Code Factory

100+ Programming Solutions

Navigating the wonders beyond binary brilliance

Shikha Jain

Kavita Pandey

www.bpbonline.com

First Edition 2024

ISBN: 978-93-55519-115

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

My son, ***Vidit****, for igniting the idea behind this book*

My husband, parents, and siblings for their endless support and blessings

—***Shikha Jain***

My dearest daughter, ***Sachi****, your love fuels every word in this book*

My husband and the Almighty God, for their guidance and inspiration, without whom this book would not have been possible

—***Kavita Pandey***

About the Authors

- **Shikha Jain** is an Assistant Professor at Jaypee Institute of Information Technology (JIIT), NOIDA, INDIA. She has more than twenty years of research and academic experience. She has received her PhD in computer science from Jaypee Institute of Information Technology, Noida, India. She has contributed significantly to the field, publishing numerous research papers in prestigious journals and conferences.

  She was a guest editor for the notable Advances in Computational Intelligence and its Applications issue in the International Journal of Information Retrieval Research. She is an active reviewer of many International Journals and a technical program committee member of various International Conferences. Her research area includes Affective Computing, Emotion Modelling, Cognitive Affective Architectures, Machine Learning, and Soft Computing. She also takes on the editor role for the book Artificial Intelligence, Machine Learning, and Mental Health in Pandemics: A Computational Approach.

- **Kavita Pandey** is an Assistant Professor (Senior Grade) at Jaypee Institute of Information Technology (JIIT), Noida. She has more than 20 years of academic experience. She received her B.Tech. in Computer Science and Engineering from M.D. University in 2002 and M.Tech. (CS) from Banasthali Vidyapeeth University in 2003. She obtained her Ph.D. (CS) from JIIT, Noida, India, in January 2017. Presently, she is guiding two Ph.D. students. She has mentored 16 M.Tech. thesis/industrial projects and 57 + undergraduate projects.

  She is an editor of a book named Artificial Intelligence, Machine Learning, and Mental Health in Pandemics: A Computational Approach. She has published one patent. She also contributed to IGNOU study material titled Data Communications and Computer Network. Her research interests include Soft Computing, Machine Learning, Vehicular Ad hoc Networks, the Internet of Things, and Optimization Techniques. She has published 60+ articles in International journals and conferences, including Wiley, IEEE, Springer, Inderscience, etc. She was a guest editor of the special issue Advances in Computational Intelligence and Applications in the International Journal of Information Retrieval Research (IJIRR), IGI Global, ESCI, and Web of Science. She is working as an Academic Editor on the Editorial Board of the International Journal of Distributed Sensor Networks, Hindawi. She was a reviewer in many international journals of renowned publishers, including Elsevier, Inderscience, and IEEE Access, among others. She is an active TPC member of many conferences, such as FSDM, REDSET, IC3, UPCON, TEAMC, ICTCS, and many more. She is also a senior member of the IEEE Society.

About the Reviewer

Sachin Jain, a seasoned professional with over two decades of experience in the IT industry, stands as a testament to the fusion of passion and expertise. Born with a natural inclination towards technology, he embarked on a journey that has defined him as a master in his field. He pursued his academic aspirations with a Master's in Computer Applications, laying the foundation for an illustrious career.

With an impressive 20-plus years of experience, Sachin has become a stalwart in the IT industry, contributing significantly to developing and designing large-scale enterprise applications. His coding prowess and innovative approach have set him apart, making him a go-to technology expert.

Beyond the world of coding and algorithms, Sachin finds solace in his passion for travel and cinema. Exploring new landscapes and immersing himself in diverse cultures fuels his creativity and broadens his perspective. Amid intricate lines of code, Sachin discovers balance through the magic of storytelling, whether on the screen or in the pages of a good book.

Sachin Jain's journey is a testament to the harmonious blend of technical understanding and a rich, multifaceted life. As he continues to push the boundaries of what is possible in the IT landscape, Sachin remains a true connoisseur of the digital age.

Acknowledgement

Alone we can do so little; together we can do so much.

—Helen Keller

We humbly acknowledge the divine presence that has guided and inspired the creation of this book. We sincerely thank Almighty God for providing wisdom, strength, and inspiration throughout this journey.

A heartfelt thanks go to the entire team of BPB Publishers for their unwavering support in bringing our dream to fruition.

We are grateful to known and unknown individuals whose contributions have enriched our pool of knowledge, enabling us to author this book.

