

!= (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
&& (Logical operator and)	
<i>lecture01</i>	p. 23
' (Single quote)	
<i>lecture01</i>	p. 15
* operator (dereferencing)	
<i>lecture01</i>	p. 10
.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
Algorithm	
<i>lecture01</i>	p. 12
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
argv	
<i>lecture02</i>	p. 2
Array	
<i>lecture01</i>	p. 10,12-14
ASCII	
<i>lecture01</i>	p. 9
ASCII table	
<i>lecture01</i>	p. 22
assert	
<i>lecture02</i>	p. 4
Assignment	
<i>lecture01</i>	p. 22,23
<i>lecture02</i>	p. 2

B

Bell Labs	
<i>lecture01</i>	p. 6
Bell labs	
<i>lecture01</i>	p. 7
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
break	
<i>lecture01</i>	p. 24
Built-in functions	
<i>lecture02</i>	p. 5,8
Byte	
<i>lecture01</i>	p. 8

C

C environment

lecture01 p. 7

C program structure

lecture01 p. 18

C standard library

lecture02 p. 8

C vs Java

lecture01 p. 6,12-14,17,18

lecture02 p. 5

C11

lecture01 p. 7

C89

lecture01 p. 7

C99

lecture01 p. 7

case

lecture01 p. 24

Case

lecture02 p. 11

Case insensitive comparison

lecture02 p. 13

Changing case

lecture02 p. 11

char

lecture01 p. 10,11

Character classification

lecture02 p. 10,11

Character encoding

lecture01 p. 15,22

Character Encoding

lecture02 p. 17

Character encoding

lecture02 p. 15-18

Chinese characters

lecture02 p. 14,16,17

CJK

lecture02 p. 15

Classification of characters

lecture02 p. 10,11

Code

lecture01 p. 8

codepoint

lecture02 p. 15

lecture02 p. 15

Command-line parameters

lecture02 p. 2

Comparison of strings

lecture02 p. 12,13

Comparison operators

lecture01 p. 22,23

lecture02 p. 2

Compiler

lecture01 p. 16,17,19,20

Compiling a C program

lecture01 p. 17

Compiling on Linux

lecture01 p. 17

Condition

lecture01 p. 21

Constants

lecture01 p. 16,18

Course expectations

lecture01 p. 2

Course notes

lecture01 p. 3

Course Organization

lecture01 p. 6

Course schedule

lecture01 p. 1

Craftsmanship

lecture01 p. 5

ctype.h

lecture02 p. 10

lecture02 p. 11

Curly brackets

lecture01 p. 15,22

Cygwin

lecture01 p. 2

D

Data

lecture01 p. 8

Data types

lecture01 p. 10-12

Declaration of variable

<i>lecture01</i>	p. 9
do ... while	
<i>lecture02</i>	p. 1
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

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
Error checking	
<i>lecture02</i>	p. 3,4
Error management	
<i>lecture02</i>	p. 4,5
Exam	
<i>lecture01</i>	p. 3
Exam dates	
<i>lecture01</i>	p. 3
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

F

fgetc()	
<i>lecture02</i>	p. 9
fgets()	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 10
Final exam	
<i>lecture01</i>	p. 3
float	
<i>lecture01</i>	p. 12
Flow control	
<i>lecture01</i>	p. 21,23,24
<i>lecture02</i>	p. 1
for	
<i>lecture02</i>	p. 1
Formatted input and output	
<i>lecture02</i>	p. 10
fprintf()	
<i>lecture02</i>	p. 10
fputc()	
<i>lecture02</i>	p. 9
fputs()	
<i>lecture02</i>	p. 10
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
Functions, nesting	
<i>lecture02</i>	p. 6

G

gcc	
<i>lecture01</i>	p. 17
<i>lecture01</i>	p. 7
getchar()	
<i>lecture02</i>	p. 9

gets()	
<i>lecture02</i>	p. 10
Grades	
<i>lecture01</i>	p. 4

H

Header file	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 7
Heap	
<i>lecture01</i>	p. 8
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
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

K

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

L

Labs	
<i>lecture01</i>	p. 3,4
ld	
<i>lecture01</i>	p. 20
Linker	
<i>lecture01</i>	p. 16,17,19,20
Linux	
<i>lecture01</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
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
Midcourse exam	
<i>lecture01</i>	p. 3

N

Name of variable	
<i>lecture01</i>	p. 9
Nesting functions	
<i>lecture02</i>	p. 6
Not (logical operator)	
<i>lecture01</i>	p. 23
NULL	
<i>lecture02</i>	p. 10,13

O

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

P

Pipe	
<i>lecture02</i>	p. 9
Pointer	

<i>lecture01</i>	p. 10
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
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
Schedule	

<i>lecture01</i>	p. 1
Semi-colon	
<i>lecture01</i>	p. 15
setlocale	
<i>lecture02</i>	p. 15
short	
<i>lecture01</i>	p. 11
signed	
<i>lecture01</i>	p. 11,12
Single quote	
<i>lecture01</i>	p. 15
sscanf()	
<i>lecture01</i>	p. 16
Stack	
<i>lecture01</i>	p. 8
stderr	
<i>lecture02</i>	p. 9
<i>lecture02</i>	p. 9
stdin	
<i>lecture02</i>	p. 9
<i>lecture02</i>	p. 9,10
stdout	
<i>lecture02</i>	p. 9,10
<i>lecture02</i>	p. 9,10
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
Stream	
<i>lecture02</i>	p. 9
String	
<i>lecture01</i>	p. 10,14,15
String comparison	
<i>lecture02</i>	p. 12,13
String search	
<i>lecture02</i>	p. 13
string.h	
<i>lecture02</i>	p. 11-13
Strings	

<i>lecture02</i>	p. 11-13
strlen()	
<i>lecture02</i>	p. 11
strncasecmp()	
<i>lecture02</i>	p. 13
strncat()	
<i>lecture02</i>	p. 12
strncmp()	
<i>lecture02</i>	p. 12,13
strncpy()	
<i>lecture02</i>	p. 12
strrchr()	
<i>lecture02</i>	p. 13
strsep()	
<i>lecture02</i>	p. 14
strstr()	
<i>lecture02</i>	p. 13
strtok()	
<i>lecture02</i>	p. 13,14
switch	
<i>lecture01</i>	p. 24

T

Thomson (Ken)	
<i>lecture01</i>	p. 7
Thomson, Ken	
<i>lecture01</i>	p. 6
Tokenizing	
<i>lecture02</i>	p. 13,14
tolower()	
<i>lecture02</i>	p. 11
toupper()	
<i>lecture02</i>	p. 11

U

Unicode	
<i>lecture02</i>	p. 15,17,18
UNIX	
<i>lecture01</i>	p. 6
Unix	
<i>lecture01</i>	p. 7

Unix pipe	
<i>lecture02</i>	p. 9
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
Von Neumann (John)	
<i>lecture01</i>	p. 8

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