
CONTENTS

| | | |
|----------|--|-----------|
| | PREFACE | XV |
| 1 | INTRODUCTION TO COMPUTERS AND PROGRAMMING | 1 |
| 1.1 | COMPUTER HARDWARE | 1 |
| 1.2 | PROGRAMMING LANGUAGES | 8 |
| 1.3 | THE C PROGRAMMING LANGUAGE | 10 |
| 1.4 | ADVANTAGES OF C | 11 |
| 1.5 | DISADVANTAGES OF C | 15 |
| 1.6 | ANSI C STANDARD | 16 |
| 1.7 | CREATING A C PROGRAM | 16 |
| 1.8 | THE STANDARD C LIBRARY | 19 |
| 1.9 | SUMMARY | 20 |
| | EXERCISES | 21 |
| 2 | BASIC ELEMENTS OF C PROGRAMS | 23 |
| 2.1 | STRUCTURE OF A C PROGRAM | 23 |
| 2.2 | A SIMPLE C PROGRAM | 25 |
| 2.3 | FUNCTIONS | 30 |
| 2.4 | STATEMENTS | 31 |
| 2.5 | COMMENTS | 33 |
| 2.6 | PUNCTUATION AND STYLE | 34 |
| 2.7 | THE C PREPROCESSOR | 35 |
| 2.8 | THE print() FUNCTION | 39 |
| 2.9 | THE scanf() FUNCTION | 43 |
| 2.10 | SUMMARY | 48 |
| | EXERCISES | 52 |

| | | |
|----------|--|------------|
| 3 | PROBLEM SOLVING ON A COMPUTER | 56 |
| 3.1 | PROBLEM-SOLVING STEPS | 56 |
| 3.2 | ALGORITHMS | 59 |
| 3.3 | FLOWCHARTS | 61 |
| 3.4 | PSEUDOCODE | 64 |
| 3.5 | FUNDAMENTAL CONTROL STRUCTURES | 65 |
| 3.6 | TOP-DOWN DESIGN | 77 |
| 3.7 | MODULAR DESIGN | 79 |
| 3.8 | STRUCTURED PROGRAMMING | 82 |
| 3.9 | PROGRAM ERRORS | 83 |
| 3.10 | DEBUGGING TECHNIQUES | 86 |
| 3.11 | PROGRAM TESTING | 88 |
| 3.12 | PROGRAM DOCUMENTATION | 89 |
| 3.13 | MAINTENANCE AND MODIFICATION | 90 |
| 3.14 | PROGRAM EFFICIENCY | 90 |
| 3.15 | SUMMARY | 91 |
| | EXERCISES | 93 |
| 4 | VARIABLES, CONSTANTS, AND DATA TYPES | 96 |
| 4.1 | VARIABLES AND CONSTANTS | 96 |
| 4.2 | DATA TYPES | 99 |
| 4.3 | INTEGERS | 100 |
| 4.4 | REAL NUMBERS | 105 |
| 4.5 | PROGRAMMING PROJECT: WEIGHT OF HOLLOW CYLINDER | 112 |
| 4.6 | CHARACTERS | 113 |
| 4.7 | THE sizeof OPERATOR | 118 |
| 4.8 | USING THE DIFFERENT DATA TYPES | 118 |
| 4.9 | INACCURACIES IN FLOATING POINT COMPUTATIONS | 121 |
| 4.10 | OVERFLOW AND UNDERFLOW | 122 |
| 4.11 | PROGRAMMING PROJECT: SOLUTION OF SIMULTANEOUS EQUATIONS | 125 |
| 4.12 | SUMMARY | 128 |
| | EXERCISES | 132 |
| 5 | ARITHMETIC OPERATIONS | 135 |
| 5.1 | OPERATORS | 136 |
| 5.2 | ASSIGNMENT STATEMENTS | 136 |
| 5.3 | ARITHMETIC OPERATORS | 139 |
| 5.4 | ARITHMETIC ASSIGNMENT OPERATORS | 143 |

| | | |
|----------|--|------------|
| 5.5 | INCREMENT AND DECREMENT OPERATORS | 145 |
| 5.6 | PRECEDENCE OF OPERATORS 148 | |