We sincerely appreciate the reviewers whose insightful suggestions and feedback significantly enhanced the quality of this book.

Lastly, we express our gratitude to all the readers for their interest in our book. Your encouragement is invaluable.

Preface

Welcome to Code Factory. This is specially designed for new entrants in the programming world. The main objective of the book is to trigger and nurture logic-building skills among students. It provides a comprehensive introduction to the fundamental programming concepts through the lens of two powerful languages—C and Python. The book will help the reader develop problem-solving and logic-building abilities with the help of algorithms and flowcharts. It will also teach how to transform the flowchart/algorithm into a program. Simultaneously, engaging with flowcharts, algorithms, and programming enhances the reader's ability to comprehend the translation of a given problem into logical solutions and subsequently into executable programs. By working in parallel with these components, readers gain a holistic perspective on the problem-solving process. It reinforces the connection between the problem statement, the logical constructs devised in algorithms, and their representation in a programming language.

The book is structured to introduce concepts gradually, ensuring a smooth learning curve. Each chapter presents challenges of increasing difficulty, allowing students to build on their skills progressively. One book chapter is designed to expose real-world examples of applying learned concepts to solve coding problems, enhancing their ability to tackle programming challenges in various domains.

This book will help the readers develop a well-rounded skill set covering flowcharts, algorithmic thinking, and practical implementation in both C and Python languages. It will provide a holistic foundation for anyone aspiring to become fearless in coding. I hope this guide will empower you to embark on a fulfilling journey into coding.

Chapter 1: Simple Input Output Program – It provides a comprehensive understanding of the fundamental concepts and techniques involved in handling input and output operations in programming. It aims to offer a thorough comprehension of fundamental algorithms flowcharts and Python and C code implementations. These programs help users grasp crucial concepts, including user input, data manipulation, and result display. Consider this chapter a foundational step for constructing more intricate applications in subsequent book sections.

Chapter 2: Conditional Statements – It covers the concept of decision statements(if-else), elucidating them through flowcharts, algorithms, and coding. Using numerous examples, it has effectively illustrated the various facets of conditional statements such as simple if, if-else, if-else if-else, and so on. Decision statements enable the execution of specific code

blocks based on certain conditions. They provide the logic for making choices within a program, making them a fundamental building block.

Chapter 3: Simple Loops – It discusses the basic loop statements such as while-loop and for-loop. In programming, a loop is a fundamental control structure that allows you to execute a block of code repeatedly as long as a specific condition is met. Loops are essential for automating repetitive tasks, iterating over certain statements, and performing various computational tasks efficiently. It optimizes the code as well. Overall, the chapter underscores the significance of loops as essential tools for improving code automation and computational efficiency in programming

Chapter 4: Complex Loops – It delves into the intricate applications of loops, showcasing their capacity to address complex problems. Loops are not confined to essential repetition; they can be seamlessly integrated with multiple conditional statements to devise sophisticated solutions. Furthermore, the chapter elevates their significance in handling certain challenges that demand the utilization of multiple loops, either sequentially or in the form of nested loops.

Chapter 5: Complex Problem Solving – It presents the complex logic building using various programming constructs (as discussed in previous chapters). The main focus of the chapter is on solving challenging problems by thinking of some logic. The chapter presents the problems or challenges that require creative and intricate solutions. These problems often go beyond simple algorithmic steps and may involve multiple steps, conditional logic, loops, lists, and matrices. The chapter discusses creating a logical flow for solving complex problems, including algorithms and flowcharts, to plan solutions before writing executable code.

Chapter 6: Real World Problems – It guides readers to navigate the intersection of real-world problems and programming constructs. The primary emphasis lies in strategically mapping these tangible challenges to previously studied concepts. By aligning real-world problems with established programming principles, the chapter seeks to demystify complex issues and demonstrate how simple programming constructs can be ingeniously applied.

