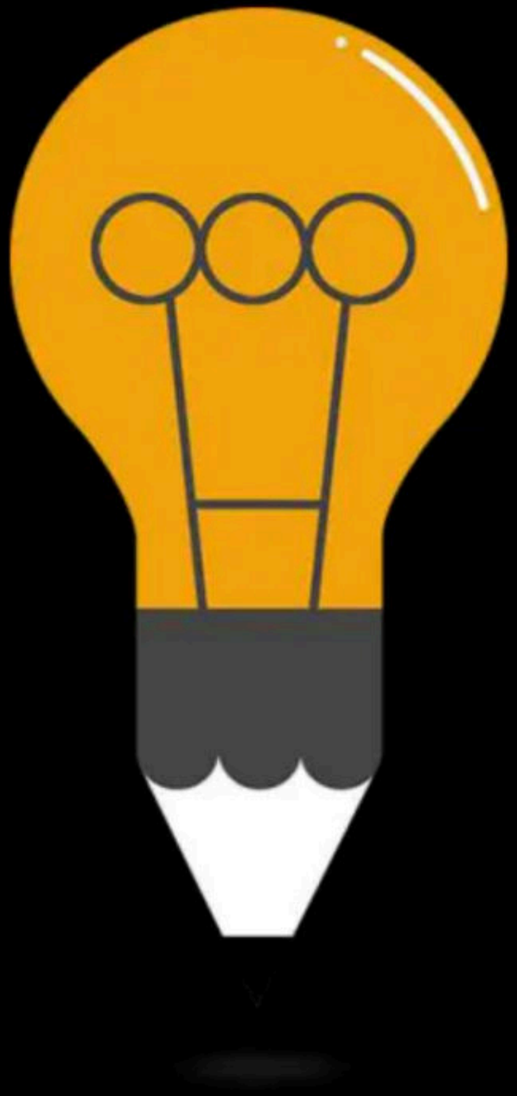
Branch Statement and Loops

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Vishvadeep Gothi • Lesson 3 • Nov 28, 2022



C-Tokens

By: Vishvadeep Gothi

C-tokens

C-tokens

1. Keywords
2. Identifiers
3. Operators
4. Literals (Constants)

Keywords

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	volatile
const	float	short	unsigned

Identifiers

Operators

1. Arithmetic
2. Logical
3. Relational
4. Bitwise Operators
5. Assignment Operators
6. Conditional Operator
7. Termination Operator ⇒ ; semicolon
8. Special Operators

Arithmetic Operators

Operator Precedence

Operator Associativity

Operator Precedence Associativity

Operator	Description	Associativity
() [ ] . -> ++ --	Parentheses: grouping or function call Brackets (array subscript) Member selection via object name Member selection via pointer Postfix increment/decrement	left-to-right
++ -- + - ! ~ (*type*) * & sizeof	Prefix increment/decrement Unary plus/minus Logical negation/bitwise complement Cast (convert value to temporary value of *type*) Dereference Address (of operand) Determine size in bytes on this implementation	right-to-left
* / %	Multiplication/division/modulus	left-to-right
+ -	Addition/subtraction	left-to-right
<< >>	Bitwise shift left, Bitwise shift right	left-to-right
< <= > >=	Relational less than/less than or equal to Relational greater than/greater than or equal to	left-to-right
== !=	Relational is equal to/is not equal to	left-to-right
&	Bitwise AND	left-to-right
^	Bitwise exclusive OR	left-to-right
|	Bitwise inclusive OR	left-to-right
&&	Logical AND	left-to-right
||	Logical OR	left-to-right
? :	Ternary conditional	right-to-left
= += -= *= /= %= &= ^= |= <<= >>=	Assignment Addition/subtraction assignment Multiplication/division assignment Modulus/bitwise AND assignment Bitwise exclusive/inclusive OR assignment Bitwise shift left/right assignment	right-to-left
,	Comma (separate expressions)	left-to-right

Logical Operators

Relational Operators

Bitwise Operators

Assignment Operators

=

+=

-=

/=

*=

%=

&&=

a += b ⇒ a = a + b

ex:-

a = ~~8~~ 3

b = 5

a %= b

a = a % b

3 % 5

Conditional Operator ⇒ Ternary operator

?:

!!

3 operands

condition ? expression 1 : expression 2

ex:-

a = 5
b = 3
c = 2

d = a ? b : c

3

d = (a<b) ? a : c

a = 5, b = 3, c = 2

$$d = (a<b)\ ?\ a+c\ :\ a+b$$

$d = 5+3$

$= 8$

$$d = (a>b)\ ?\ (b>c\ ?\ a\ :\ c)\ :\ (c>a\ :\ c\ ?\ b)$$

$d = 5$

$d = (6 != 3) ? (a = b) : (b = c)$

True

a = 5
b = 3
c = 2

$d = 1$

$d = (a = b) ? 3 : 5$

a = ~~5~~ 3
b = 3
a = 2

$d = 3$

Special Operators

* ⇒ pointer dereference

& ⇒ address extraction

-> ⇒ used to access structure member using pointers

. ⇒ —"— —"— using variables

sizeof ⇒ returns size of input in bytes

(type) ⇒ type casting

Data Types

Types	Data Types
Basic Data Type	int, char, float, double
Derived Data Type	array, pointer, structure, union
Enumeration Data Type	enum
Void Data Type	void

up to 6 decimal place

(user defined)

using basic datatype

no type

---

int ⇒ integers 0, 1, 2, 3, 4, −8, −−−

char ⇒ characters a, b, c.... z, A, ... Z,
, + − 0 1 2 9

float ⇒ real value 2.16, 3.1, −6.231
double ⇒ large size float

1 byte = (8 bits)

