

!= (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
<b>argc</b>	
<i>lecture02</i>	p. 2
<i>lecture03</i>	p. 14,15
<i>lecture03</i>	p. 13
<b>argv[ ]</b>	
<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
Arrow reference to structure field	
<i>lecture03</i>	p. 19,20
ASCII	
<i>lecture01</i>	p. 9
ASCII table	
<i>lecture01</i>	p. 22
<b>assert</b>	
<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
<b>atof()</b>	
<i>lecture03</i>	p. 1
<b>atoi()</b>	
<i>lecture03</i>	p. 1
<b>atol()</b>	
<i>lecture03</i>	p. 1

## 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
Box-Müller	
<i>lecture03</i>	p. 3
<b>break</b>	
<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
C11	
<i>lecture01</i>	p. 7
C89	
<i>lecture01</i>	p. 7
C99	
<i>lecture01</i>	p. 7
<b>case</b>	
<i>lecture01</i>	p. 24
Case	
<i>lecture02</i>	p. 11
Case insensitive comparison	
<i>lecture02</i>	p. 13
Changing case	
<i>lecture02</i>	p. 11
<b>char</b>	
<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
<b>codepoint</b>	
<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
<b>ctime()</b>	
<i>lecture03</i>	p. 2,3,21
<b>ctype.h</b>	
<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
Dereferencing	
<i>lecture03</i>	p. 7,8,19,20
Distribution	

<i>lecture03</i>	p. 3
<b>do ... while</b>	
<i>lecture02</i>	p. 1
Dot reference to structure field	
<i>lecture03</i>	p. 16,20
<b>double</b>	
<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

<b>else</b>	
<i>lecture01</i>	p. 21,23
<b>else if</b>	
<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
<b>EOF</b>	
<i>lecture02</i>	p. 9,10
Epoch	
<i>lecture03</i>	p. 2
<b>errno</b>	
<i>lecture03</i>	p. 1
<b>errno.h</b>	
<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

## F

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

<b>fprint()</b>	
<i>lecture03</i>	p. 26
<b>fprintf()</b>	
<i>lecture02</i>	p. 10
<b>fputc()</b>	
<i>lecture02</i>	p. 9
<b>fputs()</b>	
<i>lecture02</i>	p. 10
<i>lecture03</i>	p. 26
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

<b>gcc</b>	
<i>lecture01</i>	p. 17
<i>lecture01</i>	p. 7
<b>getchar()</b>	
<i>lecture02</i>	p. 9
<b>getopt()</b>	
<i>lecture03</i>	p. 15
<b>gets()</b>	
<i>lecture02</i>	p. 10
Global variable	
<i>lecture03</i>	p. 1
<b>gmtime()</b>	
<i>lecture03</i>	p. 21
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

<b>if</b>	
<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
<b>int</b>	
<i>lecture01</i>	p. 11
<b>integer operations</b>	
<i>lecture01</i>	p. 11
<b>isalnum()</b>	
<i>lecture02</i>	p. 11
<b>isalpha()</b>	
<i>lecture02</i>	p. 11
<b>isdigit()</b>	
<i>lecture02</i>	p. 11
<b>islower()</b>	
<i>lecture02</i>	p. 11
ISO	
<i>lecture02</i>	p. 16
<b>isprint()</b>	
<i>lecture02</i>	p. 11
<b>ispunct()</b>	
<i>lecture02</i>	p. 11
<b>isspace()</b>	
<i>lecture02</i>	p. 11
<b>isupper()</b>	
<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

## K

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

## L

Labs	
<i>lecture01</i>	p. 3,4
<b>ld</b>	
<i>lecture01</i>	p. 20
Linker	
<i>lecture01</i>	p. 16,17,19,20
Linux	
<i>lecture01</i>	p. 2
<b>localtime()</b>	
<i>lecture03</i>	p. 21
Logical operators	
<i>lecture01</i>	p. 23
<b>long</b>	
<i>lecture01</i>	p. 11
<i>lecture01</i>	p. 11
Loop	
<i>lecture02</i>	p. 1

## M

<b>main()</b>	
<i>lecture01</i>	p. 16
<b>make</b>	
<i>lecture01</i>	p. 7
<b>man</b>	

<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
<b>memory</b>	
<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
<b>mktime()</b>	
<i>lecture03</i>	p. 21
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
<b>NULL</b>	
<i>lecture02</i>	p. 10,13
<i>lecture03</i>	p. 1,7

## O

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

## P

<b>perror()</b>	
<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
Preprocessor	
<i>lecture01</i>	p. 16-18,20
<b>printf()</b>	
<i>lecture02</i>	p. 8,10
Prototype (function)	
<i>lecture01</i>	p. 16
Prototype (functions)	
<i>lecture02</i>	p. 7
<b>putchar()</b>	
<i>lecture02</i>	p. 9
<b>puts()</b>	
<i>lecture02</i>	p. 10

## Q

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

## R

Radix	
<i>lecture01</i>	p. 12
<b>random()</b>	
<i>lecture03</i>	p. 2,3

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

<b>scanf()</b>	
<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
<b>setlocale</b>	
<i>lecture02</i>	p. 15
<b>setlocale()</b>	
<i>lecture03</i>	p. 3
<b>short</b>	
<i>lecture01</i>	p. 11
<b>signed</b>	
<i>lecture01</i>	p. 11,12
Single quote	
<i>lecture01</i>	p. 15
<b>sizeof()</b>	
<i>lecture03</i>	p. 6,13
<b>sscanf()</b>	
<i>lecture01</i>	p. 16
Stack	
<i>lecture01</i>	p. 8
<b>stderr</b>	
<i>lecture02</i>	p. 9

<i>lecture02</i>	p. 9
<i>lecture03</i>	p. 1
<b>stdin</b>	
<i>lecture02</i>	p. 9
<i>lecture02</i>	p. 9,10
<i>lecture03</i>	p. 24,25
<b>stdio.h</b>	
<i>lecture03</i>	p. 25
<b>stdlib.h</b>	
<i>lecture03</i>	p. 1
<b>stdout</b>	
<i>lecture02</i>	p. 9,10
<i>lecture02</i>	p. 9,10
<i>lecture03</i>	p. 24,25
<b>strcasecmp()</b>	
<i>lecture02</i>	p. 13
<b>strcat()</b>	
<i>lecture02</i>	p. 12
<b>strchr()</b>	
<i>lecture02</i>	p. 13
<b>strcmp()</b>	
<i>lecture02</i>	p. 12,13
<b>strcpy()</b>	
<i>lecture02</i>	p. 12
Stream	
<i>lecture02</i>	p. 9
Stream redirection	
<i>lecture03</i>	p. 24
<b>strerror()</b>	
<i>lecture03</i>	p. 1
String	
<i>lecture01</i>	p. 10,14,15
String array	
<i>lecture03</i>	p. 12,13
String comparison	
<i>lecture02</i>	p. 12,13
String conversion to number	
<i>lecture03</i>	p. 1
String declaration	
<i>lecture03</i>	p. 11
String search	
<i>lecture02</i>	p. 13
<b>string.h</b>	
<i>lecture02</i>	p. 11-13
Strings	

<i>lecture02</i>	p. 11-13
<i>lecture03</i>	p. 1
<b>strlen()</b>	
<i>lecture02</i>	p. 11
<b>strncasecmp()</b>	
<i>lecture02</i>	p. 13
<b>strncat()</b>	
<i>lecture02</i>	p. 12
<b>strncmp()</b>	
<i>lecture02</i>	p. 12,13
<b>strncpy()</b>	
<i>lecture02</i>	p. 12
<b>strrchr()</b>	
<i>lecture02</i>	p. 13
<b>strsep()</b>	
<i>lecture02</i>	p. 14
<b>strstr()</b>	
<i>lecture02</i>	p. 13
<b>strtod()</b>	
<i>lecture03</i>	p. 1
<b>strtok()</b>	
<i>lecture02</i>	p. 13,14
<b>strtol()</b>	
<i>lecture03</i>	p. 1
Struct	
<i>lecture03</i>	p. 20
<b>struct</b>	
<i>lecture03</i>	p. 15,17,20,23
<i>lecture03</i>	p. 16-20
<b>struct tm</b>	
<i>lecture03</i>	p. 21
Structure alignment	
<i>lecture03</i>	p. 18
Structure and pointer	
<i>lecture03</i>	p. 19,20
Structure initialization	
<i>lecture03</i>	p. 16
Structure naming	
<i>lecture03</i>	p. 16,17
Structures	
<i>lecture03</i>	p. 15-20
<b>switch</b>	
<i>lecture01</i>	p. 24

## T

Thomson (Ken)	
<i>lecture01</i>	p. 7
Thomson, Ken	
<i>lecture01</i>	p. 6
Time functions	
<i>lecture03</i>	p. 2,20,21
<b>time()</b>	
<i>lecture03</i>	p. 2,21
<b>time.h</b>	
<i>lecture03</i>	p. 21
<i>lecture03</i>	p. 2,21
<b>timegm()</b>	
<i>lecture03</i>	p. 21
<b>time_t</b>	
<i>lecture03</i>	p. 21
<i>lecture03</i>	p. 2
Tokenizing	
<i>lecture02</i>	p. 13,14
<b>tolower()</b>	
<i>lecture02</i>	p. 11
<b>toupper()</b>	
<i>lecture02</i>	p. 11
<b>typedef</b>	
<i>lecture03</i>	p. 17

## U

Unicode	
<i>lecture02</i>	p. 15,17,18
<b>union</b>	
<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
<b>unsigned</b>	
<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

<b>wchar</b>	
<i>lecture02</i>	p. 14,15
<b>while</b>	
<i>lecture02</i>	p. 1
Wide char	
<i>lecture02</i>	p. 14,15

## X

Xcode	
<i>lecture01</i>	p. 7