

!= (Logical operator not)
lecture01 p. 23

" (Double quote)
lecture01 p. 15

#define
lecture01 p. 16,18
lecture09 p. 3,4

#ifndef
lecture09 p. 3

#include
lecture01 p. 16,20
lecture02 p. 7
lecture09 p. 3

%p
lecture09 p. 22

& (bit and)
lecture01 p. 23

& operator (address)
lecture01 p. 10
lecture03 p. 4,6,8

&& (Logical operator and)
lecture01 p. 23

' (Single quote)
lecture01 p. 15

* operator (dereferencing)
lecture01 p. 10
lecture03 p. 7

- (arrow) reference to structure
lecture03 p. 19,20

. (dot) reference to structure
lecture03 p. 16,20

.h File
lecture01 p. 16

.hpp file
lecture08 p. 14

2-3-4 Tree
lecture06 p. 20

\0
lecture01 p. 14,15
lecture02 p. 10-12

\n
lecture02 p. 10

__FILE__
lecture09 p. 5,6

__LINE__
lecture09 p. 5,6

((Curly brackets)
lecture01 p. 22

| (bit or)
lecture01 p. 23

|| (Logical operator or)
lecture01 p. 23

A

accept()
lecture11 p. 4

Address
lecture01 p. 8-10,12,13

Address of variable
lecture03 p. 4

Adelson-Velsky, Georgy
lecture06 p. 17

Algorithm
lecture01 p. 12

Alignment of structures
lecture03 p. 18

And (logical operator)
lecture01 p. 23

Angle brackets vs double quotes for header files
lecture02 p. 7

Architecture
lecture01 p. 10

argc
lecture02 p. 2
lecture03 p. 14,15
lecture03 p. 13

Argument passed as reference
lecture08 p. 16

Argument passed by reference
lecture08 p. 17

argv[]
lecture02 p. 2
lecture03 p. 13-15

Array

<i>lecture01</i>	p. 10,12-14
<i>lecture03</i>	p. 6
<i>lecture05</i>	p. 17-21
<i>lecture06</i>	p. 1,9,10
<i>lecture08</i>	p. 20
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
Array:Pointer	
<i>lecture05</i>	p. 2
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
Assignment operator	
<i>lecture10</i>	p. 18-20
ATK	
<i>lecture09</i>	p. 18
atof()	
<i>lecture03</i>	p. 1
atoi()	
<i>lecture03</i>	p. 1
atol()	
<i>lecture03</i>	p. 1
autoconf	
<i>lecture09</i>	p. 11
automake	

<i>lecture09</i>	p. 11
autoscan	
<i>lecture09</i>	p. 10
Autotools	
<i>lecture09</i>	p. 9-11
AVL tree	
<i>lecture06</i>	p. 17-19

B

B-Tree	
<i>lecture06</i>	p. 20-23
<i>lecture07</i>	p. 2
Balanced tree	
<i>lecture06</i>	p. 17-19
Beeper program	
<i>lecture11</i>	p. 16
Bell Labs	
<i>lecture01</i>	p. 6,7
Berkeley Software Distribution (BSD)	
<i>lecture07</i>	p. 2
Berners-Lee, Tim	
<i>lecture11</i>	p. 7
Binary file	
<i>lecture04</i>	p. 2
Binary search	
<i>lecture06</i>	p. 9,10,13,14
Binary tree	
<i>lecture06</i>	p. 14-17
bind()	
<i>lecture11</i>	p. 4
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
BSD (Berkeley Software Distribution)	
<i>lecture07</i>	p. 2
<i>lecture10</i>	p. 22
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
<i>lecture05</i>	p. 1,2,16
<i>lecture06</i>	p. 23
<i>lecture09</i>	p. 15-17
C++	
<i>lecture08</i>	p. 10-13
C++ constructor	
<i>lecture10</i>	p. 4
C++ initialization	
<i>lecture10</i>	p. 4
C++ vs Java	
<i>lecture08</i>	p. 18-20
<i>lecture10</i>	p. 13,14
C11	
<i>lecture01</i>	p. 7
C89	
<i>lecture01</i>	p. 7
C99	
<i>lecture01</i>	p. 7
Cairo	
<i>lecture09</i>	p. 18
Calling functions	
<i>lecture04</i>	p. 8