Data Types

Data Types	Modifiers
int ⇒ signed ⇒ 2 bytes	short
char ⇒ 1 byte	long
float ⇒ 4 bytes	unsigned
double ⇒ 8 bytes	signed
void	

short int
long int ⇒ 4bytes
long double ⇒ 10 bytes

signed int
unsigned int

Access specifiers

int ⇒ %d
long int ⇒ %ld
unsigned int ⇒ %u

hexadecimal int ⇒ %x

char ⇒ %c

float ⇒ %f

double ⇒ %f

long double ⇒ %lf

character value will always be represented under single quote.

Variables

↓

identifiers

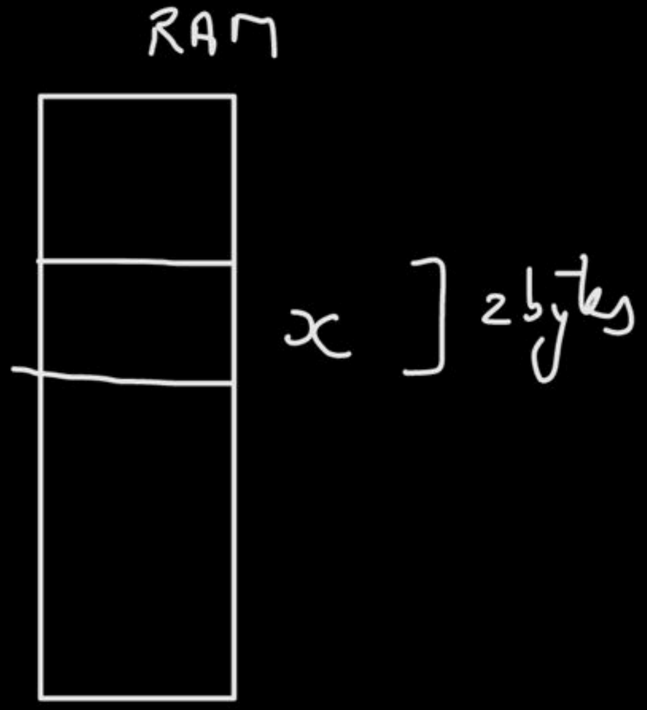
⇒ placeholders for one value

Variable declaration:-

type name ;

ex:

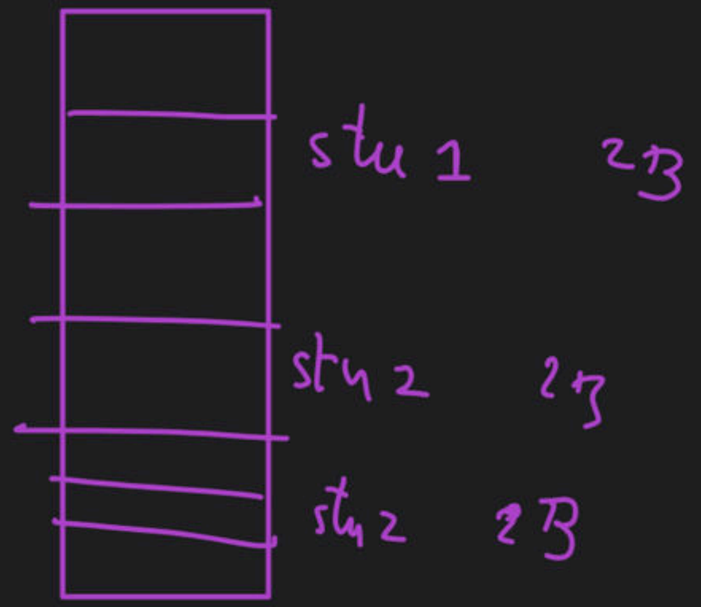
int x;



datatype var1, var2, var3;

ex:

int stu1, stu2, stu3;



int x, y, z;
char ch1, ch2;
float f1, f2, f3;
long int l1, l2;

char ch1 = 'y';

```
int a, b, c;
c = a + b;
```

what will be the value in c ? ⇒ Ans :- unpredicted garbage value

---

```
int a, b, c;
a = 5;
b = 3;
c = a + b;
```

value of c = 8

variable initialization

datatype name1 = value, name2 = value ;

ex :-

```
int a = 5, b = 3, c;
```

Output Function

printf()

