

Code Factory

100+ Programming Solutions

Navigating the wonders beyond binary brilliance

Shikha Jain
Kavita Pandey



www.bpbonline.com

First Edition 2024

Copyright © BPB Publications, India

ISBN: 978-93-55519-115

All Rights Reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means or stored in a database or retrieval system, without the prior written permission of the publisher with the exception to the program listings which may be entered, stored and executed in a computer system, but they can not be reproduced by the means of publication, photocopy, recording, or by any electronic and mechanical means.

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.

To View Complete
BPB Publications Catalogue
Scan the QR Code:



www.bpbonline.com

Kup książkę

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.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

business@bpbonline.com for more details.

At www.bpbonline.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

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
<i>Sample input output.....</i>	3
<i>Algorithm</i>	3
<i>Flowchart.....</i>	3
<i>Python code</i>	4
<i>Python code explanation</i>	4
<i>C code</i>	4
<i>C code explanation</i>	5
Program 1.2: Print your lucky number	5
<i>Sample input output.....</i>	5
<i>Algorithm</i>	5
<i>Flowchart.....</i>	5
<i>Python code</i>	6
<i>Python code explanation</i>	6
<i>C code</i>	6
<i>C code explanation</i>	7
Program 1.3: Print your details	7
<i>Sample input output.....</i>	7
<i>Algorithm</i>	8
<i>Flowchart.....</i>	8
<i>Python code</i>	8
<i>Python code explanation</i>	9
<i>C code</i>	9
<i>C code explanation</i>	10

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

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

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

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

Program 1.20: Find distance between two points	52
<i>Sample input output</i>	52
<i>Algorithm</i>	53
<i>Flowchart</i>	53
<i>Python code</i>	54
<i>Python code explanation</i>	55
<i>C code</i>	55
<i>C code explanation</i>	55
Program 1.21: Obtain logical truth table	56
<i>Sample input output</i>	56
<i>Algorithm</i>	56
<i>Flowchart</i>	56
<i>Python code</i>	57
<i>Python code explanation</i>	58
<i>C code</i>	58
<i>C code explanation</i>	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
<i>Sample input output</i>	63
<i>Algorithm</i>	63
<i>Flowchart</i>	63
<i>Python code</i>	64
<i>Python code explanation</i>	64
<i>C code</i>	64
<i>C code explanation</i>	65

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

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

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

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

Program 2.18: Generate shopping bill during sale	113
<i>Sample input output</i>	113
<i>Algorithm</i>	113
<i>Flowchart</i>	114
<i>Python code</i>	114
<i>Python code explanation</i>	115
<i>C code</i>	115
<i>C code explanation</i>	115
Conclusion	116
Points to remember	116
Questions.....	116
3. Simple Loops	117
Introduction	117
Structure	118
Objectives	118
Program 3.1: Display '*'	118
<i>Sample input output</i>	119
<i>Algorithm</i>	119
<i>Flowchart</i>	119
<i>Python code</i>	120
<i>Python code explanation</i>	120
<i>C code</i>	120
<i>C code explanation</i>	121
Program 3.2: Display natural numbers	121
<i>Sample input output</i>	121
<i>Algorithm</i>	121
<i>Flowchart</i>	122
<i>Python code</i>	122
<i>Python code explanation</i>	122
<i>C code</i>	123
<i>C code explanation</i>	123

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

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

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

Program 3.15: Find sum of n elements	154
<i>Sample input output</i>	154
<i>Algorithm</i>	155
<i>Flowchart</i>	155
<i>Python code</i>	156
<i>Python code explanation</i>	156
<i>C code</i>	156
<i>C code explanation</i>	157
Program 3.16: Display Fibonacci series	157
<i>Sample input output</i>	157
<i>Algorithm</i>	157
<i>Flowchart</i>	158
<i>Python code</i>	158
<i>Python code explanation</i>	158
<i>C code</i>	159
<i>C code explanation</i>	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
<i>Sample input output</i>	162
<i>Algorithm</i>	162
<i>Flowchart</i>	163
<i>Python code</i>	163
<i>Python code explanation</i>	164
<i>C code</i>	164
<i>C code explanation</i>	165