calloc()	
<i>lecture04</i>	p. 18
Canonical class	
<i>lecture10</i>	p. 6,7,9,10,15,18,19
case	
<i>lecture01</i>	p. 24
Case	
<i>lecture02</i>	p. 11
Case insensitive comparison	
<i>lecture02</i>	p. 13
catch	
<i>lecture08</i>	p. 16
Catching errors	
<i>lecture08</i>	p. 11,12
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
cin	
<i>lecture08</i>	p. 15
CJK	
<i>lecture02</i>	p. 15
class	
<i>lecture08</i>	p. 18
<i>lecture09</i>	p. 15
Class (canonical)	
<i>lecture10</i>	p. 6,7,9,10,15,18,19
Class naming rules	
<i>lecture10</i>	p. 3,4
Classes	
<i>lecture08</i>	p. 17,18
<i>lecture10</i>	p. 3
Classification of characters	

<i>lecture02</i>	p. 10,11
close()	
<i>lecture11</i>	p. 3
Code	
<i>lecture01</i>	p. 8
codepoint	
<i>lecture02</i>	p. 15
<i>lecture02</i>	p. 15
Collection	
<i>lecture08</i>	p. 19
Collections	
<i>lecture05</i>	p. 16
Command-line parameters	
<i>lecture02</i>	p. 2
<i>lecture03</i>	p. 13-15
Comparison of Data structures	
<i>lecture07</i>	p. 3-5
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
Conditional compiling	
<i>lecture09</i>	p. 6,9
<i>lecture10</i>	p. 1
configure	
<i>lecture09</i>	p. 10,11
conio.h	
<i>lecture09</i>	p. 8
connect()	
<i>lecture11</i>	p. 2-4
Constants	
<i>lecture01</i>	p. 16,18
Constructor	
<i>lecture08</i>	p. 18
Constructor (copy)	
<i>lecture10</i>	p. 10-12,14,15

Constructor (default)	
<i>lecture10</i>	p. 7-9
Coplien, Jim	
<i>lecture10</i>	p. 6,7,9,10,15,18,19
Copy (shallow vs deep)	
<i>lecture10</i>	p. 14
Copy constructor	
<i>lecture10</i>	p. 10-12,14,15
Copy operator	
<i>lecture10</i>	p. 19,20
Core dump	
<i>lecture11</i>	p. 14
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
cout	
<i>lecture08</i>	p. 14,15
Craftsmanship	
<i>lecture01</i>	p. 5
Cryptography	
<i>lecture06</i>	p. 12
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

Dahl, Ole-Johan	
<i>lecture08</i>	p. 12
Data	
<i>lecture01</i>	p. 8
<i>lecture05</i>	p. 16,17
Data structure	

<i>lecture05</i>	p. 17
Data structure functions	
<i>lecture06</i>	p. 23
<i>lecture07</i>	p. 2,3
Data structures	
<i>lecture05</i>	p. 16,17,21,22
<i>lecture06</i>	p. 1-13
<i>lecture07</i>	p. 4,5
Data structures comparison	
<i>lecture07</i>	p. 3-5
Data types	
<i>lecture01</i>	p. 10-12
Database	
<i>lecture06</i>	p. 20,23
<i>lecture07</i>	p. 5
ddd	
<i>lecture09</i>	p. 22
Debugging	
<i>lecture09</i>	p. 22
Declaration	
<i>lecture05</i>	p. 1
Declaration of pointer	
<i>lecture03</i>	p. 5,6
Declaration of variable	
<i>lecture01</i>	p. 9
Deep copy	
<i>lecture10</i>	p. 14,15
Default constructor	
<i>lecture10</i>	p. 7-9
Default destructor	
<i>lecture10</i>	p. 10
Default parameters	
<i>lecture08</i>	p. 16
Degenarated binary tree	
<i>lecture06</i>	p. 17
delete	
<i>lecture08</i>	p. 15
Deleting a file	
<i>lecture04</i>	p. 2
Dereferencing	
<i>lecture03</i>	p. 7,8,19,20
Destructor	
<i>lecture08</i>	p. 18
Destructor (default)	
<i>lecture10</i>	p. 10

