

# Table of Contents

<b>Chapter 1 Welcome to Computer and Programming World!</b>	<b>1</b>
1.1 Why Computers?	1
1.1.1 What is a Computer?	2
1.1.2 Block Diagram of a Computer	3
1.1.2.1. Input unit	3
1.1.2.2. Memory unit	3
1.1.2.3. Central Processing Unit (CPU)	5
1.1.2.4. Output unit	5
1.2 Machine Language	6
1.3 Assembly Language	8
1.4 Operating System	9
1.5 C Language	10
1.5.1 History	10
1.5.2 Why one should use C language for programming?	10
1.5.3 Embarking C Programming: First C Program	11
1.6 Typical C Language Environment	15
1.6.1 Editor	16
1.6.2 Preprocessor	16
1.6.3 Compiler	16
1.6.4 Linker	17
1.7 Points to Remember	20
1.8 Deep Knowledge Section	27
Questions	29
General Information	32
A. A Brief History of Computer Technology	32
B. History of computers	32
C. History of Algorithms	33
Appendix	34
<b>Chapter 2 Structured Programming: Sequential and Selection Control Flow</b>	<b>35</b>
2.1 Sequential Control Flow	35
2.2 Selection Control Flow	40
2.2.1 if-else statement	40
2.2.2 if statement	43

2.2.3 switch-case Construct.....	49
2.2.4 enum .....	60
2.2.5 Conditional Operator .....	63
2.3 Points to Remember.....	65
2.4 Popular Postfix/Prefix operator .....	66
2.5 Operator Precedence.....	70
2.6 Logical Operators.....	73
2.7 Deep Knowledge Section.....	76
Questions.....	79
Appendix.....	84

**Chapter 3 Structured Programming: Iteration Control Flow ...85**

3.1 for .....	85
3.2 while loop.....	92
3.3 do-while loop .....	93
3.4 Points to Remember.....	94
3.5 break.....	99
3.6 continue .....	101
3.7 Nested loops.....	102
3.8 Counter Controlled and Sentinel Controlled Loops .....	104
3.9 goto statement.....	105
3.10 Deep Knowledge Section.....	110
Questions.....	117

**Chapter 4 Structured Programming: Function .....123**

4.1 Introduction.....	123
4.2 What is a function?.....	126
4.3 Scope And Storage Specifier of a Variable .....	134
4.3.1 Team Work is Spirit .....	145
4.3.2 register variable .....	153
4.4 Advantages and disadvantages of using functions .....	154
4.5 Points to Remember.....	155
4.6 Recursive Function .....	162
4.6.1 Practice Program 1:.....	166
4.6.2 Practice Program 2:.....	167
4.6.3 Practice Program 3:.....	167
4.6.4 Practice Program 4:.....	168
4.6.5 Practice Program 5:.....	168
4.7 Deep Knowledge Section.....	170
Questions.....	174

General Information.....	177
<b>Chapter 5 Array .....</b>	<b>179</b>
5.1 Introduction .....	179
5.2 Declaration and Definition of an Array .....	181
5.2.1 Points to Remember .....	188
5.2.2 Initialisation of Array .....	192
5.3 Array and Function .....	194
5.4 Array and String .....	196
5.5 Searching Array Element .....	197
5.5.1 Linear Search.....	197
5.5.2 Binary Search .....	199
5.6 Multi-dimensional Arrays.....	200
5.7 How Multi-dimensional Arrays are Stored in Memory?.....	204
5.8 Arrays and for loops.....	205
5.9 Points to Remember.....	208
5.10 Matrix and Multidimensional Array.....	208
5.11 Deep Knowledge Section .....	213
Questions.....	215
<b>Chapter 6 Structure.....</b>	<b>219</b>
6.1 What, Why and When Structure Data Type? .....	219
6.2 Structure Data Type and Structure Variable .....	222
6.2.1 Structure Data Type.....	222
6.2.1.1 Declaration.....	222
6.2.1.2 Definition .....	222
6.2.2 Structure Variable Declaration .....	237
6.2.3 Anonymous Structure .....	237
6.3 Structure and Function.....	239
6.3.1 Function Returning Structure Variable.....	240
6.4 Array and Structure.....	241
6.5 Nested Structure- Structure As A Data Member of Another Structures .....	243
6.6 Points to Remember.....	249
6.7 typedef.....	249
6.8 union .....	254
Questions .....	256
<b>Chapter 7 Pointer .....</b>	<b>265</b>
7.1 Your Memory, Your Time and You.....	265