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

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

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

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

Program 4.18: Check if string is a palindrome	216
<i>Sample input output</i>	217
<i>Algorithm</i>	217
<i>Flowchart</i>	218
<i>Python code</i>	218
<i>Python code explanation</i>	219
<i>C code</i>	219
<i>C code explanation</i>	220
Program 4.19: Display even indexed characters	220
<i>Sample input output</i>	220
<i>Algorithm</i>	220
<i>Flowchart</i>	221
<i>Python code</i>	221
<i>Python code explanation</i>	222
<i>C code</i>	222
<i>C code explanation</i>	222
Program 4.20: Concatenate two strings	223
<i>Sample input output</i>	223
<i>Algorithm</i>	223
<i>Flowchart</i>	223
<i>Python code</i>	224
<i>Python code explanation</i>	225
<i>C code</i>	225
<i>C code explanation</i>	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
<i>Sample input output</i>	230
<i>Algorithm</i>	230
<i>Flowchart</i>	231
<i>Python code</i>	232
<i>Python code explanation</i>	232
<i>C code</i>	232
<i>C code explanation</i>	233
Program 5.2: Reverse the number	233
<i>Sample input output</i>	233
<i>Algorithm</i>	233
<i>Flowchart</i>	234
<i>Python code</i>	234
<i>Python code explanation</i>	235
<i>C code</i>	235
<i>C code explanation</i>	235
Program 5.3: Find all Armstrong numbers	236
<i>Sample input output</i>	236
<i>Algorithm</i>	236
<i>Flowchart</i>	237
<i>Python code</i>	237
<i>Python code explanation</i>	238
<i>C code</i>	238
<i>C code explanation</i>	239
Program 5.4: Find n based on a condition	239
<i>Sample input output</i>	239
<i>Algorithm</i>	240
<i>Flowchart</i>	240
<i>Python code</i>	241
<i>Python code explanation</i>	241
<i>C code</i>	241
<i>C code explanation</i>	242

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

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

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

Program 5.17: Print 2-D pattern 6	281
<i>Sample input output</i>	281
<i>Algorithm</i>	281
<i>Flowchart</i>	282
<i>Python code</i>	282
<i>Python code explanation</i>	283
<i>C code</i>	283
<i>C code explanation</i>	283
Program 5.18: Print 2-D pattern 7	284
<i>Sample input output</i>	284
<i>Algorithm</i>	284
<i>Flowchart</i>	285
<i>Python code</i>	285
<i>Python code explanation</i>	285
<i>C code</i>	286
<i>C code explanation</i>	286
Program 5.19: Print 2-D pattern 8	286
<i>Sample input output</i>	287
<i>Flowchart</i>	287
<i>Python code</i>	288
<i>Python code explanation</i>	288
<i>C code</i>	289
<i>C code explanation</i>	289
Program 5.20: Print 2-D pattern 9	289
<i>Sample input output</i>	290
<i>Algorithm</i>	290
<i>Flowchart</i>	290
<i>Python code</i>	291
<i>Python code explanation</i>	291
<i>C code</i>	292
<i>C code explanation</i>	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
<i>Sample input output</i>	296
<i>Algorithm</i>	297
<i>Flowchart</i>	297
<i>Python code</i>	298
<i>Python code explanation</i>	298
<i>C code</i>	298
<i>C code explanation</i>	299
Program 6.2: Identify living creatures	299
<i>Sample input output</i>	299
<i>Algorithm</i>	300
<i>Flowchart</i>	300
<i>Python code</i>	301
<i>Python code explanation</i>	302
<i>C code</i>	302
<i>C code explanation</i>	303
Program 6.3: Identify objects	303
<i>Sample input output</i>	303
<i>Algorithm</i>	304
<i>Flowchart</i>	305
<i>Python code</i>	305
<i>Python code explanation</i>	306
<i>C code</i>	306
<i>C code explanation</i>	307

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

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

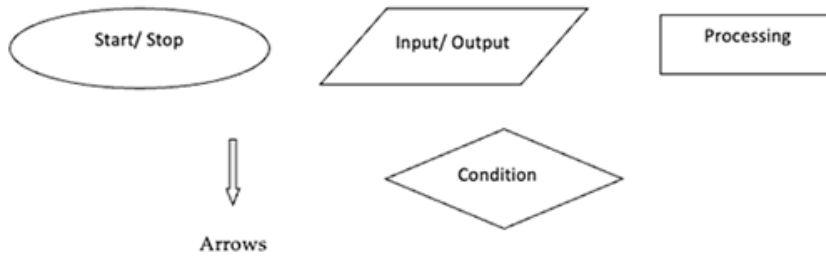


Figure 1.1: The basic building blocks of a flowchart

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:

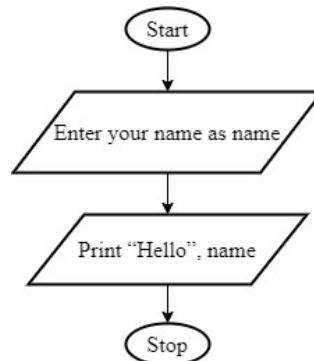


Figure 1.2: Flowchart illustrating program 1.1

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: