

TABLE OF CONTENT

Topic	Page
CHAPTER ONE1
CONCEPTS AND PRINCIPLES OF A PROGRAMMING. . LANGUAGE	1
LEARNING OBJECTIVES1
PROGRAMMING LANGUAGES1
INTRODUCTION	2
MACHINE LANGUAGES.	4
ASSEMBLY LANGUAGE6
HIGH LEVEL PROGRAMMING LANGUAGES . . .	9
TRANSLATORS	10
HIGH LEVEL LANGUAGE TRANSLATORS11
HOW DOES A COMPILER WORK?11
STAGES DURING COMPILATION12
ANALYSIS OF THE SOURCE CODE13
SEMANTIC ANALYSIS15
THE SYMBOL TABLE MANAGER.15
PRE-PROCESSORS15
ASSEMBLERS16
LOADER AND LINK EDITOR16
DATA TYPES AND THEIR REPRESENTATION17

THE BINARY NUMBER SYSTEM17
REPRESENTING NEGATIVE NUMBERS18
FUNDAMENTAL DATA TYPES20
DATA STRUCTURES.23
ARRAYS23
MULTIDIMENSIONAL ARRAYS24
STRINGS AS CHARACTER ARRAYS24
RECORDS26
LISTS28
ONE-WAY LINKED LIST.	29
CREATING A LIST29
CIRCULAR LIST.30
TWO-WAY LINKED LIST32
QUEUES, STACKS AND SHELVES33
TREES36
LANGUAGE OPERATORS38
EXPRESSIONS38
ADDITIONAL OPERATORS39
LOCATION (l) AND REFERENCE (r) VALUES40
ASSESSMENT QUESTIONS AND EXERCISES41
CHAPTER TWO43
THE C PROGRAMMING LANGUAGE.43

LEARNING OBJECTIVES43
INTRODUCTION44
FUNDAMENTALS OF ANSI C45
FUNCTIONS45
THE FUNCTION <code>main()</code>45
PROGRAMMING EXERCISE 2-146
AN ABSTRACTED VIEW OF A C FUNCTION.47
SOURCE CODE LAYOUT50
PROGRAMMING EXERCISE 2-250
PRE-PROCESSOR DIRECTIVES51
(1) INCLUSION OF EXTERNAL FILES51
(2) MACRO DEFINITION AND REPLACEMENT51
(2.1) CONCATENATION52
(2.2) UNDEFINED MACROS52
(3) CONDITIONAL COMPILATION/INCLUSION.53
(4) COMPILATION ERROR MESSAGE54
(5) PROGRAM LINE CONTROL55
DECLARATION AND DATA TYPES57
HOW TO INPUT AND OUTPUT INFORMATION58
EDITED DATA OUTPUT58
PROGRAMMING EXERCISE 2-361
DATA INPUT63

PROGRAMMING EXERCISE 2-464
FORMATTED INPUT DATA64
DECLARING COMPLEX DATA TYPES IN C67
AN EXAMPLE PROGRAM PROBLEM68
THE PROGRAM SPECIFICATION68
OUTPUT FROM THE PROGRAM69
THE CONTROL LOGIC69
THE SOURCE PROGRAM70
ADDITIONAL PROGRAMMING EXERCISE 2-171
ADDITIONAL PROGRAMMING EXERCISE 2-272
ADDITIONAL PROGRAMMING EXERCISES 2-373
SUPPLEMENTARY PROGRAMMING EXERCISES75
CHAPTER THREE79
DATA TYPES AND LANGUAGE OPERATORS IN C79
LEARNING OBJECTIVES79
INTRODUCTION80
VARIABLES AND DATA TYPES81
FUNDAMENTAL DATA TYPES81
VARIABLE NAMES AND THEIR DECLARATION IN C82
DATA TYPE DECLARATION IN C83
INITIALISING VARIABLES84