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

1. Simple Input Output Program 1

Introduction 1

Structure 2

Objectives 2

Program 1.1: Print your name 3

Sample input output 3

Algorithm 3

Flowchart 3

Python code 4

Python code explanation 4

C code 4

C code explanation 5

Program 1.2: Print your lucky number 5

Sample input output 5

Algorithm 5

Flowchart 5

Python code 6

Python code explanation 6

C code 6

C code explanation 7

Program 1.3: Print your details 7

Sample input output 7

Algorithm 8

Flowchart 8

Python code 8

Python code explanation 9

C code 9

C code explanation 10

Program 1.4: Find sum of two numbers 10
Sample input output *10*
Algorithm *11*
Flowchart *11*
Python code *12*
Python code explanation *12*
C code *12*
C code explanation *13*
Program 1.5: Find product of two numbers 13
Sample input output *13*
Algorithm *14*
Flowchart *14*
Python code *14*
Python code explanation *15*
C code *15*
C code explanation *15*
Program 1.6: Perform division 16
Sample input output *16*
Algorithm *16*
Flowchart *17*
Python code *17*
Python code explanation *17*
C code *18*
C code explanation *18*
Program 1.7: Print equivalent rational numbers 19
Sample input output *19*
Algorithm *19*
Flowchart *19*
Python code *20*
Python code explanation *20*
C code *21*
C code explanation *21*

Program 1.8: Swapping using temporary variable 21
Sample input output 21
Algorithm 22
Flowchart 22
Python code 23
Python code explanation 23
C code 23
C code explanation 24
Program 1.9: Swapping without temporary variable 24
Sample input output 24
Algorithm 25
Flowchart 25
Python code 25
C code 26
C code explanation 27
Program 1.10: Find square of a number 27
Sample input output 27
Algorithm 28
Flowchart 28
Python code 28
Python code explanation 29
C code 29
C code explanation 29
Program 1.11: Find cube of a number 29
Sample input output 30
Algorithm 30
Flowchart 30
Python code 31
Python code explanation 31
C code 31
C code explanation 31

Program 1.12: Find average 31
Sample input output *32*
Algorithm *32*
Flowchart *32*
Python code *33*
Python code explanation *34*
C code *34*
C code explanation *34*
Program 1.13: Find percentage marks 34
Sample input output *35*
Algorithm *35*
Flowchart *35*
Python code *36*
Python code explanation *37*
C code *37*
C code explanation *37*
Program 1.14: Convert meter to centimeter 37
Sample input output *38*
Algorithm *38*
Flowchart *38*
Python code *39*
Python code explanation *39*
C code *39*
C code explanation *39*
Program 1.15: Convert degree F to degree C 40
Sample input output *40*
Algorithm *40*
Flowchart *40*
Python code *41*
Python code explanation *41*
C code *42*
C code explanation *42*

Program 1.16: Convert inch to feet 42
Sample input output *42*
Algorithm *43*
Flowchart *43*
Python code *43*
Python code explanation *44*
C code *44*
C code explanation *44*
Program 1.17: Find area of a rectangle 45
Sample input output *45*
Flowchart *45*
Python code *46*
Python code explanation *46*
C code *46*
C code explanation *47*
Program 1.18: Find area of a circle 47
Sample input output *47*
Algorithm *47*
Flowchart *48*
Python code *48*
Python code explanation *49*
C code *49*
C code explanation *49*
Program 1.19: Find area of parallelogram 50
Sample input output *50*
Algorithm *50*
Flowchart *51*
Python code *51*
Python code explanation *51*
C code *52*
C code explanation *52*

Program 1.20: Find distance between two points 52
Sample input output 52
Algorithm 53
Flowchart 53
Python code 54
Python code explanation 55
C code 55
C code explanation 55
Program 1.21: Obtain logical truth table 56
Sample input output 56
Algorithm 56
Flowchart 56
Python code 57
Python code explanation 58
C code 58
C code explanation 59
Conclusion 59
Points to remember 59
Questions 60

2. Conditional Statements 61

Introduction 61
Structure 62
Objectives 62
Program 2.1: Find absolute value of a number 63
Sample input output 63
Algorithm 63
Flowchart 63
Python code 64
Python code explanation 64
C code 64
C code explanation 65

Program 2.2: Check if you are an adult? 65
Sample input output 66
Algorithm 66
Flowchart 66
Python code 67
Python code explanation 67
C code 67
C code explanation 67
Program 2.3: Check if number is even or odd 68
Sample input output 68
Algorithm 68
Flowchart 68
Python code 69
Python code explanation 69
C code 69
C code explanation 70
Program 2.4: Check type of integer 70
Sample input output 70
Algorithm 71
Flowchart 71
Python code 71
Python code explanation 72
C code 72
C code explanation 72
Program 2.5: Validate a triangle 73
Sample input output 73
Algorithm 73
Flowchart 74
Python code 74
Python code explanation 74
C code 75
C code explanation 75

