

!= (Logical operator not)	
<i>lecture01</i>	p. 23
" (Double quote)	
<i>lecture01</i>	p. 15
#define	
<i>lecture01</i>	p. 16,18
<i>lecture09</i>	p. 3,4
#ifndef	
<i>lecture09</i>	p. 3
#include	
<i>lecture01</i>	p. 16,20
<i>lecture02</i>	p. 7
<i>lecture09</i>	p. 3
%p	
<i>lecture09</i>	p. 22
& (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
.hpp file	
<i>lecture08</i>	p. 14
2-3-4 Tree	
<i>lecture06</i>	p. 20
\0	
<i>lecture01</i>	p. 14,15
<i>lecture02</i>	p. 10-12
\n	
<i>lecture02</i>	p. 10

__FILE__	
<i>lecture09</i>	p. 5,6
__LINE__	
<i>lecture09</i>	p. 5,6
((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
Adelson-Velsky, Georgy	
<i>lecture06</i>	p. 17
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
Argument passed as reference	
<i>lecture08</i>	p. 16
Argument passed by reference	
<i>lecture08</i>	p. 17
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

<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
atoi()	
<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
Bell Labs	
<i>lecture01</i>	p. 6,7
Berkeley Software Distribution (BSD)	
<i>lecture07</i>	p. 2
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
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		
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
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
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
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
<i>lecture10</i>	p. 21
for	
<i>lecture02</i>	p. 1

Formatted input and output

lecture02 p. 10

fprint()

lecture03 p. 26

fprintf()

lecture02 p. 10

fputc()

lecture02 p. 9

lecture04 p. 1

fputs()

lecture02 p. 10

lecture03 p. 26

fread()

lecture04 p. 1

Free Software Foundation (FSF)

lecture09 p. 9

free()

lecture04 p. 19

lecture04 p. 18,21-23

Freeing a binary tree

lecture06 p. 16

friend

lecture10 p. 18

fseek()

lecture04 p. 2

FSF

lecture09 p. 9

Function call

lecture04 p. 8-12

lecture10 p. 2

Function declaration

lecture02 p. 6,7

Function identification

lecture02 p. 5,6

Function nesting

lecture01 p. 15

Function pointer

lecture06 p. 23,24

Function pointers

lecture09 p. 16,17

Function prototype

lecture01 p. 16

lecture02 p. 7

Function vs method

lecture10 p. 16

Function: Pointers as argument

lecture04 p. 13,14

lecture05 p. 2,3

Function: returning an array

lecture04 p. 11-13

Functions

lecture04 p. 8

Functions, nesting

lecture02 p. 6

fwrite()

lecture04 p. 1

G

g++

lecture08 p. 15

Garbage collector

lecture04 p. 19

lecture08 p. 19

Gateway

lecture10 p. 25

gcc

lecture01 p. 17

lecture01 p. 7

gcd()

lecture04 p. 8,9

getchar()

lecture02 p. 9

getopt()

lecture03 p. 15

gets()

lecture02 p. 10

Git

lecture09 p. 21

Glib

lecture07 p. 3

lecture09 p. 18

Global variable

lecture03 p. 1

lecture04 p. 15

lecture09 p. 13,15,16

gmtime()

lecture03 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

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	
<i>lecture02</i>	p. 9,10
Insertion in a binary tree	
<i>lecture06</i>	p. 15,16
int	
<i>lecture01</i>	p. 11
integer operations	
<i>lecture01</i>	p. 11
iostream	
<i>lecture08</i>	p. 14
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()	

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
Kerighan (Brian)
lecture01 p. 6

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
Listener
lecture10 p. 25
localtime()
lecture03 p. 21
Locking a file
lecture04 p. 2
Logical operators
lecture01 p. 23
long
lecture01 p. 11
lecture01 p. 11
Loop
lecture02 p. 1

M

MAC address
lecture10 p. 27
Macro
lecture09 p. 4,5
main()
lecture01 p. 16
make
lecture01 p. 7
lecture04 p. 5,6
lecture04 p. 4-6
lecture09 p. 9
Makefile
lecture04 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

N

Name of variable	
<i>lecture01</i>	p. 9
namespace	
<i>lecture08</i>	p. 14
Naming a structure	
<i>lecture03</i>	p. 16,17
Naming of classes, members and methods	
<i>lecture10</i>	p. 3,4
Nesting functions	
<i>lecture02</i>	p. 6
Network programming	
<i>lecture10</i>	p. 22-28
Networks	
<i>lecture10</i>	p. 27,28
new	
<i>lecture08</i>	p. 15
<i>lecture09</i>	p. 16
nm	
<i>lecture09</i>	p. 14
Node	
<i>lecture05</i>	p. 21,22
Non binary tree	
<i>lecture06</i>	p. 20-23
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
Nygaard, Kristen	
<i>lecture08</i>	p. 12
Nygaard, Kirsten	
<i>lecture08</i>	p. 12

O

Object	
<i>lecture08</i>	p. 19
Object creation/destruction	
<i>lecture10</i>	p. 4-6
Object Oriented Programming	
<i>lecture09</i>	p. 17
Object reference	
<i>lecture08</i>	p. 19
Object-Oriented Programming	
<i>lecture06</i>	p. 24
od	
<i>lecture04</i>	p. 3
Operating system	
<i>lecture10</i>	p. 20,21
operator	
<i>lecture08</i>	p. 14,15
Operator (assignment)	
<i>lecture10</i>	p. 18-20
Operator (copy)	
<i>lecture10</i>	p. 19,20
Operator as function	
<i>lecture10</i>	p. 17-19
Operator as method	
<i>lecture10</i>	p. 17-19
Operator overloading	
<i>lecture10</i>	p. 15,16
Or (logical operator)	
<i>lecture01</i>	p. 23
Order	
<i>lecture05</i>	p. 20,21
<i>lecture07</i>	p. 5
ostream	
<i>lecture10</i>	p. 17
Output overloading	