PROGRAMMING EXERCISE 3-185
PROGRAMMING EXERCISE 3-286
TYPE MODIFIERS87
DECLARING VARIABLES IN A C PROGRAM89
THE USE OF VARIABLES IN A PROGRAM90
WORKING WITH NUMERIC DATA TYPES93
NUMERIC DATA TYPES93
THE SIGNED AND UNSIGNED MODIFIERS94
FLOATING-POINT NUMBERS.94
PROGRAMMING EXERCISE 3-395
ENUMERATED DATA TYPES96
TYPE CONVERSION97
COMPLEX DATA TYPES100
EXPRESSIONS AND OPERATORS100
HOW EXPRESSIONS ARE EVALUATED101
ANSI C LANGUAGE OPERATORS102
THE ASSIGNMENT OPERATOR104
THE ARITHMETIC OPERATORS105
INCREMENT AND DECREMENT OPERATORS . .	.106
PROGRAMMING EXERCISE 3-4.107
THE LOGICAL AND RELATIONAL OPERATORS. .	.107
BIT-WISE OPERATORS109
BIT SHIFTS110

BIT-WISE LOGICAL OPERATORS.111
PROGRAMMING EXERCISE 3-5.112
THE CONDITIONAL OPERATORS.112
THE COMMA OPERATOR.112
EXAMPLE PROGRAMMING PROBLEM113
PROGRAM SPECIFICATION113
BACKGROUND INFORMATION.113
INPUT TEST DATA114
OUTPUT114
THE PROGRAM LOGIC DESIGN115
PROGRAMMING EXERCISE 3-6117
ADDITIONAL PROGRAMMING EXERCISE 3-1118
ADDITIONAL PROGRAMMING EXERCISES 3-2.	.119
CHAPTER FOUR121
ARRAYS STRUCTURES AND POINTERS121
LEARNING OBJECTIVES121
INTRODUCTION123
ARRAYS IN C123
MULTIDIMENSIONAL ARRAYS124
POINTERS AND ARRAYS126
PROGRAMMING EXERCISE 4-1.128
NESTED POINTERS130

CHARACTER STRINGS132
PROGRAMMING EXERCISE 4-2133
PROGRAMMING EXERCISE 4-3135
STRUCTURES IN C.136
DECLARING STRUCTURES136
NESTED STRUCTURES137
THE STRUCTURE COMPONENT SELECT OPERATOR. .	.138
POINTERS TO STRUCTURES139
SELF REFERENCE140
THE TYPEDEF OPERATOR471
BIT FIELDS144
UNIONS146
AN EXAMPLE PROGRAMMING PROBLEM149
PROGRAMMING PROBLEM SPECIFICATION .	.149
THE INPUT DATA FILE149
THE PROGRAM LOGIC DESIGN150
THE PROGRAM CONTROL LOGIC151
THE PROGRAM SOURCE CODE151
ADDITIONAL PROGRAMMING EXERCISE 4-4 . .	.153
EXCERCISES ON ARRAY PROCESSING.154
PROGRAM PROBLEM SPECIFICATION154
THE MAIN CONTROL LOGIC155
SORTING THE DATA VALUES.156

ALGORITHM TO DETERMINE AVERAGE VALUES156
ALGORITHM TO DETERMINE SYMMETRICAL MATRIX	158
ANALYSIS AND QUESTIONNAIRE159
THE PROGRAM SOURCE CODE160
CHAPTER FIVE161
USING FUNCTIONS IN C161
LEARNING OBJECTIVES161
INTRODUCTION162
THE FUNCTION main().163
FUNCTION PROTOTYPE.164
PROGRAMMING EXERCISE 5-1.166
PROGRAMMING EXERCISE 5-2.167
COMMAND-LINE ARGUMENTS167
THE FUNCTION DEFINITION168
FUNCTION CALLS169
AN ABSTRACTED VIEW OF A C FUNCTION171
ACCESSING EXTERNAL VALUES VIA FUNCTION CALLS174
USING ARRAYS AS PARAMETER IN A FUNCTION CALL178
SCOPE AND DURATION OF FUNCTION VARIABLES180
(1) THE SCOPE OF A VARIABLE180
PROGRAMMING EXERCISE 5-3.181
(2) THE DURATION OF A VARIABLE183

AUTOMATIC VARIABLES184
STATIC VARIABLES.184
REGISTER VARIABLES185
POINTERS TO FUNCTIONS186
RECURSION.192
PROGRAMMING EXERCISE 5-4.194
USING C LIBRARY FUNCTIONS.196
PROGRAMMING EXERCISES 5-5197
EXAMPLE PROGRAMMING PROBLEMS197
PROGRAMMING ASSIGNMENT #1197
THE MAIN PROGRAM MENU.197
SUB-MENU: ENTER BOOK INFORMATION.198
SUB-MENU: BOOK REPORT BY TITLE198
SUB-MENU: BOOK REPORT BY AUTHORS199
INPUT DATA VALIDATION199
THE PROGRAM DESIGN199
ADDITIONAL EXERCISES199
THE PROGRAM CONTROL STRUCTURE200
MAIN PROGRAM HIERARCHY CHART200
MAIN CONTROL LOGIC200
CONTROL LOGIC TO DISPLAY THE MAIN MENU AND GET USER SELECTION203
CONTROL LOGIC: ENTER BOOK INFORMATION204
PROGRAMMING EXERCISE 5-6.206