| 5.7 | ARITHMETIC WITH MIXED TYPES | 154 |
| 5.8 | MATHEMATICAL FUNCTIONS | 163 |
| 5.9 | PROGRAMMING PROJECT: RESULTANT OF TWO FORCES | 170 |
| 5.10 | PROGRAMMING PROJECT: OPEN CHANNEL FLOW | 174 |
| 5.11 | PROGRAMMING PROJECT: ALTERNATING CURRENT CIRCUIT | 178 |
| 5.12 | SUMMARY | 181 |
| | EXERCISES | 186 |
| 6 | CONTROL STRUCTURES: SELECTION | 195 |
| 6.1 | THE if STATEMENT | 196 |
| 6.2 | RELATIONAL OPERATORS | 198 |
| 6.3 | LOGICAL OPERATORS | 204 |
| 6.4 | THE if-else STATEMENT | 208 |
| 6.5 | PROGRAMMING PROJECT: GROSS SALARY OF AN EMPLOYEE | 210 |
| 6.6 | PROGRAMMING PROJECT: BEAM ANALYSIS | 215 |
| 6.7 | NESTED if STATEMENTS | 221 |
| 6.8 | PROGRAMMING PROJECT: AXIAL LOAD FOR A STEEL COLUMN | 222 |
| 6.9 | THE else-if CONSTRUCT | 229 |
| 6.10 | PROGRAMMING PROJECT: ROOTS OF A QUADRATIC EQUATION | 230 |
| 6.11 | THE switch AND break STATEMENTS | 235 |
| 6.12 | THE CONDITIONAL OPERATOR | 238 |
| 6.13 | SUMMARY | 239 |
| | EXERCISES | 244 |
| 7 | CONTROL STRUCTURES: LOOPS | 248 |
| 7.1 | THE while LOOP | 249 |
| 7.2 | PROGRAMMING PROJECT: SQUARE ROOTS BY ITERATION | 255 |
| 7.3 | PROGRAMMING PROJECT: PROJECTILE TRAJECTORY | 261 |
| 7.4 | THE for LOOP | 266 |
| 7.5 | PROGRAMMING PROJECT: TEST SCORES AND GRADES | 273 |
| 7.6 | THE do-while LOOP | 277 |
| 7.7 | THE break STATEMENT | 279 |
| 7.8 | THE continue STATEMENT | 282 |

| | | |
|----------|--|------------|
| 7.9 | SUMMATION OF SERIES | 283 |
| 7.10 | PROGRAMMING PROJECT: EVALUATION OF LOG(1+x) USING SERIES | 288 |
| 7.11 | PROGRAMMING PROJECT: HEIGHT OF A CATENARY CABLE | 291 |
| 7.12 | SUMMARY | 294 |
| | EXERCISES | 299 |
| 8 | FUNCTIONS | 307 |
| 8.1 | MODULAR DESIGN | 307 |
| 8.2 | FUNCTIONS | 309 |
| 8.3 | THE STRUCTURE OF A C FUNCTION | 309 |
| 8.4 | INVOKING A FUNCTION | 313 |
| 8.5 | FUNCTIONS THAT RETURN A VALUE | 320 |
| 8.6 | FUNCTION PROTOTYPES | 322 |
| 8.7 | CALL BY VALUE | 325 |
| 8.8 | PROGRAMMING PROJECT: TRIANGLE SOLUTION | 333 |
| 8.9 | PROGRAMMING PROJECT: COLOR CODING OF RESISTORS | 340 |
| 8.10 | STORAGE CLASSES | 349 |
| 8.11 | RECURSIVE FUNCTIONS | 360 |
| 8.12 | SUMMARY | 368 |
| | EXERCISES | 374 |
| 9 | ONE-DIMENSIONAL ARRAYS | 378 |
| 9.1 | ARRAYS IN C | 379 |
| 9.2 | ARRAY DECLARATION | 380 |
| 9.3 | ARRAY INITIALIZATION | 382 |
| 9.4 | ARRAY SUBSCRIPTS | 383 |
