

!= (Logical operator not)	
<i>lecture01</i>	p. 23
" (Double quote)	
<i>lecture01</i>	p. 15
#define	
<i>lecture01</i>	p. 16,18
#include	
<i>lecture01</i>	p. 16,20
<i>lecture02</i>	p. 7
& (bit and)	
<i>lecture01</i>	p. 23
& operator (address)	
<i>lecture01</i>	p. 10
<i>lecture03</i>	p. 4,6,8
&& (Logical operator and)	
<i>lecture01</i>	p. 23
' (Single quote)	
<i>lecture01</i>	p. 15
* operator (dereferencing)	
<i>lecture01</i>	p. 10
<i>lecture03</i>	p. 7
- (arrow) reference to structure	
<i>lecture03</i>	p. 19,20
. (dot) reference to structure	
<i>lecture03</i>	p. 16,20
.h File	
<i>lecture01</i>	p. 16
\0	
<i>lecture01</i>	p. 14,15
<i>lecture02</i>	p. 10-12
\n	
<i>lecture02</i>	p. 10
((Curly brackets)	
<i>lecture01</i>	p. 22
(bit or)	
<i>lecture01</i>	p. 23
(Logical operator or)	
<i>lecture01</i>	p. 23

A

Address

<i>lecture01</i>	p. 8-10,12,13
Address of variable	
<i>lecture03</i>	p. 4
Algorithm	
<i>lecture01</i>	p. 12
Alignment of structures	
<i>lecture03</i>	p. 18
And (logical operator)	
<i>lecture01</i>	p. 23
Angle brackets vs double quotes for header files	
<i>lecture02</i>	p. 7
Architecture	
<i>lecture01</i>	p. 10
argc	
<i>lecture02</i>	p. 2
<i>lecture03</i>	p. 14,15
<i>lecture03</i>	p. 13
argv[]	
<i>lecture02</i>	p. 2
<i>lecture03</i>	p. 13-15
Array	
<i>lecture01</i>	p. 10,12-14
<i>lecture03</i>	p. 6
Array in C and in Java	
<i>lecture03</i>	p. 12
Array of strings	
<i>lecture03</i>	p. 12,13
Array of structures	
<i>lecture03</i>	p. 16
Array vs pointer	
<i>lecture03</i>	p. 6,8,9,12
Array – multidimensional	
<i>lecture03</i>	p. 13
Array: returned by a function	
<i>lecture04</i>	p. 11-13
Arrow reference to structure field	
<i>lecture03</i>	p. 19,20
ASCII	
<i>lecture01</i>	p. 9
ASCII table	
<i>lecture01</i>	p. 22
assert	
<i>lecture02</i>	p. 4
Assigning address to pointer	

<i>lecture03</i>	p. 6
Assignment	
<i>lecture01</i>	p. 22,23
<i>lecture02</i>	p. 2
atoi()	
<i>lecture03</i>	p. 1
atoi()	
<i>lecture03</i>	p. 1
atol()	
<i>lecture03</i>	p. 1
B	
Bell Labs	
<i>lecture01</i>	p. 6
Bell labs	
<i>lecture01</i>	p. 7
Binary file	
<i>lecture04</i>	p. 2
Bit	
<i>lecture01</i>	p. 8
Bit operators	
<i>lecture01</i>	p. 23
Block	
<i>lecture01</i>	p. 22
Block of instructions	
<i>lecture01</i>	p. 15
Boolean	
<i>lecture01</i>	p. 10,11,21
Box-Müller	
<i>lecture03</i>	p. 3
break	
<i>lecture01</i>	p. 24
Built-in functions	
<i>lecture02</i>	p. 5,8
Byte	
<i>lecture01</i>	p. 8

C

C environment	
<i>lecture01</i>	p. 7
C program structure	

<i>lecture01</i>	p. 18
C standard library	
<i>lecture02</i>	p. 8
C vs Java	
<i>lecture01</i>	p. 6,12-14,17,18
<i>lecture02</i>	p. 5
<i>lecture03</i>	p. 12,20
<i>lecture04</i>	p. 7,17
C11	
<i>lecture01</i>	p. 7
C89	
<i>lecture01</i>	p. 7
C99	
<i>lecture01</i>	p. 7
Calling functions	
<i>lecture04</i>	p. 8
calloc()	
<i>lecture04</i>	p. 18
case	
<i>lecture01</i>	p. 24
Case	
<i>lecture02</i>	p. 11
Case insensitive comparison	
<i>lecture02</i>	p. 13
CFLAGS	
<i>lecture04</i>	p. 5
Changing case	
<i>lecture02</i>	p. 11
char	
<i>lecture01</i>	p. 10,11
Character classification	
<i>lecture02</i>	p. 10,11
Character encoding	
<i>lecture01</i>	p. 15,22
Character Encoding	
<i>lecture02</i>	p. 17
Character encoding	
<i>lecture02</i>	p. 15-18
Chinese characters	
<i>lecture02</i>	p. 14,16,17
CJK	
<i>lecture02</i>	p. 15
Classification of characters	
<i>lecture02</i>	p. 10,11
Code	

<i>lecture01</i>	p. 8
codepoint	
<i>lecture02</i>	p. 15
<i>lecture02</i>	p. 15
Command-line parameters	
<i>lecture02</i>	p. 2
<i>lecture03</i>	p. 13-15
Comparison of strings	
<i>lecture02</i>	p. 12,13
Comparison operators	
<i>lecture01</i>	p. 22,23
<i>lecture02</i>	p. 2
Compiler	
<i>lecture01</i>	p. 16,17,19,20
Compiling a C program	
<i>lecture01</i>	p. 17
Compiling on Linux	
<i>lecture01</i>	p. 17
Condition	
<i>lecture01</i>	p. 21
Constants	
<i>lecture01</i>	p. 16,18
Course expectations	
<i>lecture01</i>	p. 2
Course notes	
<i>lecture01</i>	p. 3
Course Organization	
<i>lecture01</i>	p. 6
Course schedule	
<i>lecture01</i>	p. 1
Craftsmanship	
<i>lecture01</i>	p. 5
ctime()	
<i>lecture03</i>	p. 2,3,21
ctype.h	
<i>lecture02</i>	p. 10
<i>lecture02</i>	p. 11
Curly brackets	
<i>lecture01</i>	p. 15,22
Cygwin	
<i>lecture01</i>	p. 2

D

Data	
<i>lecture01</i>	p. 8
Data types	
<i>lecture01</i>	p. 10-12
Declaration of pointer	
<i>lecture03</i>	p. 5,6
Declaration of variable	
<i>lecture01</i>	p. 9
Deleting a file	
<i>lecture04</i>	p. 2
Dereferencing	
<i>lecture03</i>	p. 7,8,19,20
Direct access	
<i>lecture04</i>	p. 2
Directory operations	
<i>lecture04</i>	p. 3
dirent.h	
<i>lecture04</i>	p. 3
Distribution	
<i>lecture03</i>	p. 3
do ... while	
<i>lecture02</i>	p. 1
Dot reference to structure field	
<i>lecture03</i>	p. 16,20
double	
<i>lecture01</i>	p. 12
Double quote	
<i>lecture01</i>	p. 15
Double quotes vs angle brackets for header files	
<i>lecture02</i>	p. 7
Dumping a binary file	
<i>lecture04</i>	p. 3
Dynamic memory	
<i>lecture04</i>	p. 16-18
Dynamic memory example	
<i>lecture04</i>	p. 19

E

else	
<i>lecture01</i>	p. 21,23
else if	
<i>lecture01</i>	p. 23

Encoding	
<i>lecture01</i>	p. 9
<i>lecture02</i>	p. 15
End-of-string marker	
<i>lecture01</i>	p. 14,15
EOF	
<i>lecture02</i>	p. 9,10
Epoch	
<i>lecture03</i>	p. 2
errno	
<i>lecture03</i>	p. 1
errno.h	
<i>lecture03</i>	p. 1
Error checking	
<i>lecture02</i>	p. 2-4
<i>lecture03</i>	p. 1
Error management	
<i>lecture02</i>	p. 4,5
Exam	
<i>lecture01</i>	p. 3
Exam dates	
<i>lecture01</i>	p. 3
Example of pointer usage	
<i>lecture03</i>	p. 8
Example: day of the week when you were born	
<i>lecture03</i>	p. 21-23
Exams	
<i>lecture01</i>	p. 2-4
Exception	
<i>lecture02</i>	p. 4,5
Executable	
<i>lecture01</i>	p. 16
Expectations	
<i>lecture01</i>	p. 2
Exponent	
<i>lecture01</i>	p. 12
Exponential distribution	
<i>lecture03</i>	p. 3
extern	
<i>lecture04</i>	p. 7
<i>lecture04</i>	p. 7