<i>lecture10</i>	p. 17
Over-engineering	
<i>lecture07</i>	p. 5
Overflow	
<i>lecture02</i>	p. 12
Overloading	
<i>lecture02</i>	p. 5
<i>lecture08</i>	p. 16
<i>lecture10</i>	p. 2
Overloading output operator	
<i>lecture10</i>	p. 17

P

Pango	
<i>lecture09</i>	p. 18
Pascal, Blaise	
<i>lecture05</i>	p. 9
perror()	
<i>lecture03</i>	p. 1
Persistence	
<i>lecture07</i>	p. 6
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
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
Project	
<i>lecture09</i>	p. 1
Protocol	
<i>lecture10</i>	p. 25
Prototype (function)	
<i>lecture01</i>	p. 16
<i>lecture02</i>	p. 7
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
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
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>lecture04</i>	p. 9-12
<i>lecture09</i>	p. 21	Stallman, Richard	
Schedule		<i>lecture09</i>	p. 9
<i>lecture01</i>	p. 1	Standard C++ library	
Search		<i>lecture08</i>	p. 15
<i>lecture06</i>	p. 9,10	static	
<i>lecture07</i>	p. 5	<i>lecture04</i>	p. 7,16
search.h		<i>lecture09</i>	p. 14
<i>lecture07</i>	p. 2	<i>lecture09</i>	p. 13,14
Self-managing list		Static analysis:oclint	
<i>lecture06</i>	p. 8	<i>lecture09</i>	p. 22
Semi-colon		Static function	
<i>lecture01</i>	p. 15	<i>lecture09</i>	p. 14
setlocale		Static variable	
<i>lecture02</i>	p. 15	<i>lecture04</i>	p. 16
setlocale()		std	
<i>lecture03</i>	p. 3	<i>lecture08</i>	p. 14
SHA1		stderr	
<i>lecture06</i>	p. 12	<i>lecture02</i>	p. 9
Shallow copy		<i>lecture02</i>	p. 9
<i>lecture10</i>	p. 14	<i>lecture03</i>	p. 1
Shared library		<i>lecture10</i>	p. 1
<i>lecture04</i>	p. 7	stdin	
short		<i>lecture02</i>	p. 9
<i>lecture01</i>	p. 11	<i>lecture02</i>	p. 9,10
Side-effects		<i>lecture03</i>	p. 24,25
<i>lecture09</i>	p. 5	stdio.h	
signed		<i>lecture03</i>	p. 25
<i>lecture01</i>	p. 11,12	stdlib.h	
Simula		<i>lecture03</i>	p. 1
<i>lecture08</i>	p. 12	<i>lecture04</i>	p. 18
Single quote		stdout	
<i>lecture01</i>	p. 15	<i>lecture02</i>	p. 9,10
sizeof()		<i>lecture02</i>	p. 9,10
<i>lecture03</i>	p. 6,13	<i>lecture03</i>	p. 24,25
Sorting		<i>lecture10</i>	p. 1
<i>lecture05</i>	p. 6-8,11-14,19	Strategy	
Source control		<i>lecture06</i>	p. 7,8
<i>lecture09</i>	p. 20,21	strcasecmp()	
Splitting code		<i>lecture02</i>	p. 13
<i>lecture09</i>	p. 11,12	strcat()	
sscanf()		<i>lecture02</i>	p. 12
<i>lecture01</i>	p. 16	strchr()	
Stack		<i>lecture02</i>	p. 13
<i>lecture01</i>	p. 8	strcmp()	

<i>lecture02</i>	p. 12,13
strcpy()	
<i>lecture02</i>	p. 12
strdup()	
<i>lecture04</i>	p. 18
<i>lecture05</i>	p. 18
Stream	
<i>lecture02</i>	p. 9
Stream redirection	
<i>lecture03</i>	p. 24
strerror()	
<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

strrchr()	
<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 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

T

Tail pointer	
<i>lecture06</i>	p. 7,8
TCP/IP	
<i>lecture10</i>	p. 25,26
Testing	
<i>lecture10</i>	p. 1
this	
<i>lecture10</i>	p. 4
Thomson (Ken)	
<i>lecture01</i>	p. 7
Thomson, Ken	
<i>lecture01</i>	p. 6
Threads	
<i>lecture09</i>	p. 16
throw	
<i>lecture08</i>	p. 16
Time functions	
<i>lecture03</i>	p. 2,20,21
time()	
<i>lecture03</i>	p. 2,21
time.h	
<i>lecture03</i>	p. 21
<i>lecture03</i>	p. 2,21
timegm()	
<i>lecture03</i>	p. 21
time_t	
<i>lecture03</i>	p. 21
<i>lecture03</i>	p. 2
Tokenizing	
<i>lecture02</i>	p. 13,14
tolower()	
<i>lecture02</i>	p. 11
Tools	
<i>lecture09</i>	p. 1
toupper()	
<i>lecture02</i>	p. 11
Towers of Hanoi	
<i>lecture05</i>	p. 15
Tree	
<i>lecture06</i>	p. 13,20-23
<i>lecture07</i>	p. 1,2,4
try	

<i>lecture08</i>	p. 16
typedef	
<i>lecture03</i>	p. 17
<i>lecture09</i>	p. 2

U

Unicode	
<i>lecture02</i>	p. 15,17,18
union	
<i>lecture03</i>	p. 23
<i>lecture03</i>	p. 24
unistd.h	
<i>lecture09</i>	p. 8
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
vector	
<i>lecture08</i>	p. 19,20
Virtual machine	
<i>lecture09</i>	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

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

X

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

Z

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