CONTROL LOGIC: LIST BOOKS BY TITLE . . .	207
PROGRAMMING EXERCISE 5-7.209
LIST BOOKS BY AUTHORS209
CONTROL LOGIC: SORT BOOKS ON SELECTED KEY .	.210
PROGRAMMING PROBLEM #2211
BOTTOM-UP DESIGN TECHNIQUE ILLUSTRATED .	.211
PROGRAMMING PROBLEM #1211
PROGRAMMING PROBLEM #2211
PROGRAMMING PROBLEM #3211
PROGRAMMING PROBLEM #4212
PROGRAMMING PROBLEMS SOLUTIONS . .	.212
PROGRAMMING PROBLEM #1 SOLUTION212
GETTING THE PROGRAM TO LOOP213
INSERTING A VALIDATION FUNCTION216
THE ABSTRACTED CONTROL LOGIC217
PROGRAMMING EXERCISE 5-8.220
PROGRAMMING EXERCISE 5-9220
A MENU DRIVEN PROGRAM.221
MAIN CONTROL LOGIC221
PROGRAMMING ASSIGNMENT #2222
PROGRAM SCREEN DISPLAY222
THE MAIN PROGRAM MENU222
CODE 1- ENTER CLASS INFORMATION223
CODE 2-DISPLAY CLASS IN NAME SEQUENCE223

CODE 3 -DISPLAY CLASS LISTING IN TEACHER NAME SEQUENCE223
CODE 4-DO INQUIRY ON TEACHER NAME AND ENROLMENT		224
INPUT DATA VALIDATION224
CHAPTER SIX226
CONTROL STRUCTURES IN C.226
LEARNING OBJECTIVES226
INTRODUCTION.227
GROUPING ANSI C STATEMENTS.228
THE CASE STRUCTURE229
THE IF STATEMENT230
THE IF-ELSE STATEMENT232
PROGRAMMING EXERCISE 6-1.234
THE NESTED CASE STRUCTURE234
PROGRAMMING EXERCISE 6-2.239
PROGRAMMING EXERCISE 6-3.240
HOSPITAL BILLING SYSTEM240
CONTROL LOGIC.241
PROGRAM LOGIC, ASSIGNMENT #1243
PROGRAMMING EXERCISE 6-4.245
THE SWITCH STATEMENT246
GENERAL SYNTAX246

CONTROL LOOPING.248
LOOPING AND THE LOOP LOGIC STRUCTURE	.248
(1) INITIALISE THE LOOP CONTROL VARIABLE248
(2) TEST THE LOOP CONTROL VARIABLE249
(3) EXECUTE THE LOOP INSTRUCTIONS249
(4) MODIFY THE LOOP CONTROL VARIABLE249
(5) EXIT FROM THE LOOP249
LOOP SINGLE ENTRY/EXIT POINT PRINCIPLE250
THE WHILE LOOP250
THE FOR STATEMENT.252
GENERAL SYNTAX252
EXAMPLE CODE253
THE DO-WHILE LOOP255
WHEN TO USE WHICH LOOP STATEMENT258
BREAK AND CONTINUE259
GOTO AND LABELS259
THE RETURN COMMAND.260
RECURSION261
PROGRAMMING EXERCISE 6-5.261
AN EXAMPLE PROGRAMMING PROBLEM263
THE TOP-DOWN DESIGN AND STRING PROCESSING263
THE USER INSTRUCTION263
CREATING THE PERSONALISED LETTER264
CREATING THE MAILING LABELS265