F

fclose()	
<i>lecture03</i>	p. 25
fepf()	
<i>lecture04</i>	p. 1
ferror()	
<i>lecture04</i>	p. 1
fgetc()	
<i>lecture02</i>	p. 9
<i>lecture04</i>	p. 1
fgets()	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 10
<i>lecture04</i>	p. 14
fgets():Return value	
<i>lecture03</i>	p. 1
FILE	
<i>lecture03</i>	p. 26
FILE *	
<i>lecture03</i>	p. 25
Files	
<i>lecture03</i>	p. 24-26
Final exam	
<i>lecture01</i>	p. 3
float	
<i>lecture01</i>	p. 12
flock()	
<i>lecture04</i>	p. 2
Flow control	
<i>lecture01</i>	p. 21,23,24
<i>lecture02</i>	p. 1
fopen()	
<i>lecture03</i>	p. 25,26
for	
<i>lecture02</i>	p. 1
Formatted input and output	
<i>lecture02</i>	p. 10
fprint()	
<i>lecture03</i>	p. 26
fprintf()	
<i>lecture02</i>	p. 10
fputc()	
<i>lecture02</i>	p. 9
<i>lecture04</i>	p. 1
fputs()	
<i>lecture02</i>	p. 10

<i>lecture03</i>	p. 26
fread()	
<i>lecture04</i>	p. 1
free()	
<i>lecture04</i>	p. 19
<i>lecture04</i>	p. 18,21-23
fseek()	
<i>lecture04</i>	p. 2
Function call	
<i>lecture04</i>	p. 8-12
Function declaration	
<i>lecture02</i>	p. 6,7
Function identification	
<i>lecture02</i>	p. 5,6
Function nesting	
<i>lecture01</i>	p. 15
Function prototype	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 7
Function: Pointers as argument	
<i>lecture04</i>	p. 13,14
Function: returning an array	
<i>lecture04</i>	p. 11-13
Functions	
<i>lecture04</i>	p. 8
Functions, nesting	
<i>lecture02</i>	p. 6
fwrite()	
<i>lecture04</i>	p. 1

G

Garbage collector	
<i>lecture04</i>	p. 19
gcc	
<i>lecture01</i>	p. 17
<i>lecture01</i>	p. 7
gcd()	
<i>lecture04</i>	p. 8,9
getchar()	
<i>lecture02</i>	p. 9
getopt()	
<i>lecture03</i>	p. 15
gets()	

<i>lecture02</i>	p. 10
Global variable	
<i>lecture03</i>	p. 1
Global variables	
<i>lecture04</i>	p. 15
gmtime()	
<i>lecture03</i>	p. 21
Grades	
<i>lecture01</i>	p. 4

H

head	
<i>lecture04</i>	p. 3
Header file	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 7
Heap	
<i>lecture01</i>	p. 8
<i>lecture04</i>	p. 17
Help on functions	
<i>lecture02</i>	p. 5
History of C	
<i>lecture01</i>	p. 7
Honesty	
<i>lecture01</i>	p. 5

I

if	
<i>lecture01</i>	p. 21,23
Initialization of pointer	
<i>lecture03</i>	p. 7,8
Initialization of structure	
<i>lecture03</i>	p. 16
Input/Output	
<i>lecture02</i>	p. 9,10
int	
<i>lecture01</i>	p. 11
integer operations	
<i>lecture01</i>	p. 11
isalnum()	
<i>lecture02</i>	p. 11

isalpha()	
<i>lecture02</i>	p. 11
isdigit()	
<i>lecture02</i>	p. 11
islower()	
<i>lecture02</i>	p. 11
ISO	
<i>lecture02</i>	p. 16
isprint()	
<i>lecture02</i>	p. 11
ispunct()	
<i>lecture02</i>	p. 11
isspace()	
<i>lecture02</i>	p. 11
isupper()	
<i>lecture02</i>	p. 11

J

Java vs C	
<i>lecture01</i>	p. 6,12-14,17,18
<i>lecture02</i>	p. 5
<i>lecture03</i>	p. 12,20
<i>lecture04</i>	p. 7
java vs C	
<i>lecture04</i>	p. 17

K

K&R	
<i>lecture01</i>	p. 6
Kernighan (Brian)	
<i>lecture01</i>	p. 6

L

Labs	
<i>lecture01</i>	p. 3,4
ld	
<i>lecture01</i>	p. 20
Library file	
<i>lecture04</i>	p. 6

Linker	
<i>lecture01</i>	p. 16,17,19,20
<i>lecture04</i>	p. 7-9
Linux	
<i>lecture01</i>	p. 2
localtime()	
<i>lecture03</i>	p. 21
Locking a file	
<i>lecture04</i>	p. 2
Logical operators	
<i>lecture01</i>	p. 23
long	
<i>lecture01</i>	p. 11
<i>lecture01</i>	p. 11
Loop	
<i>lecture02</i>	p. 1

M

main()	
<i>lecture01</i>	p. 16
make	
<i>lecture01</i>	p. 7
<i>lecture04</i>	p. 5,6
<i>lecture04</i>	p. 4-6
Makefile	
<i>lecture04</i>	p. 5,6
malloc()	
<i>lecture04</i>	p. 18-20
man	
<i>lecture02</i>	p. 5
Marker (end-of-string)	
<i>lecture01</i>	p. 14,15
Mathematical functions	
<i>lecture01</i>	p. 19,20
Mathematical functions:Compiler	
<i>lecture01</i>	p. 19
memory	
<i>lecture01</i>	p. 8
<i>lecture01</i>	p. 8
Memory address	
<i>lecture01</i>	p. 9,10
Memory leak	
<i>lecture04</i>	p. 22

Midcourse exam	
<i>lecture01</i>	p. 3
mktime()	
<i>lecture03</i>	p. 21
Multi-threading	
<i>lecture04</i>	p. 15,16
Multidimensional array	
<i>lecture03</i>	p. 13

N

Name of variable	
<i>lecture01</i>	p. 9
Naming a structure	
<i>lecture03</i>	p. 16,17
Nesting functions	
<i>lecture02</i>	p. 6
Normal distribution	
<i>lecture03</i>	p. 3
Not (logical operator)	
<i>lecture01</i>	p. 23
NULL	
<i>lecture02</i>	p. 10,13
<i>lecture03</i>	p. 1,7

O

od	
<i>lecture04</i>	p. 3
Or (logical operator)	
<i>lecture01</i>	p. 23
Overflow	
<i>lecture02</i>	p. 12
Overloading	
<i>lecture02</i>	p. 5

P

perror()	
<i>lecture03</i>	p. 1
Pipe	
<i>lecture02</i>	p. 9

Pointer	
<i>lecture01</i>	p. 10
<i>lecture03</i>	p. 4-8,19,20
Pointer arithmetic	
<i>lecture03</i>	p. 10,11
Pointer on structure	
<i>lecture03</i>	p. 19,20
Pointer to a file	
<i>lecture03</i>	p. 25
Pointer vs array	
<i>lecture03</i>	p. 6,8,9,12
Pointers	
<i>lecture04</i>	p. 11,12
Pointers as arguments to a function	
<i>lecture04</i>	p. 13,14
pptx	
<i>lecture04</i>	p. 3
Preprocessor	
<i>lecture01</i>	p. 16-18,20
printf()	
<i>lecture02</i>	p. 8,10
Prototype (function)	
<i>lecture01</i>	p. 16
Prototype (functions)	
<i>lecture02</i>	p. 7
putchar()	
<i>lecture02</i>	p. 9
puts()	
<i>lecture02</i>	p. 10

Q

Quality	
<i>lecture01</i>	p. 5

R

Radix	
<i>lecture01</i>	p. 12
random()	
<i>lecture03</i>	p. 2,3
Reading ZIP or XML	
<i>lecture04</i>	p. 3

realloc()	
<i>lecture04</i>	p. 18
Reference	
<i>lecture03</i>	p. 7
Reference to structure filed	
<i>lecture03</i>	p. 16
Return value	
<i>lecture02</i>	p. 3,4,8
Return value from main()	
<i>lecture01</i>	p. 16
Ritchie (Dennis)	
<i>lecture01</i>	p. 6,7
Ritchie, Dennis	
<i>lecture01</i>	p. 6
Robustness	
<i>lecture01</i>	p. 5
Rounding error	
<i>lecture01</i>	p. 12

S

scanf()	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 3,4,10
<i>lecture04</i>	p. 14
Schedule	
<i>lecture01</i>	p. 1
Semi-colon	
<i>lecture01</i>	p. 15
setlocale	
<i>lecture02</i>	p. 15
setlocale()	
<i>lecture03</i>	p. 3
Shared library	
<i>lecture04</i>	p. 7
short	
<i>lecture01</i>	p. 11
signed	
<i>lecture01</i>	p. 11,12
Single quote	
<i>lecture01</i>	p. 15
sizeof()	
<i>lecture03</i>	p. 6,13
sscanf()	

<i>lecture01</i>	p. 16
Stack	
<i>lecture01</i>	p. 8
<i>lecture04</i>	p. 9-12
static	
<i>lecture04</i>	p. 7,16
<i>lecture04</i>	p. 16
Static variables	
<i>lecture04</i>	p. 16
stderr	
<i>lecture02</i>	p. 9
<i>lecture02</i>	p. 9
<i>lecture03</i>	p. 1
stdin	
<i>lecture02</i>	p. 9
<i>lecture02</i>	p. 9,10
<i>lecture03</i>	p. 24,25
stdio.h	
<i>lecture03</i>	p. 25
stdlib.h	
<i>lecture03</i>	p. 1
<i>lecture04</i>	p. 18
stdout	
<i>lecture02</i>	p. 9,10
<i>lecture02</i>	p. 9,10
<i>lecture03</i>	p. 24,25
strcasecmp()	
<i>lecture02</i>	p. 13
strcat()	
<i>lecture02</i>	p. 12
strchr()	
<i>lecture02</i>	p. 13
strcmp()	
<i>lecture02</i>	p. 12,13
strcpy()	
<i>lecture02</i>	p. 12
strdup()	
<i>lecture04</i>	p. 18
Stream	
<i>lecture02</i>	p. 9
Stream redirection	
<i>lecture03</i>	p. 24
strerror()	
<i>lecture03</i>	p. 1
String	

lecture01 p. 10,14,15

String array

lecture03 p. 12,13

String comparison

lecture02 p. 12,13

String conversion to number

lecture03 p. 1

String declaration

lecture03 p. 11

String search

lecture02 p. 13

string.h

lecture02 p. 11-13

lecture04 p. 18

Strings

lecture02 p. 11-13

lecture03 p. 1

strlen()

lecture02 p. 11

strncasecmp()

lecture02 p. 13

strncat()

lecture02 p. 12

strncmp()

lecture02 p. 12,13

strncpy()

lecture02 p. 12

strrchr()

lecture02 p. 13

strsep()

lecture02 p. 14

strstr()

lecture02 p. 13

strtod()

lecture03 p. 1

strtok()

lecture02 p. 13,14

strtol()

lecture03 p. 1

Struct

lecture03 p. 20

struct

lecture03 p. 15,17,20,23

lecture03 p. 16-20

struct tm

lecture03 p. 21

Structure alignment

lecture03 p. 18

Structure and pointer

lecture03 p. 19,20

Structure initialization

lecture03 p. 16

Structure naming

lecture03 p. 16,17

Structures

lecture03 p. 15-20

switch

lecture01 p. 24

T

Thomson (Ken)

lecture01 p. 7

Thomson, Ken

lecture01 p. 6

Time functions

lecture03 p. 2,20,21

time()

lecture03 p. 2,21

time.h

lecture03 p. 21

lecture03 p. 2,21

timegm()

lecture03 p. 21

time_t

lecture03 p. 21

lecture03 p. 2

Tokenizing

lecture02 p. 13,14

tolower()

lecture02 p. 11

toupper()

lecture02 p. 11

typedef

lecture03 p. 17

U

Unicode	
<i>lecture02</i>	p. 15,17,18
union	
<i>lecture03</i>	p. 23
<i>lecture03</i>	p. 24
UNIX	
<i>lecture01</i>	p. 6
Unix	
<i>lecture01</i>	p. 7
Unix pipe	
<i>lecture02</i>	p. 9
unlink()	
<i>lecture04</i>	p. 2
unsigned	
<i>lecture01</i>	p. 11,12
UTF-16	
<i>lecture02</i>	p. 17
UTF-32	
<i>lecture02</i>	p. 17
UTF-8	
<i>lecture02</i>	p. 15,18

V

Variable declaration	
<i>lecture01</i>	p. 9
Variable name	
<i>lecture01</i>	p. 9
Variable number of parameters	
<i>lecture02</i>	p. 6
Visual Studio	
<i>lecture01</i>	p. 7
void	
<i>lecture04</i>	p. 13
void*	
<i>lecture04</i>	p. 18
Von Neumann (John)	
<i>lecture01</i>	p. 8
Von Neumann, John	
<i>lecture04</i>	p. 9

W

wchar	
<i>lecture02</i>	p. 14,15
while	
<i>lecture02</i>	p. 1
Wide char	
<i>lecture02</i>	p. 14,15
X	
Xcode	
<i>lecture01</i>	p. 7
XML	
<i>lecture04</i>	p. 3
Z	
ZIP	
<i>lecture04</i>	p. 3