

# introduction C programming language

error

- Semantic error
- Syntax error

Debugger

bug detection

editor

Compiler

Debugger

} IDE  
integrated  
Development  
environment

## System programming language

Unix Os

1972

Mid-level programming language

Case-Sensitive

Compiler (Derleyici)

Source file(s) (text)

Object file(s) (binary)

Linker

Executable File (Binary)

IDEs

Dev C++

Visual Studio

Eclipse

Netbeans

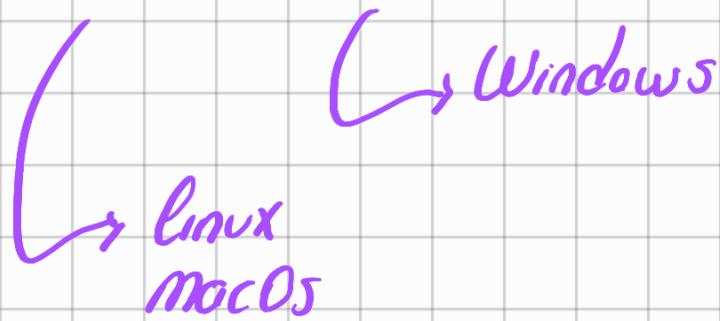
CodeBlocks

# Clion

toploma.C (source file)

toploma.o (object file)

toploma /toploma.exe (executable File)



#include <stdio.h>

"stdio.h"

int main() {

return 0; } → exit(0);

}

Read / write operations

scanf(" %d ", &ol); gets();

( → Variable )

→ operand  
printf("%d", a); putstr()

printf("Hello world");

printf("Hello world \n");

scanf("%d %d", &a, &b);

scanf("%d", &a);

scanf("%d", &b);

printf("A'nın değeri : %d - B'nın değeri  
%d", a, b);

## Aritmetic operations

+ addition

- subtraction

\* multiplication

/ Division

% modular arithmetic

5 / 2 → 2 Div

$$5.0 / 2 \quad \left. \begin{array}{l} \\ \end{array} \right\} 2.5$$

$$5 / 2.0 \quad \left. \begin{array}{l} \\ \end{array} \right\} 2.5$$

$$5.0 / 2.0 \quad \left. \begin{array}{l} \\ \end{array} \right\}$$

## Control flow (if)

$\text{if}(x)$

- non-zero (true)
- zero (false)

Statement 1.

Statement 2.

$\text{if}(1) \quad \left. \begin{array}{l} \\ \end{array} \right\}$

$\text{if}(5) \quad \left. \begin{array}{l} \\ \end{array} \right\} \text{true}$

$\text{if}(-5) \quad \left. \begin{array}{l} \\ \end{array} \right\}$

$\text{if}(0) \rightarrow \text{false}$

$\text{if}(x) \{$

Statement 1;

}

else {

Statement 2;

{

Statement 3;

if(x) {

Statement 1;

Statement 2;

}

else {

Statement 3;

Statement 4;

}

Statement 5;

## Comparison expressions

< less than

> greater than

<= less than or equal to

>= greater than or equal to

== equal to

$\neq$  not equal to

if  $\underbrace{l-1 < 0}_1 \{$   
} else {  
}

if  $l < -1 \{$   
}  
else {  
|  
|  
}

if  $\underbrace{l-1 < 0 < 1}_1 \{$   
 $\underbrace{l < 1}_\emptyset$   
}  
else {  
|  
|  
}

if  $\underbrace{l-1 < 0 < 2}_1 \{$

if  
    {  
        else {  
            }  
    }

## logical operations

And &&  
Or ||

$x < y \& z$

if ((x < y) && (y < z))

if ((x < 0) || (x > 100))

## Nested if statements

if (a < b) {  
    if (a < c) {

        d = a;

    }

    else {

        d = c;

    }

a,b,c  
① 3 4 5

```
else if(b < c) {  
    d = b  
}  
else {  
    d = c  
}  
printf("%d", d);
```

## Scalar Data Types

Char (1 byte)  
int  
float  
double

Short  
long  
unsigned  
signed

char < short int < int < long int

float < double

short int (2 byte)  
long int 4-8 byte  
int 4 byte

unsigned int a;

## Variable declaration

int a, b, c;

float x; x = 0.0;

double y, z;

char harf, m;

float x = 0.0;

int main() {

int num1, num2, result;

printf("iki sayı giriniz \n");

scanf("%d", &num1);

scanf("%d", &num2);

result = num1 + num2;

printf("Sonuç = %d", result);

return 0;

}

↳ iki sayı giriniz

5 (enter)

6 (enter)

Syntax  $\rightarrow$  II

## loops

for  
while  
do while

### for statement

for( $i=0; i < 100; i++$ )  
  └ exp1   └ exp2   └ exp3

Statement 1;

$i = i + 2$      $i = i * 2$

INC	ADD
$i++ \leftrightarrow$	$i = i + 1$
Dec	Sub
$i--$	$i = i - 1$

### Do-while statement

$i = \emptyset$

do {

i = i + 1

Statement1;

} while (i < 100);

While statement

i = 0

while (i < 100) {

Statement1;

i++;

}

scanf ("%d %d", &x, &y)

while (x < y) {

x++;

y--;

}

## Arrays

`int a[100]; → a[0] —— a[99]`

`a[100]` } segmentation  
`x[-50]` } fault  
`x[100]` } fault

`float x[50]; → x[0] —— x[49]`

`#define N 100`

`scanf("%d", &N);`

`int a[N];`

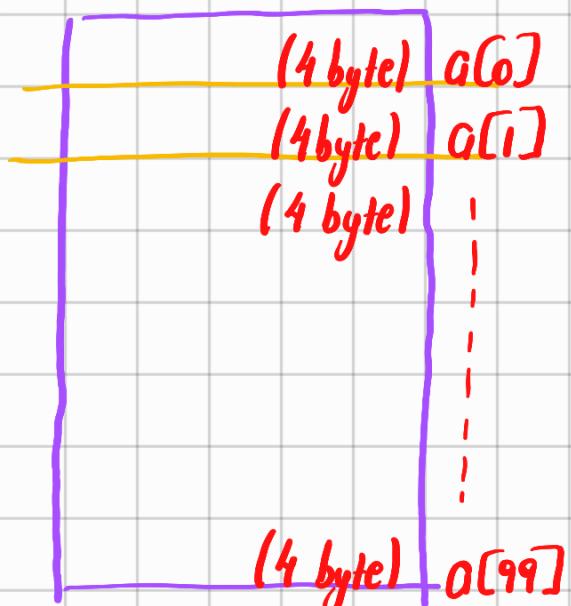
`for (i=0; i<N; i++) {`

`int a[N];  
 scanf("%d", &a[i]);`

`}`

`for (i=0; i<N; i++) {`

`printf("%f", x[i]);`



## C programming language

- general structure

read / write operation

Control flow (if)

Arithmetic operations, Composition expressions

Scalar data type

loops (for, while, do-while)

Arrays

Multi-dimensional Array

## Array of characters (string)

```
int main() {  
    // comment line  
  
    /* block 1  
     * block 2  
  
    */  
    return 0;  
}
```

Ansi C-Standart

Source file 1

Source file 2

Obj1

Obj2

Linker

# executable file

## Read / write operations

decimal %d

character %c

floating-point %f

String %s

hexadecimal %x

1	2	-3	5	0	7	6
Void type						

'a'	'b'	,	;	"	:	:	"

1.1	2.1	3.2

Char a;

scanf("%c", &a);

printf("%c", a);

printf("%d", a);