VALIDATING INPUT DATA266
THE PROGRAM DESIGN267
THE PROGRAM MODULES HIERARCHY CHART	.268
THE PROGRAM CONTROL LOGIC269
MAIN CONTROL LOGIC269
SOURCE CODE FOR MAIN CONTROL FUNCTION .	.270
DISPLAY MAIN MENU-SOURCE CODE270
PROGRAMMING EXERCISE 6-5.272
DISPLAY USER INSTRUCTION - SOURCE CODE .	.273
CONTROL LOGIC-PREPARE AND PRINT THE LETTER .	.274
SOURCE CODE-PREPARE AND PRINT THE SALES LETTERS	.275
PROGRAMMING EXERCISE 6-6.275
CONTROL LOGIC-GET VALID USER DATA276
SOURCE CODE-GET VALID USER DATA277
PROGRAMMING EXERCISE 6-7.277
CONTROL LOGIC-VALIDATE USER INPUT DATA .	.278
SOURCE CODE-VALIDATE USER INPUT DATA . .	.280
PROGRAMMING EXERCISE 6-8.281
CONTROL LOGIC-PRINT THE LETTER282
SOURCE CODE-PRINT THE LETTER283
PROGRAMMING EXERCISE 6-9.283
CONTROL LOGIC-PRINT THE MAILING LABELS .	.284
PROGRAMMING EXERCISE 6-10285
CONTROL LOGIC-OBTAIN VALID DATE FROM USER .	.286

ADDITIONAL PROGRAMMING ASSIGNMENT287
THE MAIN PROGRAM MENU287
THE PROGRAM DESIGN288
THE PROGRAM HIERARCHY CHART289
THE MAIN PROGRAM LOGIC289
PRODUCE SALES LETTERS290
PROGRAMMING EXRCISE290
CHAPTER SEVEN291
INPUT/OUTPUT AND LIBRARY FUNCTIONS IN C .291	
LEARNING OBJECTIVES291
INTRODUCTION292
FILE HANDLING293
OPENING, USING AND CLOSING A C FILE294
STANDARD C LIBRARY FUNCTIONS297
CHARACTER HANDLING297
CHARACTER TESTING FUNCTIONS:298
CHARACTER CASE MAPPING FUNCTIONS298
LIMITS OF FLOAT TYPE299
SIZES OF INTEGRAL TYPES300
LOCALISATION300
MATHEMATICAL OPERATIONS301
INTEGER AND ABSOLUTE VALUE FUNCTIONS301

POWER FUNCTIONS302
EXPONENTIAL AND LOGARITHMIC FUNCTIONS .	.302
TRIGONOMETRIC FUNCTIONS303
HYPERBOLIC FUNCTIONS303
NON-LOCAL JUMPS304
SIGNAL HANDLING304
VARIABLE ARGUMENTS306
STANDARD DEFINITIONS307
INPUT AND OUTPUT OPERATIONS308
FUNCTIONS TO PERFORM OPERATIONS ON FILES.	.308
FILE ACCESS FUNCTIONS309
INPUT AND OUTPUT FORMATTING FUNCTIONS .	.310
FILE POSITIONING FUNCTIONS311
THE STANDARD LIBRARY312
PSEUDO-RANDOM NUMBER FUNCTIONS313
MEMORY MANAGEMENT FUNCTIONS314
ENVIRONMENT COMMUNICATION FUNCTIONS .	.314
SEARCHING AND SORTING UTILITIES315
INTEGER ARITHMETIC FUNCTIONS315
STRING HANDLING FUNCTIONS315
STRING COPY FUNCTIONS316
STRING CONCATENATION FUNCTIONS316
STRING COMPARISON FUNCTIONS316
STRING SEARCH FUNCTIONS317

MISCELLANEOUS STRING FUNCTIONS318
DATE AND TIME MANIPULATING FUNCTIONS318
TIME MANIPULATION FUNCTIONS318
TIME CONVERSION FUNCTIONS319
PROGRAMMING EXERCISE 7-1.321
PERFORMING OPERATIONS ON MATRICES321
MATRIX ARITHMETIC OPERATIONS321
MATRIX ALGEBRAIC OPERATIONS322
AN EXAMPLE PROGRAMMING PROBLEM324
FILE PROCESSING AND INPUT/OUTPUT OPERATIONS324
FILES AND REPORT GENERATION.325
THE STOCK PURCHASE RECORD325
SEQUENTIAL FILES.325
WRITING DATA ON A SECONDARY STORAGE MEDIUM326
READING DATA FROM SECONDARY MEDIUM.329
THE CONTROL BREAK LOGIC330
PROGRAMMING EXERCISE 7-2.333