| 9.5 | ARRAY PROCESSING | 385 |
| 9.6 | ARRAYS AND FUNCTIONS | 389 |
| 9.7 | PROGRAMMING PROJECT: TEMPERATURE DISTRIBUTION ALONG A BAR | 397 |
| 9.8 | PROGRAMMING PROJECT: VECTOR OPERATIONS | 402 |
| 9.9 | PROGRAMMING PROJECT: BEAM ANALYSIS | 409 |
| 9.10 | SORTING | 422 |
| 9.11 | SELECTION SORT | 423 |
| 9.12 | THE BUBBLE SORT | 426 |
| 9.13 | THE SHELL SORT | 431 |
| 9.14 | THE QUICK SORT | 436 |
| 9.15 | COMPARISON OF SORTING ROUTINES | 438 |

| | | |
|-----------|--|------------|
| 9.16 | SUMMARY | 441 |
| | EXERCISES | 444 |
| 10 | POINTERS | 450 |
| 10.1 | ADDRESSES AND THE ADDRESS-OF OPERATOR | 451 |
| 10.2 | POINTER VARIABLES | 453 |
| 10.3 | THE INDIRECTION OPERATOR (*) | 457 |
| 10.4 | POINTERS AND ARRAYS | 463 |
| 10.5 | POINTER ARITHMETIC | 464 |
| 10.6 | ARRAY NOTATION AND POINTER NOTATION | 468 |
| 10.7 | POINTERS AS FUNCTION ARGUMENTS | 472 |
| 10.8 | PROGRAMMING PROJECT: SOLUTION OF SIMULTANEOUS EQUATIONS | 476 |
| 10.9 | PROGRAMMING PROJECT: REAL ROOTS OF A CUBIC EQUATION | 484 |
| 10.10 | SUMMARY | 491 |
| | EXERCISES | 495 |
| 11 | CHARACTER STRINGS | 502 |
| 11.1 | CHARACTER STRING CONSTANTS | 502 |
| 11.2 | DECLARING AND INITIALIZING STRINGS | 504 |
| 11.3 | STRING INPUT AND OUTPUT | 509 |
| 11.4 | THE C LIBRARY STRING HANDLING FUNCTIONS | 513 |
| 11.5 | DETERMINING THE LENGTH OF STRINGS | 514 |
| 11.6 | COPYING STRINGS | 519 |
| 11.7 | CONCATENATING STRINGS | 524 |
| 11.8 | COMPARING STRINGS | 526 |
| 11.9 | SEARCHING FOR A CHARACTER IN A STRING | 530 |
| 11.10 | STRING TO NUMBER CONVERSIONS | 530 |
| 11.11 | PROGRAMMING PROJECT: AZIMUTH AND BEARING CONVERSIONS | 533 |
| 11.12 | ARRAYS OF STRINGS | 541 |
| 11.13 | ARRAYS OF POINTERS | 542 |
| 11.14 | COMMAND LINE ARGUMENTS | 543 |
| 11.15 | SUMMARY | 546 |
| | EXERCISES | 549 |
| 12 | FILE OPERATIONS | 553 |
| 12.1 | STREAMS, BUFFERS, AND FILE I/O FUNCTIONS | 554 |
| 12.2 | THE FILE POINTER | 554 |
| 12.3 | OPENING A FILE | 555 |

| | | |
|-----------|--|------------|
| 12.4 | CLOSING FILES | 558 |
| 12.5 | STANDARD FILES | 560 |
| 12.6 | READING AND WRITING CHARACTERS | 561 |
| 12.7 | STRING INPUT AND OUTPUT | 567 |
| 12.8 | FORMATTED FILE I/O | 571 |
| 12.9 | PROGRAMMING PROJECT: WEATHER DATA | 576 |
| 12.10 | BINARY AND BLOCK I/O | 583 |
| 12.11 | RANDOM ACCESS | 586 |
| 12.12 | SUMMARY | 589 |
| | EXERCISES | 591 |
| 13 | FORMATTED INPUT AND OUTPUT | 596 |
| 13.1 | FORMATTED OUTPUT | 596 |
| 13.2 | FORMAT SPECIFIERS | 599 |
| 13.3 | EXAMPLES OF printf() STATEMENTS | 603 |
| 13.4 | FORMATTED INPUT | 609 |
| 13.5 | PROGRAMMING PROJECT: PRINTING A REPORT | 616 |
| 13.6 | FORMATTED STRING INPUT AND OUTPUT | 620 |
| 13.7 | CHARACTER TESTING MACROS | 623 |
| 13.8 | CHARACTER CONVERSION ROUTINES | 624 |
| 13.9 | SUMMARY | 625 |
| | EXERCISES | 628 |
| 14 | TWO-DIMENSIONAL ARRAYS AND MATRICES | 632 |
| 14.1 | TWO DIMENSIONAL ARRAYS | 633 |
| 14.2 | PROGRAMMING PROJECT: TEMPERATURE DISTRIBUTION IN A PLATE | 641 |
| 14.3 | MULTIDIMENSIONAL ARRAYS AS FUNCTION ARGUMENTS | 646 |
| 14.4 | MATRICES | 650 |
| 14.5 | MATRICES IN C | 653 |
| 14.6 | THE DETERMINANT OF A SQUARE MATRIX | 665 |
| 14.7 | PROGRAMMING PROJECT: DETERMINANT OF A 4 BY 4 MATRIX | 670 |
| 14.8 | PROGRAMMING PROJECT: MATRIX OPERATIONS PROGRAM | 677 |
| 14.9 | SUMMARY | 680 |
| | EXERCISES | 686 |
| 15 | STRUCTURES, UNIONS AND TYPEDEF | 694 |
| 15.1 | STRUCTURES | 695 |
| 15.2 | OPERATIONS ON STRUCTURES | 704 |

| | | |
|-----------|--|------------|
| 15.3 | NESTED STRUCTURES | 706 |
| 15.4 | ARRAYS AND STRUCTURES | 709 |
| 15.5 | PROGRAMMING PROJECT: STUDENT GRADES | 712 |
| 15.6 | POINTERS TO STRUCTURES | 718 |
| 15.7 | FUNCTIONS AND STRUCTURES | 723 |
| 15.8 | THE typedef STATEMENT | 730 |
| 15.9 | PROGRAMMING PROJECT: COMPLEX ARITHMETIC | 733 |
| 15.10 | UNIONS | 742 |
| 15.11 | ENUMERATED TYPES | 744 |
| 15.12 | SUMMARY | 746 |
| | EXERCISES | 750 |
| 16 | THE C PREPROCESSOR | 755 |
| 16.1 | THE C PREPROCESSOR | 755 |
| 16.2 | THE #define DIRECTIVE | 756 |
| 16.3 | MACRO DEFINITIONS WITH ARGUMENTS | 758 |
| 16.4 | FILE INCLUSION | 763 |
| 16.5 | CONDITIONAL COMPILATION | 764 |
| 16.6 | PREDEFINED PREPROCESSOR MACROS | 771 |
| 16.7 | OTHER PREPROCESSOR DIRECTIVES | 773 |
| 16.8 | SUMMARY | 774 |
| | EXERCISES | 778 |
| 17 | DYNAMIC DATA STRUCTURES | 781 |
| 17.1 | DYNAMIC MEMORY ALLOCATION | 782 |
| 17.2 | LINKED LISTS | 789 |
| 17.3 | PROGRAMMING PROJECT: LINKED LIST FOR STORING PARTS DATA | 791 |
| 17.4 | DOUBLY LINKED LISTS | 806 |
| 17.5 | STACKS | 808 |
| 17.6 | QUEUES | 816 |
| 17.7 | SUMMARY | 823 |
| | EXERCISES | 825 |
| 18 | ROOTS OF NONLINEAR EQUATIONS | 828 |
| 18.1 | APPROXIMATIONS TO THE ROOTS | 830 |
| 18.2 | SEARCH METHOD | 831 |
| 18.3 | BISECTION METHOD | 833 |
| 18.4 | C PROGRAM FOR THE BISECTION METHOD | 836 |
| 18.5 | APPLICATION: MAXIMUM DEFLECTION OF A BEAM | 841 |
| 18.6 | METHOD OF FALSE POSITION | 845 |

| | | |
|-----------|--|------------|
| 18.7 | C PROGRAM FOR THE METHOD OF FALSE POSITION | 847 |
| 18.8 | APPLICATION: OPEN-CHANNEL FLOW | 850 |
| 18.9 | THE NEWTON-RAPHSON METHOD | 854 |
| 18.10 | COMPUTER PROGRAM FOR NEWTON-RAPHSON METHOD | 859 |
| 18.11 | THE SECANT METHOD | 864 |
| 18.12 | SUMMARY | 865 |
| | EXERCISES | 867 |
| 19 | LINEAR SIMULTANEOUS EQUATIONS | 872 |
| 19.1 | GAUSS ELIMINATION | 875 |
| 19.2 | GAUSS ELIMINATION WITH PARTIAL PIVOTING | 878 |
| 19.3 | C PROGRAM FOR GAUSS ELIMINATION | 880 |
| 19.4 | DETERMINANTS USING GAUSS ELIMINATION | 885 |
| 19.5 | MATRIX INVERSION | 886 |
| 19.6 | C FUNCTION FOR COMPUTING THE INVERSE OF A MATRIX | 890 |
| 19.7 | ITERATIVE METHODS | 893 |
| 19.8 | THE JACOBI METHOD | 894 |
| 19.9 | C PROGRAM FOR THE JACOBI METHOD | 897 |
| 19.10 | THE GAUSS-SEIDEL METHOD | 900 |
| 19.11 | C PROGRAM FOR GAUSS-SEIDEL ITERATION | 903 |
| 19.12 | APPLICATION: ELECTRICAL CIRCUIT NETWORKS | 907 |
| 19.13 | APPLICATION: STRUCTURAL ANALYSIS | 910 |
| 19.14 | COMPARISON OF DIRECT AND INDIRECT METHODS | 913 |
| 19.15 | SUMMARY | 914 |
| | EXERCISES | 915 |
| 20 | CURVE FITTING AND INTERPOLATION | 920 |
| 20.1 | LINEAR REGRESSION ANALYSIS | 921 |
| 20.2 | C PROGRAM FOR LINEAR REGRESSION ANALYSIS | 926 |
| 20.3 | APPLICATION: RELATIONSHIP BETWEEN PRECIPITATION AND RUNOFF | 930 |
| 20.4 | LINEARIZATION | 931 |
| 20.5 | APPLICATION: VISCOSITY AND TEMPERATURE RELATIONSHIP | 933 |
| 20.6 | APPLICATION: RADIOACTIVE DECAY | 935 |
| 20.7 | POLYNOMIAL REGRESSION | 936 |
| 20.8 | C PROGRAM FOR POLYNOMIAL LEAST SQUARES CURVE FITTING | 940 |
| 20.9 | INTERPOLATION | 944 |
| 20.10 | LINEAR INTERPOLATION | 946 |

| | | |
|-----------|--|-------------|
| 20.11 | SECOND-ORDER INTERPOLATING POLYNOMIAL | 948 |
| 20.12 | HIGHER-ORDER LAGRANGE POLYNOMIALS | 949 |
| 20.13 | C PROGRAM FOR INTERPOLATION USING LAGRANGE POLYNOMIAL | 952 |
| 20.14 | SUMMARY | 955 |
| | EXERCISES | 956 |
| 21 | NUMERICAL INTEGRATION AND DIFFERENTIATION | 959 |
| 21.1 | NUMERICAL INTEGRATION | 960 |