- Statement
- Result of expression or any variable

printf("Vishrudeep Sir is amazing");

```
int a = 5;
printf("%d", a);          5
printf("%d", a*2);        10
```

```
int a = 5, b = 3;
printf("%d%d", a, b);     53
printf("%d %d", a, b);    5 3
printf("%d\n%d", a, b);   5
                          3
```

```
printf("%d\t%d", a, b);
```

5 3

```
int a, b, c;
a = 5;
b = 3;
c = a + b;
printf("Addition of %d and %d is = %d", a, b, c);
```

a = 5
b = 3
c = 8

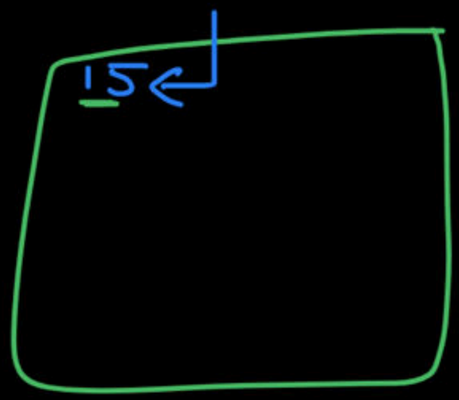
output:- Addition of 5 and 3 is = 8

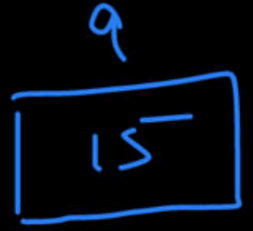
Input Function

scanf()

```
int a;

scanf("%d", &a);
printf("%d", a*2);
```





```
int a,b;
scanf("%d %d", &a, &b);
printf("%d\n %d", a, b);
```

Header File

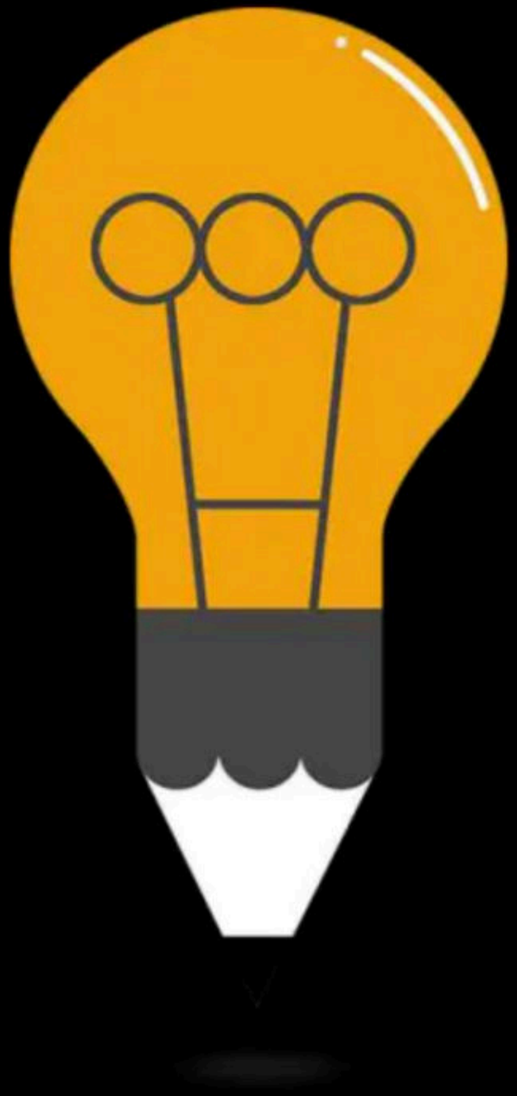
↳ in built files

↳ library function

printf()
scanf()
→ stdio.h

#include <stdio.h>

Structure of Program in C-language



DPP

By: Vishvadeep Gothi

Question

The tool used by a programmer to convert a high-level source program to an object module is a

A. Complier

B. Language translator

C. Linker

D. Preprocessor

Question

Which of the following C types could is more suitable to store pi(3.14159)?

A. Short int

B. double

C. long int

D. double imaginary

Question

Which of the following output formatting statements would print the following results?

23 z 4.100000

A. printf(“%d%c%f”, 23, z, 4.1);
B. printf(“%i%c%f”, 23, z, 4.1);
C. printf(“%d %c %f”, 23, z, 4.1);
D. printf(“%i%z%f”, 23, z, 4.1);

Question

Given the following code, what is the value of x after the print statement?

```
int x;
x = 4;
printf(“%d", --x);
```

A. 2
B. 3
C. 4
D. 5

Question

Which of the following is not a valid assignment expression?

a) a = b
b) a *= b
c) a + b = c
d) a = b = 0

Question

Which of the following expressions uses associatively?

A. a * b + c
B. a * b/c
C. + b % c
D. a - b * c

Question

Which of the following is not a logical operator?

A. if

B. not

C. and

D. or

Question

Which of the following is the complement of equal (= =)?

A. >

B. >=

C. <=

D. !=

Question

Which of the following is not a relational operator?

A. =

B. <

C. >=

D. >

Question

The _____ operator is used to extract the address for a variable.

A. address (&)

B. assignment (=)

C. indirection (*)

D. selection (>)

Happy Learning.!