INDEX OF TERMS

- abort, 298, 306, 315
- abstraction, 2, 3, 41-47, 89, 125, 163, 164, 218, 219
- accumulation, 71, 157, 252
- address, 6, 13, 16, 23-34, 48, 63-65, 73, 81-85, 94, 95, 103, 124-147, 173-198, 264, 267, 289, 314, 316
- address bus, 81
- address calculation, 24
- address format, 6
- address reference, 64
- address value, 63, 94, 128, 145, 187
- alternative processing, 229, 231
- ANSI, 9, 42-46, 54, 66, 88-102, 141, 142, 155, 165, 170, 230, 292, 306, 326
- arguments, 15, 29, 63, 64, 92, 93, 112, 168-178, 189, 254, 304-315
- array, 15, 23-29, 33-38, 42, 65-81, 102, 112, 122-168, 179, 180, 191-210, 252-266, 274, 285, 295, 305-323, 328, 334
- array name, 24, 124-128, 179
- ASCII, 17-26, 58, 62, 83-91, 145, 258, 299
- ASCII code. *See ASCII.*
- assembler, 7, 8, 10, 16
- assembly language, 2-11
- assignment operator, 99, 103, 104
- associativity, 99-104, 112
- atoi, 137, 166, 167, 313
- automatic scaling, 127
- binary number, 4, 5, 17-20, 94, 110
- binary number system, 4, 5, 17, 94
- binary operators, 104
- binary words. *See binary number.*
- BIOS, 12
- bistable devices. *See transistors*
- bit field, 146
- bit pattern, 62, 82, 110-120, 148
- bottom-up design, 212, 218
- branch node, 35
- break statement, 247, 248, 256-260
- bubble-sort, 157
- buffer, 25, 294, 295, 309, 310, 327-334
- byte, 18, 27, 40, 81-84, 94-99, 123, 132, 145-149, 308, 314
- case statement, 247, 248
- case structure, 52, 203, 231-250, 256, 270
- character strings, 134, 135
- character variable, 62, 83, 91, 128
- characters, 20, 21, 82, 83, 193, 279, 280
- circular list, 30
- coded instructions. *See Program code.*
- column-major, 126
- comma operator, 255
- command line, 168, 169, 235
- comment, 8, 45, 121, 182, 212
- commutative**, 101
- compilation, 1, 11-16, 40, 41, 50-54, 132, 265
- compilation variable, 54
- compile time, 11, 67, 81, 127, 132, 136, 165
- compiler, 10-15, 40-56, 67, 81-110, 127-140, 165-173, 181-187
- complement. *See sign and magnitude.*
- component select, 328
- compound statement, 230, 239, 250, 251
- computer program. *See program*
- concatenation, 24, 51
- conditional compilation, 52, 53
- continue command. *See continue statement.*
- continue statement, 260, 261
- control break, 3292, 293, 31-334
- control character, 57, 299
- control field. *See key field.*
- control logic, 49, 67, 69, 74, 76, 85, 115-118, 155, 195-207, 215-233, 248, 258-278, 289-291
- control string, 57-59, 64, 320
- control structures, 9, 43, 165, 170, 218, 219, 228-230, 248, 249
- control variable, 85, 105, 247-259
- CPU, 2-6, 41, 94, 106, 118, 186, 192
- data hiding, 89
- data objects, 124
- data processing, 20, 105, 110, 123, 128, 187, 293, 327
- data structure, 1, 15, 20-29, 35-43, 66, 69, 81, 82, 123-128, 146, 160
- data type, 1, 17, 20-26, 41-46, 56, 66, 79-100, 111, 120-123, 134-139, 141-143, 165, 167, 171, 187, 322
- debug session, 53
- decimal format, 20, 83, 95
- declaration, 43-48, 56, 66-72, 80-105, 111, 112, 124-149, 164-189, 298-307, 313, 326
- declared variable, 89
- decrement, 37, 103-107, 187, 253, 255
- default statement, 247
- define, 14-16, 27, 29, 41-43, 51-59, 64-80, 97, 100, 114-122, 132-149, 162, 164, 175-179, 217, 238, 286-308, 322, 323
- define statement, 52, 97, 132
- degree, 36, 78, 80
- Diagnostics, 298
- digital computer, 2-4, 17-21, 81, 83, 94, 124, 186
- directives, 42-54, 66, 197, 298, 320
- distributive laws, 101
- double precision, 18, 84
- do-while, 108, 256, 258, 260
- duration, 179, 240, 286
- dynamic declaration. *See memory allocation*
- EBCDIC, 20, 21, 22
- editing format, 57, 86, 174
- elseif, 53
- else, 38, 52-56, 108, 112, 133, 135, 196-210, 219, 228-239, 245, 247
- endif, 52
- enum, 55, 88, 97-99
- enum structure, 97
- enumerated data, 97
- exit point, 170, 231-236, 242-251, 259
- exit point principle, 236, 242, 251
- exponent, 83
- expression, 13, 14, 27, 36-38, 43-52, 62-68, 76-86, 94-113, 124-136, 144-148, 166-188, 230, 238, 247-261, 298-307, 322, 323
- expressions, 9-15, 35-41, 57, 101-112, 128, 130, 142, 143, 229, 247-254, 301, 305
- extern key word, 182
- external medium, 293, 328
- external variable, 183
- fclose, 310, 328-331