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
Doubly linked list	
<i>lecture06</i>	p. 8
Dumping a binary file	
<i>lecture04</i>	p. 3
Dynamic analysis:gdb	
<i>lecture09</i>	p. 22
Dynamic analysis:Valgrind	
<i>lecture09</i>	p. 22
Dynamic data structures	
<i>lecture07</i>	p. 5
Dynamic memory	
<i>lecture04</i>	p. 16-18
<i>lecture05</i>	p. 1,2,17-20
Dynamic memory example	
<i>lecture04</i>	p. 19

E

Eclipse	
<i>lecture09</i>	p. 22
EDP	
<i>lecture05</i>	p. 16
Electric-Fence	
<i>lecture09</i>	p. 22
Electronic Data Processing	

<i>lecture05</i>	p. 16
else	
<i>lecture01</i>	p. 21,23
else if	
<i>lecture01</i>	p. 23
Encapsulation	
<i>lecture08</i>	p. 18
<i>lecture10</i>	p. 3
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
<i>lecture09</i>	p. 15
errno.h	
<i>lecture03</i>	p. 1
<i>lecture09</i>	p. 15
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
Example: linked list	
<i>lecture06</i>	p. 4-6
Exams	
<i>lecture01</i>	p. 2-4
Exception	
<i>lecture02</i>	p. 4,5
Exceptions	
<i>lecture08</i>	p. 16
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
<i>lecture09</i>	p. 14
F	
Factorial	
<i>lecture05</i>	p. 14
fclose()	
<i>lecture03</i>	p. 25
fepf()	
<i>lecture04</i>	p. 1
ferror()	
<i>lecture04</i>	p. 1
fflush()	
<i>lecture09</i>	p. 22
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
FIFO	
<i>lecture06</i>	p. 8
<i>lecture07</i>	p. 4
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

First In First Out			
<i>lecture06</i>	p. 8	<i>lecture02</i>	p. 6,7
float		Function identification	
<i>lecture01</i>	p. 12	<i>lecture02</i>	p. 5,6
flock()		Function nesting	
<i>lecture04</i>	p. 2	<i>lecture01</i>	p. 15
Flow control		Function pointer	
<i>lecture01</i>	p. 21,23,24	<i>lecture06</i>	p. 23,24
<i>lecture02</i>	p. 1	Function pointers	
fopen()		<i>lecture09</i>	p. 16,17
<i>lecture03</i>	p. 25,26	Function prototype	
<i>lecture10</i>	p. 21	<i>lecture01</i>	p. 16
for		<i>lecture02</i>	p. 7
<i>lecture02</i>	p. 1	Function vs method	
Formatted input and output		<i>lecture10</i>	p. 16
<i>lecture02</i>	p. 10	Function: Pointers as argument	
fprint()		<i>lecture04</i>	p. 13,14
<i>lecture03</i>	p. 26	<i>lecture05</i>	p. 2,3
fprintf()		Function: returning an array	
<i>lecture02</i>	p. 10	<i>lecture04</i>	p. 11-13
fputc()		Functions	
<i>lecture02</i>	p. 9	<i>lecture04</i>	p. 8
<i>lecture04</i>	p. 1	Functions, nesting	
fputs()		<i>lecture02</i>	p. 6
<i>lecture02</i>	p. 10	fwrite()	
<i>lecture03</i>	p. 26	<i>lecture04</i>	p. 1
fread()			
<i>lecture04</i>	p. 1		
Free Software Foundation (FSF)			
<i>lecture09</i>	p. 9		
free()			
<i>lecture04</i>	p. 19		
<i>lecture04</i>	p. 18,21-23		
Freeing a binary tree			
<i>lecture06</i>	p. 16		
friend			
<i>lecture10</i>	p. 18		
fseek()			
<i>lecture04</i>	p. 2		
FSF			
<i>lecture09</i>	p. 9		
Function call			
<i>lecture04</i>	p. 8-12		
<i>lecture10</i>	p. 2		
Function declaration			