7.2 Memory, Time and Pointer .....	266
7.3 Pointer and Functions.....	278
7.4 Pointer and Dynamic Memory Allocation.....	281
7.4.1 Uninitialised and NULL pointer.....	285
7.5 Array and Pointer .....	285
7.6 constness and pointer .....	292
7.7 Misuse of Pointer .....	296
7.7.1 Memory Leaks.....	296
Bug #1 - Uninitialised/Wild pointer.....	297
Bug #2 - Invalid Pointer References .....	297
Bug #3 - Null Pointer Reference.....	298
Bug#4 Attempting to free memory already freed .....	298
Bug#5 Attempting to write to memory that is already freed .....	299
Bug#6 Freeing memory that was not allocated. ....	299
Bug#7 Attempting to write to memory that was never allocated	299
Bug#8 Pointer re-assignment error leads to dangling pointer ...	300
7.8 Function Pointer.....	300
7.9 Double Pointer (Pointer to Pointer).....	301
7.10 Complicated Expressions .....	303
7.11 Points to Remember.....	307
7.12 Deep Knowledge Section.....	311
Questions.....	313

**Chapter 8 String .....319**

8.1 Introduction:.....	319
8.2 Character / Character constant.....	320
8.3 String Declaration and Definition.....	322
8.4 String Literals .....	328
8.5 String and Pointer .....	329
8.6 String and malloc function.....	331
8.7 Array of strings .....	334
8.8 String Common errors .....	335
8.9 String Functions Library.....	338
8.9.1 char * strcpy ( char * destination, const char * source ) .....	338
8.9.2 char * strncpy ( char * destination, const char * origin, size_t num) .....	342
8.9.3 size_t strlen ( const char * string ).....	343
8.9.4 void * memcpy ( void * destination, const void * source, size_t num) .....	344

8.10 Deep Knowledge Section .....	347
Questions.....	350
General Information.....	355
<b>Chapter 9 Input/Output .....</b>	<b>357</b>
9.1 Input/Output: What, Why And When? .....	357
9.2 Unformatted Console I/O.....	358
9.2.1 Character Input/Output .....	358
int getchar(void).....	360
int getch(void).....	362
int getche(void) .....	363
int getc(FILE *Stream).....	363
int fgetc(FILE *stream).....	364
int putchar(int c) .....	365
int putc(int char,FILE *stream) .....	366
int fputc(int c,FILE *stream).....	366
int ungetc(int c, FILE *stream) .....	367
9.2.2 String I/O:.....	368
char *fgets(char *s, int size, FILE *stream) .....	370
int puts( const char *string ).....	372
int fputs(const char * restrict s,FILE * restrict stream).....	372
9.3 Formatted Console I/O.....	372
int printf(char *control_string,argument_list) .....	372
int fprintf(FILE *stream, const char *format, ...) .....	377
int sprintf (char *str, const char *format, ...).....	378
snprintf:Safer alternatives to sprintf.....	379
int vprintf(char *format,va_list arg_ptr) .....	379
int scanf("control string",&var1,.....,&varn).....	379
int sscanf (const char *source, const char *format, ...) .....	381
int fscanf (FILE *file, const char *format, ...) .....	382
scanset.....	382
9.4 Deep knowledge section .....	384
Questions.....	385
Appendix.....	390
<b>Chapter 10 Files in C.....</b>	<b>393</b>
10.1 Introduction.....	393
10.1.1 FILE*fopen( const char *fname, const char *mode ).....	394
10.1.2 int fclose ( FILE * stream ) .....	395

**xii C for Beginners**

10.2 Sequential Files.....	395
10.3 File Positioning.....	401
10.3.1 void rewind (FILE *stream).....	401
10.3.2 int ftell (FILE *stream) .....	401
10.3.3 int fseek(FILE *stream,long offset, int origin).....	402
10.3.4.1 int fgetpos ( FILE * stream, fpos_t * position ).....	404
10.3.4.2 int fsetpos(FILE *stream,const fpos_t * position) .....	404
10.4 Random Files .....	407
10.5 size_t fwrite ( const void * ptr, size_t size, size_t count, FILE * fptr ) ..	408
Questions.....	413

**Chapter 11 Preprocessor .....417**

11.1 What is a preprocessor?.....	417
11.2 File Inclusion Directive.....	418
11.3 Symbolic Constant (Macro) Directives.....	419
11.3.1 #define .....	419
11.3.2 #undef.....	422
11.4 Conditional Compilation Directives.....	423
11.4.1 #if #elif endif .....	424
11.5 Conditional Execution Directives .....	426
11.5.1 # operator .....	426
11.5.2 ## operator .....	426
11.5.3 #error .....	427
11.5.4 #pragma.....	428
11.6 Predefined Macros .....	430
11.6.1 #line .....	431
11.7 Points to Remember .....	432
Questions.....	434

**Index .....439**