fgets, 311, 330, 331
 FIFO, 32
 FILE control structure, 295
 file mode, 295, 296
 file processing. *See data processing.*
 fix point representation, 20
 floating point, 20, 21, 59, 65, 83-86, 95, 165, 173, 174, 188, 300-305
 floating point representation, 20, 21, 84
 flowchart, 49, 68-78, 115-121, 151-160, 202, 204, 213-232, 246, 263-275
 fopen, 294-297, 310, 328-334
 for loop, 118, 129, 150, 253-260
 for statement. *See for loop.*
 formal parameter, 16, 48, 307
 format specification, 48, 59, 64, 329, 331
 format specifiers, 57-61, 72, 86
 format string, 58, 166, 311, 331
 formatting string, 48, 58, 59, 310
 function, 15, 29, 41-79, 85-150, 158-224, 230-323, 328-334
 function call, 57, 105, 112, 117, 124, 142, 162, 165, 167, 169, 171, 173, 174, 176, 179, 180, 181, 185, 190, 193
 function declaration. *See prototype.*
 function definition, 49, 89, 93, 169-176, 193
 function prototype, 93, 165-173
 garbage collection, 29
 getch, 42, 63, 137, 164, 204, 233
 getchar, 62, 63, 311
 getche, 63
 global objects, 90, 179
 global variable, 182, 183, 190, 296
 goto command, 260, 261
 hardware system. *See Hardware.*
 header file, 15, 92, 93, 210, 293-319
 hexadecimal, 2, 4, 59-65, 112, 129, 131, 145, 299
 Hexadecimal number system, 2
 hierarchy chart, 201-207, 224, 226, 268-271, 290, 291
 high level language, 1, 9, 38, 43. *See Programming languages*
 identifier, 13-15, 38, 44-56, 66-71, 82-98, 112, 117, 118, 138-150, 172-181, 193, 251, 260, 294-308
 if statement, 52, 53, 108, 196, 232-239, 251, 298
 ifdef, 53
 if-else statement, 233, 238
 ifndef, 53
 include, 12-20, 38, 45-57, 70, 76, 83, 92, 94, 105, 120, 121, 163-167, 192-197, 212, 213, 226, 238, 249, 250, 293-308, 323
 include file, 45, 54, 70, 120, 121, 192, 193, 213, 298-307
 increment, 37, 103-107, 127, 134, 187, 253-260
 index, 7, 8, 15, 23, 24, 102, 124-126, 135, 138, 209
 index calculation, 126
 initialisation, 67, 84, 85, 127, 128, 136, 141, 187, 253-255
 in-place, 114
 input parameter. *See parameter value.*
 input stream, 62, 311, 312
 integers, 20, 83
 integrated circuit, 5
 interface, 2, 3, 44, 93, 123, 162-173
 internal documentation. *See comment.*
 interpreter, 10, 11
 interrupt vectors, 192, 193
 isdigit, 254, 299
 ISO, 9
 isspace, 253
 iterative, 193, 195, 262
 key field, 26, 326
 key words, 12, 44, 55, 56, 82, 83, 167, 250
 labels, 13, 260-267, 284-288
 left shift. *See operators.*
 lexical analyser, 12-14
 library file, 16, 46-50, 62, 63, 92, 134, 163, 197, 238, 293-301, 325
 library function, 298
 LIFO list, 32
 line directive, 54
 link editing. *See loader*
 link list, 28-31
 list, 22-33, 41-46, 51-58, 61-67, 73-76, 83, 92-101, 112-118, 129-142, 160-179, 193, 200, 240, 289, 298-315, 320, 334
 literal data, 37, 45, 101, 131
 literal string, 46
 loader, 16
 local copy, 170
 local object, 90
 location, 6- 8, 18, 25-28, 36-38, 56, 63, 81-94, 103, 106, 117-127, 134, 136, 142-149, 169, 170, 177, 178, 192, 210, 314-318
 logic design, 49, 155, 213-220, 236, 261, 268, 276
 logical operator, 104
 logical relational, 104, 108
 loop control statements, 249
 loop logic structure, 216, 249
 looping, 67, 193, 228, 231, 261
 l-value, 39, 41
 machine architecture, 2
 machine code. *See machine language.*
 machine instructions, 2-16, 37-41
 machine language, 2-16, 37- 41, 81, 84, 91, 125, 128, 188, 193
 machine programs, 2
 macro, 16, 48-54, 70, 120, 132, 133, 143, 294-319, 327
 macro definition, 16, 51
 macro expansion, 51
 macroprocessor, 16
 main memory, 25, 66, 118, 138, 155, 187-294, 326-331
 maintainability, 163, 236
 malloc, 118, 124, 315, 334
 mantissa, 83
 matrix, 158-160, 322, 323
 matrix arithmetic operations, 322
 member selector, 142
 memory allocation, 82, 88, 124, 141
 merge, 114-118, 157
 microcomputer system, 1, 81, 123
 microprocessor, 4, 5, 34
 minicomputers, 5
 mnemonic, 2-6
 modulus operator, 105
 multiple decision, 245
 nested case structure, 236-247
 nested loops, 256
 nested pointers, 131
 nested structure, 128, 139, 140, 236, 242
 null character, 134, 136
 numeric digits, 21, 82