G

g++			
<i>lecture08</i>	p. 15		
Garbage collector			
<i>lecture04</i>	p. 19		
<i>lecture08</i>	p. 19		
Gateway			
<i>lecture10</i>	p. 25		
gcc			
<i>lecture01</i>	p. 17		
<i>lecture01</i>	p. 7		
gcd()			
<i>lecture04</i>	p. 8,9		
getaddrinfo()			
<i>lecture11</i>	p. 3,8		
getchar()			
<i>lecture02</i>	p. 9		

getopt()	
<i>lecture03</i>	p. 15
getpid()	
<i>lecture11</i>	p. 11
getppid()	
<i>lecture11</i>	p. 11
gets()	
<i>lecture02</i>	p. 10
Git	
<i>lecture09</i>	p. 21
Glib	
<i>lecture07</i>	p. 3
<i>lecture09</i>	p. 18
Global variable	
<i>lecture03</i>	p. 1
<i>lecture04</i>	p. 15
<i>lecture09</i>	p. 13,15,16
gmtime()	
<i>lecture03</i>	p. 21
Gnome	
<i>lecture07</i>	p. 3
Gnome Tool Kit (GTK)	
<i>lecture09</i>	p. 18-20
GNU	
<i>lecture07</i>	p. 3
GNU autotools	
<i>lecture09</i>	p. 9-11
Grades	
<i>lecture01</i>	p. 4
GTK (Gnome Tool Kit)	
<i>lecture09</i>	p. 18-20
gtk.h	
<i>lecture09</i>	p. 18
GtkWidget	
<i>lecture09</i>	p. 18,19
GTK_WINDOW	
<i>lecture09</i>	p. 19

H

Hanoi (towers of)	
<i>lecture05</i>	p. 15
Hash function	
<i>lecture06</i>	p. 11,12

Hash table	
<i>lecture06</i>	p. 11-13
<i>lecture07</i>	p. 2
head	
<i>lecture04</i>	p. 3
Head of list	
<i>lecture06</i>	p. 1
Header file	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 7
<i>lecture09</i>	p. 1,2,15
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
Hoare, Antony	
<i>lecture05</i>	p. 6
Honesty	
<i>lecture01</i>	p. 5
HTTP	
<i>lecture11</i>	p. 7-10
HTTPCnx	
<i>lecture11</i>	p. 10
httpd	
<i>lecture11</i>	p. 7

I

if	
<i>lecture01</i>	p. 21,23
In-memory database	
<i>lecture07</i>	p. 5
Information	
<i>lecture05</i>	p. 16,17
Information Technology	
<i>lecture05</i>	p. 16
Initialization of pointer	
<i>lecture03</i>	p. 7,8
Initialization of structure	
<i>lecture03</i>	p. 16
Input/Output	

lecture02 p. 9,10
Insertion in a binary tree
lecture06 p. 15,16

int

lecture01 p. 11

integer operations

lecture01 p. 11

iostream

lecture08 p. 14

isalnum()

lecture02 p. 11

isalpha()

lecture02 p. 11

isdigit()

lecture02 p. 11

islower()

lecture02 p. 11

ISO

lecture02 p. 16

isprint()

lecture02 p. 11

ispunct()

lecture02 p. 11

isspace()