Program 2.6: Check divisibility by 7 and 5 75
Sample input output *76*
Algorithm *76*
Flowchart *76*
Python code *77*
Python code explanation *77*
C code *77*
C code explanation *77*
Program 2.7: Check if character is a vowel or consonant 78
Sample input output *78*
Algorithm *78*
Flowchart *78*
Python code *79*
Python code explanation *79*
C code *79*
C code explanation *80*
Program 2.8: Check if you can sit in exam? 80
Sample input output *80*
Algorithm *81*
Flowchart *81*
Python code *82*
Python code explanation *82*
C code *82*
C code explanation *83*
Program 2.9: Check if you can sit in exam (extended)? 83
Sample input output *83*
Algorithm *84*
Flowchart *85*
Python code *85*
Python code explanation *85*
C code *86*
C code explanation *86*

Program 2.10: Who spent more money? 87
Sample input output 87
Algorithm 87
Flowchart 88
Python code 89
Python code explanation 89
C code 89
C code explanation 90
Program 2.11: Find type of triangle 90
Sample input output 90
Algorithm 91
Flowchart 91
Python code 92
Python code explanation 92
C code 92
C code explanation 93
Program 2.12: Check if year is a leap year 93
Sample input output 93
Algorithm 94
Flowchart 94
Python code 95
Python code explanation 95
C code 95
C code explanation 96
Program 2.13: Find largest number 96
Sample input output 96
Algorithm 96
Flowchart 97
Python code 97
Python code explanation 98
C code 98
C code explanation 99

Program 2.14: Design arithmetic calculator 99
Sample input output *99*
Algorithm *100*
Flowchart *100*
Python code *101*
Python code explanation *101*
C code *101*
C code explanation *102*
Program 2.15: Calculate the grade of a student 102
Sample input output *103*
Algorithm *103*
Flowchart *103*
Python code *104*
Python code explanation *105*
C code *105*
C code explanation *106*
Program 2.16: Find gross salary 106
Sample input output *107*
Algorithm *107*
Flowchart *107*
Python code *108*
Python code explanation *109*
C code *109*
C code explanation *110*
Program 2.17: Generate shopping bill 110
Sample input output *110*
Algorithm *110*
Flowchart *111*
Python code *111*
Python code explanation *112*
C code *112*
C code explanation *113*

Program 2.18: Generate shopping bill during sale 113
Sample input output *113*
Algorithm *113*
Flowchart *114*
Python code *114*
Python code explanation *115*
C code *115*
C code explanation *115*
Conclusion 116
Points to remember 116
Questions 116

3. Simple Loops 117

Introduction 117
Structure 118
Objectives 118
Program 3.1: Display `'*'` 118
Sample input output *119*
Algorithm *119*
Flowchart *119*
Python code *120*
Python code explanation *120*
C code *120*
C code explanation *121*
Program 3.2: Display natural numbers 121
Sample input output *121*
Algorithm *121*
Flowchart *122*
Python code *122*
Python code explanation *122*
C code *123*
C code explanation *123*

Program 3.3: Display square of natural numbers 123
Sample input output *123*
Algorithm *124*
Flowchart *124*
Python code *125*
Python code explanation *125*
C code *125*
C code explanation *125*
Program 3.4: Display cube of numbers 126
Sample input output *126*
Algorithm *126*
Flowchart *126*
Python code *127*
Python code explanation *127*
C code *128*
C code explanation *128*
Program 3.5: Display multiples 128
Sample input output *128*
Algorithm *129*
Flowchart *129*
Python code *129*
Python code explanation *130*
C code *130*
C code explanation *130*
Program 3.6: Find sum and average 131
Sample input output *131*
Algorithm *131*
Flowchart *131*
Python code *132*
Python code explanation *132*
C code *132*
C code explanation *133*

Program 3.7: Find sum of sqaure 133
Sample input output *133*
Algorithm *133*
Flowchart *134*
Python code *134*
Python code explanation *135*
C code *135*
C code explanation *135*
Program 3.8: Find factorial 136
Sample input output *136*
Algorithm *136*
Flowchart *136*
Python code *137*
Python code explanation *137*
C code *137*
C code explanation *138*
Program 3.9: Find 2 raise to the power n 138
Sample input output *138*
Algorithm *139*
Flowchart *139*
Python code *139*
Python code explanation *140*
C code *140*
C code explanation *141*
Program 3.10: Find x raise to the power n 141
Sample input output *141*
Algorithm *141*
Flowchart *142*
Python code *142*
Python code explanation *142*
C code *142*
C code explanation *143*