object file, 93
 object program, 10-13
 one-dimensional array, 124, 125
 OP CODE. *See Operational Code.*
 OPERAND, 6, 7
 operating system, 43-50, 89, 148, 168, 169, 193, 197, 235, 294, 315
 Operational Code, 6, 37
 operator, 12-15, 37-39, 51, 63-65, 86-120, 132-155, 160, 176, 177, 233, 254, 302, 308, 328
 operators, 12-15, 37-43, 79, 80, 101-112, 231, 251
 output listing, 42, 67-76, 240, 246
 output stream, 57, 62, 311
 parameter, 44-58, 65, 86-118, 132, 136, 162-181, 195-202, 235, 237, 263, 295-307, 328, 329
 parameter values, 44, 49, 57, 107, 113, 162-168, 175, 176, 188-202, 273, 303, 329
 parity bit, 91
 parse tree, 14
 parsing. *See syntax analysis*
 permutation sequence, 114
 PIXEL, 17
 pointer, 6, 28-34, 54-65, 82, 88, 103, 127-147, 155-160, 168, 175-179, 187-193, 207, 209, 238, 263, 273, 274, 294-320, 328, 334
polyadic, 38
 portable, 2, 9, 43
 post fixed, 106
 precedence, 98-108, 142, 188
 precision, 18, 56-61, 84, 88, 96
 predecessor, 29-35
 pre-processor, 15, 16, 42-54, 63-70, 92, 197, 298
 preprocessor directive, 132
 printf, 42-76, 85-112, 129-137, 144-150, 163, 164, 173-181, 190-203, 230, 246, 274, 284, 311
 procedures, 13, 32, 44
 program code. *See source code.*
 program logic. *See control logic.*
 program specification, 73, 118, 155, 276, 291
 programming languages, 4, 9
 prototype, 44-50, 62-76, 92, 93, 134, 162-179, 189-193, 210, 261, 284, 293, 298-316
 pseudocode, 229
 punctuation, 21, 82, 299
 punctuator, 49, 51, 112, 202
 queue, 32
 rational pre-processing. *See conditional compilation*
 real numbers, 20, 83, 84, 95
 record, 15, 24-27, 66-74, 100, 119, 120, 138, 153-160, 198, 209, 263, 286-294, 326-334
 record structure, 27, 67, 70, 198
 recursion, 14, 113, 193-195, 261, 262
 recursive, 14, 32, 51, 162, 185, 186, 193-197, 262
 recursive call, 193-196
 register, 5-8, 20, 55, 118, 186, 187, *See CPU*
 relational operators, 108
 relocatable machine, 11, 12, 16
 relocatable machine instruction, 12
 replacement, 50, 51
 reserved words, 44, 54, 56
 return instruction, 33, 170, 188, 259, 261
 return value, 105, 108, 132, 141, 165-174, 195, 197, 233, 250, 302-305
 right shift. *See operators.*
 routines, 16, 43
 row vector. *See array*
 row-major, 126-133
 run time, 10, 11, 40, 178
 r-value, 39, 41
 scalar multiplication, 322, 323
 scalar quantity, 323
 scanf, 42-50, 63-65, 92, 137, 163-173, 251, 311
 scientific notation, 21, 60, 83, 84
 scope, 15, 90, 141, 162, 179-185, 302
 secondary memory, 118, 155, 293, 326-334
 self reference, 139-142
 semantic, 3, 15, 229-239
 semantics, 9, 133, 228-233
 sequential file, 326
 service functions, 192
 service interrupts. *See service functions*
 shelf, 34
 shuffle sort, 157
 sign and magnitude, 18
 sign bit. *See sign and magnitude.*
 significant digits, 21, 60, 84, 95, 303
 sizeof operator, 132
 software, 3-8, 40, 47, 74, 152, 163, 201-207, 224-231, 268, 272, 286
 sort algorithm. *See merge.*
 source code, 1-15, 37-74, 81-96, 105-128, 134, 135, 150-158, 166-182, 195-204, 214-221, 230-246, 251, 253, 260, 271, 293-298, 322-328
 source program. *see source code.*
 stack, 6, 23, 32-34, 193, 262
 stack overflow, 34
 stack underflow, 33
 standard error file, 298, 309
 standard include, 50, 136, 197, 238, 294
 standard library functions, 42, 80, 164-173, 292
 statement delimiter, 230
 statements, 1-15, 46-59, 101-112, 165, 196, 202, 218, 219, 228-236, 245-261, 298
 static object, 141
 storage allocation, 42, 91, 126
 stored program concept, 2
 strcmp, 70, 164, 317
 strcpy, 134, 237, 316
 stream, 62, 294-297, 309-313
 string data, 22, 24, 134-137
 strlen, 164, 318
 struct, 55, 66, 67, 138-144, 319-326
 structure definition, 27, 100, 127, 138-146, 295
 structure tag, 66, 138-143
 subroutine, 33, 46, 165, 188, 192
 subscript value, 124
 sub-tree, 35
 successor, 28-36
 switch statement, 203, 247, 248, 260
 symbol table, 12, 15, 40
 symbolic addressing, 8
 symbolic constant. *See macro.*
 symbolic names, 8
 symbols, 11, 12, 37, 38, 58, 74, 108, 275
 syntactic analysis, 13, 14
 syntax, 3-15, 40, 49, 101, 113, 138-149, 228, 229, 247-260
 ternary, 38, 112, 133
 tokens, 12-14, 38-51, 230, 318
 top-down design, 162, 167, 198
 toupper, 237, 300
 transistors, 4
 translator, 10, 11, 37
 tree, 13, 14, 27, 35, 36, 126, 142

- two dimensional array, 125
- two%ocomplement, 18-20
- type cast, 86, 99, 100, 130
- type conversion, 99, 100
- type modifiers, 87, 88
- type specifier, 87, 172
- typedef, 52, 55, 66, 143-147
- typing, 43, 105, 168
- unary operator, 37, 108
- undef statement, 51
- union, 55, 66, 149, 150
- union definition, 149
- UNIX, 43
- unsigned integer, 20, 166
- validation, 78, 120, 137, 168, 194, 203-217, 226, 251, 252, 264-278, 282
- variable name, 12, 13, 24, 25, 38, 44-58, 63-67, 82-115, 124-145, 170-181, 237, 238, 247, 250, 263, 323, 331
- variable type, 71, 91-99, 142, 167
- variables, 10, 37-45, 54-63, 79-112, 122-149, 165-187, 198, 210, 247-255, 323, 328
- variadic*, 38
- visibility*. *See scope*
- while, 6, 14, 16, 32, 55, 108, 137, 153, 193-203, 215-219, 228, 230, 239, 246-261
- while loop, 251, 258
- white-space character, 57
- whole numbers, 20, 83, 84, 95, 187
- word length, 5, 18, 118, 126