lecture02 p. 11

isupper()

lecture02 p. 11

IT

lecture05 p. 16

J

Java vs C

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

lecture02 p. 5

lecture03 p. 12,20

lecture04 p. 7

java vs C

lecture04 p. 17

Java vs C

lecture05 p. 1,2,16

lecture06 p. 23

lecture09 p. 15-17

Java vs C++

lecture08 p. 18-20
lecture10 p. 13,14

K

K&R

lecture01 p. 6

Keringhan (Brian)

lecture01 p. 6

kill()

lecture11 p. 13

lecture11 p. 15

L

Lab2 hints

lecture05 p. 3-5

Labs

lecture01 p. 3,4

Landis, Evgenii

lecture06 p. 17

Last In First Out

lecture06 p. 7,8

ld

lecture01 p. 20

Library file

lecture04 p. 6

LIFO

lecture06 p. 7,8

lecture07 p. 4

Linked list

lecture05 p. 22

lecture06 p. 1-10,13

lecture07 p. 1

Linker

lecture01 p. 16,17,19,20

lecture04 p. 7-9

Linux

lecture01 p. 2

List

lecture08 p. 19

listen()

lecture11 p. 4

Listener	
<i>lecture10</i>	p. 25
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	
MAC address	
<i>lecture10</i>	p. 27
Macro	
<i>lecture09</i>	p. 4,5
main()	
<i>lecture01</i>	p. 16
make	
<i>lecture01</i>	p. 7
<i>lecture04</i>	p. 5,6
<i>lecture04</i>	p. 4-6
<i>lecture09</i>	p. 9
Makefile	
<i>lecture04</i>	p. 5,6
malloc()	
<i>lecture04</i>	p. 18-20
<i>lecture05</i>	p. 20
<i>lecture09</i>	p. 16
man	
<i>lecture02</i>	p. 5
<i>lecture10</i>	p. 20
<i>lecture10</i>	p. 21
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
Mathematical Induction	

<i>lecture05</i>	p. 10
Mathematical induction	
<i>lecture05</i>	p. 8-10
Matrix example	
<i>lecture09</i>	p. 2,3
Maurolico, Francisco	
<i>lecture05</i>	p. 9
MD5	
<i>lecture06</i>	p. 12
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
Mercurial	
<i>lecture09</i>	p. 21
Message nesting	
<i>lecture10</i>	p. 27
Method	
<i>lecture06</i>	p. 24
Method definition	
<i>lecture08</i>	p. 18
Method vs function	
<i>lecture10</i>	p. 16
Methods	
<i>lecture08</i>	p. 17
<i>lecture09</i>	p. 1
Methods in structures	
<i>lecture08</i>	p. 18
Midcourse exam	
<i>lecture01</i>	p. 3
MidCourse Exam	
<i>lecture08</i>	p. 1-10
Mixing C++ and C	
<i>lecture10</i>	p. 1,2
mktime()	
<i>lecture03</i>	p. 21
Multi-threading	
<i>lecture04</i>	p. 15,16
Multidimensional array	
<i>lecture03</i>	p. 13
Multiple inclusions	
<i>lecture09</i>	p. 3

Multithreading

lecture11 p. 6

N

Name of variable

lecture01 p. 9

namespace

lecture08 p. 14

Naming a structure

lecture03 p. 16,17

Naming of classes, members and methods

lecture10 p. 3,4

Nesting functions

lecture02 p. 6

Network programming

lecture10 p. 22-28

lecture11 p. 1-6

Networks

lecture10 p. 27,28

new

lecture08 p. 15

lecture09 p. 16

nm

lecture09 p. 14

Node

lecture05 p. 21,22

Non binary tree

lecture06 p. 20-23

Normal distribution

lecture03 p. 3

Not (logical operator)