| 21.2 | THE TRAPEZOIDAL RULE | 963 |
| 21.3 | C PROGRAM FOR TRAPEZOIDAL RULE | 965 |
| 21.4 | SIMPSON'S RULE | 968 |
| 21.5 | COMPUTER PROGRAM FOR INTEGRATION USING SIMPSON'S RULE | 971 |
| 21.6 | GAUSS QUADRATURE | 975 |
| 21.7 | C PROGRAM FOR GAUSS QUADRATURE | 980 |
| 21.8 | APPLICATION: FLOW THROUGH A CIRCULAR PIPE | 983 |
| 21.9 | NUMERICAL DIFFERENTIATION | 985 |
| 21.10 | APPROXIMATIONS FOR THE FIRST DERIVATIVE | 987 |
| 21.11 | COMPUTER IMPLEMENTATION OF NUMERICAL DIFFERENTIATION | 991 |
| 21.12 | HIGHER ORDER DERIVATIVES | 991 |
| 21.13 | SUMMARY | 995 |
| | EXERCISES | 996 |
| 22 | SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS | 1000 |
| 22.1 | TYPES OF DIFFERENTIAL EQUATIONS | 1001 |
| 22.2 | FIRST-ORDER ORDINARY DIFFERENTIAL EQUATIONS | 1004 |
| 22.3 | EULER'S METHOD | 1005 |
| 22.4 | C PROGRAM FOR EULER'S METHOD | 1008 |
| 22.5 | TAYLOR'S SERIES METHODS | 1012 |
| 22.6 | THE IMPROVED EULER METHOD | 1015 |
| 22.7 | COMPUTER PROGRAM FOR IMPROVED EULER'S METHOD | 1019 |
| 22.8 | MODIFIED EULER METHOD | 1019 |
| 22.9 | COMPUTER PROGRAM FOR MODIFIED EULER'S METHOD | 1023 |
| 22.10 | RUNGE-KUTTA METHODS | 1025 |
| 22.11 | COMPUTER IMPLEMENTATION OF RUNGE-KUTTA METHOD | 1028 |
| 22.12 | MULTISTEP METHODS | 1031 |

| | | |
|-----------|---|-------------|
| 22.13 | PREDICTOR-CORRECTOR METHODS | 1034 |
| 22.14 | C PROGRAM FOR THE ADAMS-MOULTON METHOD | 1037 |
| 22.15 | SECOND-ORDER DIFFERENTIAL EQUATIONS | 1042 |
| 22.16 | SUMMARY | 1045 |
| | EXERCISES | 1047 |
| 23 | STATISTICS AND SIMULATION | 1050 |
| 23.1 | GRAPHICAL REPRESENTATION OF DATA | 1051 |
| 23.2 | SAMPLE STATISTICS | 1055 |
| 23.3 | PROBABILITY DENSITY AND DISTRIBUTION FUNCTIONS | 1063 |
| 23.4 | SIMULATION | 1067 |
| 23.5 | RANDOM NUMBER GENERATORS | 1068 |
| 23.6 | RANDOM NUMBERS FROM A UNIFORM DISTRIBUTION | 1068 |
| 23.7 | RANDOM NUMBERS FROM A NORMAL DISTRIBUTION | 1076 |
| 23.8 | RANDOM NUMBERS FROM OTHER DISTRIBUTIONS | 1078 |
| 23.9 | APPLICATION: MONTE CARLO INTEGRATION | 1092 |
| 23.10 | APPLICATION: RANDOM WALK | 1096 |
| 23.11 | SUMMARY | 1100 |
| | EXERCISES | 1101 |
| A | NUMBER SYSTEMS | 1104 |
| A.1 | BINARY NUMBERS | 1105 |
| A.2 | HEXADECIMAL NUMBERS | 1106 |
| B | ASCII CHARACTER SET | 1109 |
| C | INDEX OF PROGRAMS | 1113 |
| | INDEX | 1121 |