Program 3.11: Display the multiplication table 143
Sample input output *143*
Algorithm *144*
Flowchart *144*
Python code *145*
Python code explanation *145*
C code *145*
C code explanation *146*
Program 3.12: Find sum of series 1 146
Sample input output *146*
Algorithm *146*
Flowchart *147*
Python code *147*
Python code explanation *148*
C code *148*
C code explanation *148*
Program 3.13: Find sum of series 2 149
Sample input output *149*
Algorithm *149*
Flowchart *149*
Python code *150*
Python code explanation *150*
C code *151*
C code explanation *151*
Program 3.14: Find sum of series 3 151
Sample input output *151*
Algorithm *152*
Flowchart *152*
Python code *153*
Python code explanation *153*
C code *153*
C code explanation *154*

Program 3.15: Find sum of n elements 154
Sample input output *154*
Algorithm *155*
Flowchart *155*
Python code *156*
Python code explanation *156*
C code *156*
C code explanation *157*
Program 3.16: Display Fibonacci series 157
Sample input output *157*
Algorithm *157*
Flowchart *158*
Python code *158*
Python code explanation *158*
C code *159*
C code explanation *159*
Conclusion 159
Points to remember 160
Questions 160

4. Complex Loops 161

Introduction 161
Structure 161
Objectives 162
Program 4.1: Find sum and check if even 162
Sample input output *162*
Algorithm *162*
Flowchart *163*
Python code *163*
Python code explanation *164*
C code *164*
C code explanation *165*

Program 4.2: Display odd numbers 165
Sample input output *165*
Algorithm *166*
Flowchart *166*
Python code *167*
Python code explanation *167*
C code *167*
C code explanation *168*
Program 4.3: Count number of even numbers 168
Sample input output *168*
Algorithm *168*
Flowchart *169*
Python code *170*
Python code explanation *170*
C code *170*
C code explanation *171*
Program 4.4: Find sum of even numbers 171
Sample input output *171*
Algorithm *171*
Flowchart *172*
Python code *172*
Python code explanation *172*
C code *173*
C code explanation *173*
Program 4.5: Find sum of numbers divisble by 9 173
Sample input output *173*
Algorithm *173*
Flowchart *174*
Python code *174*
Python code explanation *175*
C code *175*
C code explanation *175*

Program 4.6: Difference of sum of even and odd numbers 176
Sample input output *176*
Algorithm *176*
Flowchart *176*
Python code *177*
Python code explanation *178*
C code *178*
C code explanation *178*
Program 4.7: Display list of divisor 179
Sample input output *179*
Algorithm *179*
Flowchart *179*
Python code *180*
Python code explanation *180*
C code *181*
C code explanation *181*
Program 4.8: Find max 1 181
Sample input output *181*
Algorithm *182*
Flowchart *182*
Python code *183*
Python code explanation *183*
C code *183*
C code explanation *184*
Program 4.9: Find max 2 184
Sample input output *184*
Algorithm *184*
Flowchart *185*
Python code *185*
Python code explanation *186*
C code *186*
C code explanation *186*

Program 4.10: Count number of consonants 187
Sample input output *187*
Algorithm *187*
Flowchart *188*
Python code *188*
Python code explanation *189*
C code *189*
C code explanation *190*
Program 4.11: Check if a number is prime 190
Sample input output *190*
Algorithm *191*
Flowchart *191*
Python code *192*
Python code explanation *193*
C code *193*
C code explanation *194*
Program 4.12: Find LCM 194
Sample input output *194*
Algorithm *195*
Flowchart *195*
Python code *196*
Python code explanation *197*
C code *197*
C code explanation *198*
Program 4.13: Find HCF 199
Sample input output *199*
Algorithm *199*
Flowchart *200*
Python code *200*
Python code explanation *201*
C code *201*
C code explanation *202*

Program 4.14: Search a number 202
Sample input output 202
Algorithm 202
Flowchart 203
Python code 204
Python code explanation 204
C code 205
C code explanation 205
Program 4.15: Handling list/ arrays 206
Sample input output 206
Algorithm 206
Flowchart 207
Python code 207
Python code explanation 208
C code 208
C code explanation 209
Program 4.16: Frequency of occurrence of elements 210
Sample input output 210
Algorithm 210
Flowchart 211
Python code 212
Python code explanation 212
C code 213
C code explanation 214
Program 4.17: Find length of a string 214
Sample input output 214
Algorithm 214
Flowchart 215
Python code 215
Python code explanation 215
C code 216
C code explanation 216