lecture01 p. 23

NULL

lecture02 p. 10,13

lecture03 p. 1,7

Nygaard, Kristen

lecture08 p. 12

Nygaard, Kirsten

lecture08 p. 12

O

Object

lecture08 p. 19

Object creation/destruction

lecture10 p. 4-6

Object Oriented Programming

lecture09 p. 17

Object reference

lecture08 p. 19

Object-Oriented Programming

lecture06 p. 24

od

lecture04 p. 3

Operating system

lecture10 p. 20,21

operator

lecture08 p. 14,15

Operator (assignment)

lecture10 p. 18-20

Operator (copy)

lecture10 p. 19,20

Operator as function

lecture10 p. 17-19

Operator as method

lecture10 p. 17-19

Operator overloading

lecture10 p. 15,16

Or (logical operator)

lecture01 p. 23

Order

lecture05 p. 20,21

lecture07 p. 5

ostream

lecture10 p. 17

Output overloading

lecture10 p. 17

Over-engineering

lecture07 p. 5

Overflow

lecture02 p. 12

Overloading

lecture02 p. 5

lecture08 p. 16

lecture10 p. 2

Overloading output operator

lecture10 p. 17

P

Pango	
<i>lecture09</i>	p. 18
Parent process	
<i>lecture11</i>	p. 12
Pascal, Blaise	
<i>lecture05</i>	p. 9
perror()	
<i>lecture03</i>	p. 1
Persistence	
<i>lecture07</i>	p. 6
pid_t	
<i>lecture11</i>	p. 11
Pipe	
<i>lecture02</i>	p. 9
Pivot	
<i>lecture05</i>	p. 6-8
Pointer	
<i>lecture01</i>	p. 10
<i>lecture03</i>	p. 4-8,19,20
<i>lecture05</i>	p. 1,2
Pointer arithmetic	
<i>lecture03</i>	p. 10,11
Pointer on a function	
<i>lecture06</i>	p. 23,24
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
<i>lecture05</i>	p. 2,3
Pointers as parameters	
<i>lecture06</i>	p. 2,3
Pointers to functions	
<i>lecture09</i>	p. 16,17
Port	
<i>lecture10</i>	p. 25,26

<i>lecture11</i>	p. 4
Portability	
<i>lecture09</i>	p. 6-9
pptx	
<i>lecture04</i>	p. 3
Preprocessor	
<i>lecture01</i>	p. 16-18,20
<i>lecture09</i>	p. 3-6,8,9
printf()	
<i>lecture02</i>	p. 8,10
Priorities	
<i>lecture06</i>	p. 8
Process	
<i>lecture10</i>	p. 20,21
<i>lecture11</i>	p. 11,12
Process id	
<i>lecture11</i>	p. 11
Project	
<i>lecture09</i>	p. 1
Protocol	
<i>lecture10</i>	p. 25
<i>lecture11</i>	p. 7
Prototype (function)	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 7
ps	
<i>lecture11</i>	p. 12,13
public	
<i>lecture08</i>	p. 18
putchar()	
<i>lecture02</i>	p. 9
puts()	
<i>lecture02</i>	p. 10

Q

Quality	
<i>lecture01</i>	p. 5
Quick-sort	
<i>lecture05</i>	p. 6-8,11-14
Quiz 1	
<i>lecture07</i>	p. 6-8

R

Radix	
<i>lecture01</i>	p. 12
random()	
<i>lecture03</i>	p. 2,3
read()	
<i>lecture11</i>	p. 3,5
Reading ZIP or XML	
<i>lecture04</i>	p. 3
realloc()	
<i>lecture04</i>	p. 18
<i>lecture05</i>	p. 19,20
Recursion	
<i>lecture05</i>	p. 10-15
<i>lecture06</i>	p. 5,6
Recursion vs loops	
<i>lecture05</i>	p. 14
recv()	
<i>lecture11</i>	p. 3-5
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
Root	
<i>lecture06</i>	p. 14
Rounding error	
<i>lecture01</i>	p. 12
Router	
<i>lecture10</i>	p. 25

S

scanf()	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 3,4,10
<i>lecture04</i>	p. 14
SCCS	
<i>lecture09</i>	p. 21
Schedule	
<i>lecture01</i>	p. 1
Search	
<i>lecture06</i>	p. 9,10
<i>lecture07</i>	p. 5
search.h	
<i>lecture07</i>	p. 2
Self-managing list	
<i>lecture06</i>	p. 8
Semi-colon	
<i>lecture01</i>	p. 15
send()	
<i>lecture11</i>	p. 5
<i>lecture11</i>	p. 3,4
Serve	
<i>lecture11</i>	p. 4
Server	
<i>lecture11</i>	p. 4
setlocale	
<i>lecture02</i>	p. 15
setlocale()	
<i>lecture03</i>	p. 3
SHA1	
<i>lecture06</i>	p. 12
Shallow copy	
<i>lecture10</i>	p. 14
Shared library	
<i>lecture04</i>	p. 7
short	
<i>lecture01</i>	p. 11
Side-effects	
<i>lecture09</i>	p. 5
sigaction()	
<i>lecture11</i>	p. 16
SIGKILL	
<i>lecture11</i>	p. 15
Signal handler	
<i>lecture11</i>	p. 15
signal()	

<i>lecture11</i>	p. 15	Static variable	
<i>lecture11</i>	p. 16	<i>lecture04</i>	p. 16
signal.h		std	
<i>lecture11</i>	p. 13,15	<i>lecture08</i>	p. 14
Signals		stderr	
<i>lecture11</i>	p. 13,14	<i>lecture02</i>	p. 9
signed		<i>lecture02</i>	p. 9
<i>lecture01</i>	p. 11,12	<i>lecture03</i>	p. 1
SIGSTOP		<i>lecture10</i>	p. 1
<i>lecture11</i>	p. 15	stdin	
sig_t		<i>lecture02</i>	p. 9
<i>lecture11</i>	p. 15	<i>lecture02</i>	p. 9,10
Simula		<i>lecture03</i>	p. 24,25
<i>lecture08</i>	p. 12	stdio.h	
Single quote		<i>lecture03</i>	p. 25
<i>lecture01</i>	p. 15	stdlib.h	
sizeof()		<i>lecture03</i>	p. 1
<i>lecture03</i>	p. 6,13	<i>lecture04</i>	p. 18
Socket		<i>lecture11</i>	p. 12
<i>lecture11</i>	p. 1-3	stdout	
socket()		<i>lecture02</i>	p. 9,10
<i>lecture11</i>	p. 2	<i>lecture02</i>	p. 9,10
Sorting		<i>lecture03</i>	p. 24,25
<i>lecture05</i>	p. 6-8,11-14,19	<i>lecture10</i>	p. 1
Source control		Strategy	
<i>lecture09</i>	p. 20,21	<i>lecture06</i>	p. 7,8
Splitting code		strcasecmp()	
<i>lecture09</i>	p. 11,12	<i>lecture02</i>	p. 13
sscanf()		strcat()	
<i>lecture01</i>	p. 16	<i>lecture02</i>	p. 12
Stack		strchr()	
<i>lecture01</i>	p. 8	<i>lecture02</i>	p. 13
<i>lecture04</i>	p. 9-12	strcmp()	
Stallman, Richard		<i>lecture02</i>	p. 12,13
<i>lecture09</i>	p. 9	strcpy()	
Standard C++ library		<i>lecture02</i>	p. 12
<i>lecture08</i>	p. 15	strdup()	
static		<i>lecture04</i>	p. 18
<i>lecture04</i>	p. 7,16	<i>lecture05</i>	p. 18
<i>lecture09</i>	p. 14	Stream	
<i>lecture09</i>	p. 13,14	<i>lecture02</i>	p. 9
Static analysis:oclint		Stream redirection	
<i>lecture09</i>	p. 22	<i>lecture03</i>	p. 24
Static function		strerror()	
<i>lecture09</i>	p. 14	<i>lecture03</i>	p. 1