Program 4.18: Check if string is a palindrome 216
Sample input output *217*
Algorithm *217*
Flowchart *218*
Python code *218*
Python code explanation *219*
C code *219*
C code explanation *220*
Program 4.19: Display even indexed characters 220
Sample input output *220*
Algorithm *220*
Flowchart *221*
Python code *221*
Python code explanation *222*
C code *222*
C code explanation *222*
Program 4.20: Concatenate two strings 223
Sample input output *223*
Algorithm *223*
Flowchart *223*
Python code *224*
Python code explanation *225*
C code *225*
C code explanation *225*
Conclusion 226
Points to remember 226
Questions 226

5. Complex Problem Solving 229

Introduction 229
Structure 229
Objectives 230

Program 5.1: Find sum of digits of a number 230
Sample input output 230
Algorithm 230
Flowchart 231
Python code 232
Python code explanation 232
C code 232
C code explanation 233
Program 5.2: Reverse the number 233
Sample input output 233
Algorithm 233
Flowchart 234
Python code 234
Python code explanation 235
C code 235
C code explanation 235
Program 5.3: Find all Armstrong numbers 236
Sample input output 236
Algorithm 236
Flowchart 237
Python code 237
Python code explanation 238
C code 238
C code explanation 239
Program 5.4: Find n based on a condition 239
Sample input output 239
Algorithm 240
Flowchart 240
Python code 241
Python code explanation 241
C code 241
C code explanation 242

Program 5.5: Obtain list having prime numbers 242
Sample input output *242*
Algorithm *242*
Flowchart *243*
Python code *243*
Python code explanation *244*
C code *244*
C code explanation *245*
Program 5.6: Remove duplicate items 245
Sample input output *245*
Algorithm *246*
Flowchart *246*
Python code *247*
Python code explanation *247*
C code *247*
C code explanation *248*
Program 5.7: Sort the numbers 248
Sample input output *248*
Algorithm *248*
Flowchart *249*
Python code *250*
Python code explanation *250*
C code *250*
C code explanation *251*
Program 5.8: Print matrix 251
Sample input output *251*
Algorithm *252*
Flowchart *253*
Python code *254*
Python code explanation *254*
C code *254*
C code explanation *255*

Program 5.9: Add two matrics 255
Sample input output 255
Algorithm 256
Flowchart 256
Python code 258
Python code explanation 258
C code 259
C code explanation 260
Program 5.10: Check if matrix is upper triangular 260
Sample input output 260
Algorithm 261
Flowchart 261
Python code 262
Python code explanation 263
C code 263
C code explanation 264
Program 5.11: Check if matrix is symmetric 264
Sample input output 264
Algorithm 265
Flowchart 265
Python code 266
Python code explanation 267
C code 267
C code explanation 268
Program 5.12: Print 2-D pattern 1 268
Sample input output 268
Algorithm 269
Flowchart 269
Python code 270
Python code explanation 270
C code 270
C code explanation 270

Program 5.13: Print 2-D pattern 2 271
Sample input output 271
Algorithm 271
Flowchart 272
Python code 272
Python code explanation 272
C code 273
C code explanation 273
Program 5.14: Print 2-D pattern 3 273
Sample input output 273
Algorithm 274
Flowchart 274
Python code 275
Python code explanation 275
C code 275
C code explanation 275
Program 5.15: Print 2-D pattern 4 276
Sample input output 276
Algorithm 276
Flowchart 276
Python code 277
Python code explanation 277
C code 277
C code explanation 278
Program 5.16: Print 2-D pattern 5 278
Sample input output 278
Algorithm 279
Flowchart 279
Python code 280
Python code explanation 280
C code 280
C code explanation 281

Program 5.17: Print 2-D pattern 6 281
Sample input output *281*
Algorithm *281*
Flowchart *282*
Python code *282*
Python code explanation *283*
C code *283*
C code explanation *283*
Program 5.18: Print 2-D pattern 7 284
Sample input output *284*
Algorithm *284*
Flowchart *285*
Python code *285*
Python code explanation *285*
C code *286*
C code explanation *286*
Program 5.19: Print 2-D pattern 8 286
Sample input output *287*
Flowchart *287*
Python code *288*
Python code explanation *288*
C code *289*
C code explanation *289*
Program 5.20: Print 2-D pattern 9 289
Sample input output *290*
Algorithm *290*
Flowchart *290*
Python code *291*
Python code explanation *291*
C code *292*
C code explanation *292*

Conclusion 292
Points to remember 293
Questions 293

6. Real World Problems 295

Introduction 295
Structure 295
Objectives 296
Program 6.1: Optimize book storage 296
Sample input output 296
Algorithm 297
Flowchart 297
Python code 298
Python code explanation 298
C code 298
C code explanation 299
Program 6.2: Identify living creatures 299
Sample input output 299
Algorithm 300
Flowchart 300
Python code 301
Python code explanation 302
C code 302
C code explanation 303
Program 6.3: Identify objects 303
Sample input output 303
Algorithm 304
Flowchart 305
Python code 305
Python code explanation 306
C code 306
C code explanation 307

Program 6.4: Rock-paper-scissors 307
Sample input output *308*
Algorithm *308*
Flowchart *309*
Python code *310*
Python code explanation *310*
C code *310*
C code explanation *311*
Program 6.5: Print hotel bill 311
Sample input output *311*
Algorithm *312*
Flowchart *313*
Python code *314*
Python code explanation *314*
C code *315*
C code explanation *316*
Program 6.6: Find fewest denominations 317
Sample input output *317*
Algorithm *317*
Flowchart *318*
Python code *318*
Python code explanation *319*
C code *319*
C code explanation *320*
Program 6.7: Assign class rank 320
Sample input output *320*
Algorithm *321*
Flowchart *321*
Python code *322*
Python code explanation *323*
C code *323*
C code explanation *324*

Program 6.8: Minimize the difference 325
Sample input output *325*
Algorithm *326*
Flowchart *326*
Python code *328*
Python code explanation *328*
C code *329*
C code explanation *330*
Program 6.9: Convert decimals to different bases 330
Sample input output *331*
Algorithm *331*
Flowchart *332*
Python code *333*
Python code explanation *333*
C code *333*
C code explanation *335*
Program 6.10: Password validation and encryption 335
Sample input output *335*
Algorithm *336*
Flowchart *337*
Python code *339*
Python code explanation *340*
C code *340*
C code explanation *344*
Conclusion 345
Points to remember 345
Questions 345

Index 347-356

Chapter 1
Simple Input Output Program

Introduction

The ability to process input and generate output is a fundamental aspect of programming. In this chapter, we delve into the world of simple input-output programs. It comprehensively explains basic algorithms, flowcharts, Python, and C code implementations. These programs allow users to grasp essential concepts such as user input, data manipulation, and result display. This chapter will be a building block for developing more complex applications in further book chapters.

Whether you are a novice programmer seeking to strengthen your fundamental abilities or an experienced developer looking for practical examples, this chapter offers a diverse range of simple programs to explore. A basic understanding of the program would be built by studying the algorithms and visualizing the flowcharts. In addition, the Python and C code will give you valuable insights into the core principles of input-output operations in these languages. Overall, the skills necessary to solve real-world programming challenges would be gained. The basic building blocks of a flowchart are shown in *Figure 1.1*:

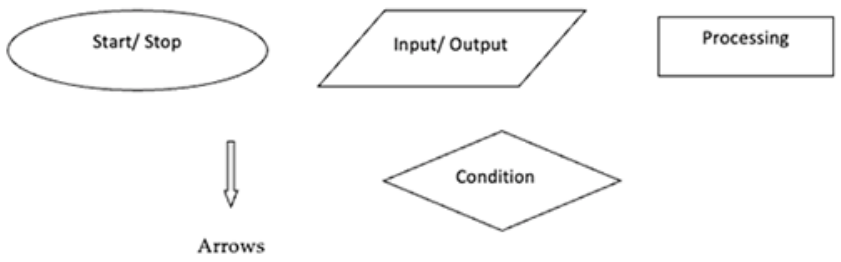

***Figure 1.1**: The basic building blocks of a flowchart*

Let us embark on this journey through simple input-output programs and discover the power of these fundamental building blocks.