String	
<i>lecture01</i>	p. 10,14,15
string	
<i>lecture08</i>	p. 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
string.h	
<i>lecture02</i>	p. 11-13
<i>lecture04</i>	p. 18
Strings	
<i>lecture02</i>	p. 11-13
<i>lecture03</i>	p. 1
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
Stroustrup, Bjarne	
<i>lecture08</i>	p. 11-14
<i>lecture10</i>	p. 13
strchr()	
<i>lecture02</i>	p. 13
strsep()	
<i>lecture02</i>	p. 14
strstr()	
<i>lecture02</i>	p. 13
strtod()	
<i>lecture03</i>	p. 1
strtok()	
<i>lecture02</i>	p. 13,14
strtol()	
<i>lecture03</i>	p. 1

Struct	
<i>lecture03</i>	p. 20
struct	
<i>lecture03</i>	p. 15,17,20,23
<i>lecture03</i>	p. 16-20
<i>lecture05</i>	p. 17
<i>lecture09</i>	p. 2
struct (C++)	
<i>lecture08</i>	p. 17
struct addrinfo	
<i>lecture11</i>	p. 2,3
struct tm	
<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
Subversion	
<i>lecture09</i>	p. 21
switch	
<i>lecture01</i>	p. 24
System call	
<i>lecture10</i>	p. 21
System calls	
<i>lecture10</i>	p. 20
System V	
<i>lecture10</i>	p. 22
system()	
<i>lecture11</i>	p. 12,13

T

Tail pointer	
<i>lecture06</i>	p. 7,8
TCP	
<i>lecture11</i>	p. 7
TCP/IP	
<i>lecture10</i>	p. 25,26

TCPAcceptor
lecture11 p. 4,5

TCPConnector
lecture11 p. 4,5,10

TCPStream
lecture11 p. 4,5,10

Testing
lecture10 p. 1

this
lecture10 p. 4

Thomson (Ken)
lecture01 p. 7

Thomson, Ken
lecture01 p. 6

Threads
lecture09 p. 16
lecture11 p. 6

throw
lecture08 p. 16

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

Tools
lecture09 p. 1

toupper()
lecture02 p. 11

Towers of Hanoi
lecture05 p. 15

Tree
lecture06 p. 13,20-23
lecture07 p. 1,2,4

try

lecture08 p. 16

typedef

lecture03 p. 17

lecture09 p. 2

U

Unicode
lecture02 p. 15,17,18

union
lecture03 p. 23
lecture03 p. 24

unistd.h
lecture09 p. 8

UNIX
lecture01 p. 6

Unix
lecture01 p. 7

Unix pipe
lecture02 p. 9

unlink()
lecture04 p. 2

unsigned
lecture01 p. 11,12

UTF-16
lecture02 p. 17

UTF-32
lecture02 p. 17

UTF-8
lecture02 p. 15,18

V

Variable declaration
lecture01 p. 9

Variable name
lecture01 p. 9

Variable number of parameters
lecture02 p. 6

vector
lecture08 p. 19,20

Virtual machine
lecture09 p. 7

Visual C++	
<i>lecture09</i>	p. 22
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

ZIP	
<i>lecture04</i>	p. 3

W

Walking a binary tree	
<i>lecture06</i>	p. 16
Wall gcc flag	
<i>lecture09</i>	p. 22
wchar	
<i>lecture02</i>	p. 14,15
while	
<i>lecture02</i>	p. 1
Wide char	
<i>lecture02</i>	p. 14,15
Wikipedia reference for operators	
<i>lecture10</i>	p. 15
wine	
<i>lecture09</i>	p. 7
write()	
<i>lecture11</i>	p. 3,5

X

Xcode	
<i>lecture01</i>	p. 7
<i>lecture09</i>	p. 22
XML	
<i>lecture04</i>	p. 3

Z