Structure

In this chapter, there are a total of 21 programs, and four sub-topics complement each program. The following topics will be covered with each program:

- Understand the program objective with sample input and outputs. It will give the user an understanding of what kind of user input is expected and what result will be projected for a particular program.
- Understand the program with an algorithm. This will help the user to understand the methodology behind writing the sequence of steps required to solve a program.
- Visualize the program flow with flowcharts. Program objectives can be understood easily with the flowchart representation. Users will understand which symbols to use in which context throughout different program's flowchart representations.
- Python source code of each program. Each program's source code in Python programming language is provided to the user so the user can understand the basic operations of the language. An explanation of each code is also provided.
- C implementation of each program. Users can learn the basic concepts of C language with the provided C language code and its explanation for every program.

Objectives

In this chapter, the reader will learn the fundamentals of input-output operations and their significance in program development. We will discuss designing algorithms to perform basic input and output operations. We will acquire the skills to create visual representations of program flow using flowcharts and the ability to implement simple input-output programs using Python and C programming languages. With the sample codes, users can gain hands-on experience. We will also gain problem-solving skills related to simple basic problems, such as accepting user input, performing calculations, and displaying results.

Program 1.1: Print your name

Take your name as input and print **`Hello name`**.

Sample input output

Test case #1:

Input: `Neo`

Output: **`Hello Neo`**

Test case #2:

Input: `Sam`

Output: **`Hello Sam`**

Test case #3:

Input: `John Will`

Output: **`Hello John Will`**

Algorithm

To run the above program, the following steps are needed to be followed:

Step 1. **`Start`**

Step 2. **`Enter your name in name`**

Step 3. **`Print "Hello" name`**

Step 4. **`Stop`**

Flowchart

Figure 1.2 illustrates the program in the form of a flowchart:

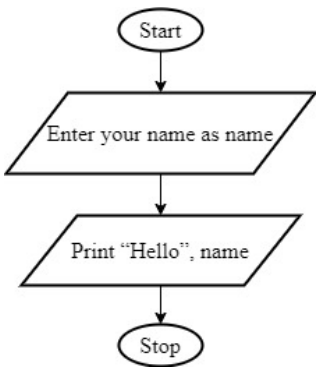

***Figure 1.2**: Flowchart illustrating program 1.1*

Python code

Python code for the *program 1.1* is as follows:

```
1. # read a input from the user
2. name= input("Enter your name: ")
3. # print the name on the screen
4. print("Hello ", name)
```

Python code explanation

The program prompts the user to enter the name and then prints a greeting message containing the name.

`name = input("Enter your name: ")`: This line prompts the user to enter the name by printing the string **`Enter your name:`**. The **`input()`** function is used to read the user's input as a string and assign it to the variable name.

`print("Hello", name)`: This line uses the **`print()`** function to print the string **`Hello`** followed by the value of the name variable.

Python is an indentation-sensitive language. Indentation is used to mark the block of code.

Any statement preceded by **`#`** is a comment which does not execute. These comments are ignored by the compiler and used only for the user to understand the code.

C code

The C code of *program 1.1* is specified below:

```
1. #include <stdio.h>
2.
3. int main()  //The execution of C program starts with main function
4. {
5.     char name[100]; //declared a variable of character type of size 10
6.     printf("Enter your name: ");
7.     gets(name);  // read a input from the user
8.     printf("Hello %s", name);   // print the name on the screen
9.     return 0;   // The main() function returns an integer value 0
10. }
```

C code explanation

`stdio.h` is a header file that should be included in the program to use various input-output functions. **`Main()`** function serves as the starting point for program execution. It is mandatory to write a main function in every C program. Here, **`int`** is written before **`main()`**. This means that the function **`main()`** will return an integer value after completing the program's execution. The last statement `return 0` returns a value 0 on successful execution of the program.

A variable **`name`** is declared as an array of 100 characters. **`gets()`** is an input function used to read a string. Referring to test *case #3*, **`gets()`** function also allows the space between words. Also, C uses **`%s`** as the format specifier for strings. Finally, we use **`printf()`** **`function`** to output the greeting message with the user's name.

Unlike Python, indentation is not important in C language. Each block of code is enclosed between parentheses.

Program 1.2: Print your lucky number

Take a number as input and print the number as **`Number entered by you is: number`**.

Sample input output

Test case #1:

Input: `10`

Output: **`Number entered by you is: 10`**

Test case #2:

Input: `55`

Output: **`Number entered by you is: 55`**

Algorithm

The below steps illustrate the working of *program 1.2*:

Step 1. **`Start`**

Step 2. **`Enter a number in n`**

Step 3. **`Print "Number entered by you is : " n`**

Step 4. **`Stop`**

Flowchart

Figure 1.3 depicts the program in the form of a flowchart as follows: