OXFORD
HIGHER EDUCATION

PYTHON PROGRAMMING

USING PROBLEM SOLVING APPROACH

REEMA THAREJA

PYTHON PROGRAMMING

USING PROBLEM SOLVING APPROACH

Reema Thareja
Assistant Professor
Department of Computer Science
Shyama Prasad Mukherji College for Women
University of Delhi.

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trademark of
Oxford University Press in the UK and in certain other countries.

Published in India by
Oxford University Press
Ground Floor, 2/11, Ansari Road, Daryaganj, New Delhi 110002, India

First published in 2017
Fourth impression 2018

ISBN-13: 978-0-19-948017-3
ISBN-10: 0-19-948017-6

Typeset in Times New Roman
by Ideal Publishing Solutions, Delhi
Printed in India by Gopsons Papers Ltd., Sivakasi

Cover image: Keo / Shutterstock

Third-party website addresses mentioned in this book are provided
by Oxford University Press in good faith and for information only.
Oxford University Press disclaims any responsibility for the material contained therein.

I dedicate this book to my family and my uncle, Mr B.L. Theraja, who is a well-known author

Preface

Computers are so widely used in our day-to-day lives that imagining a life without them has become almost impossible. They are not only used by professionals but also by children for interactively learning lessons, playing games, and doing their homework. Applications of the computer and its users are increasing by the day. Learning computer and programming basics is a stepping stone to having an insight into how the machines work. Once the reader is aware of the basic terminologies and problem solving strategies that are commonly used in computer science, he/she can then go on to develop efficient and effective computer programs that may help solve a user's problems.

Since computers cannot understand human languages, special programming languages are designed for this purpose. Python is one such language. It is an open-source, easy, high-level, interpreted, interactive, object-oriented and reliable language that uses English-like words. It can run on almost all platforms including Windows, Mac OS X, and Linux. Python is also a versatile language that supports development of a wide range of applications ranging from simple text processing to WWW browsers to games. Moreover, programmers can embed Python within their C, C++, COM, ActiveX, CORBA, and Java programs to give 'scripting' capabilities to the users.

Python uses easy syntax and short codes as well as supports multiple programming paradigms, including object oriented programming, functional Python programming, and parallel programming models. Hence, it has become an ideal choice for the programmers and even the novices in computer programming field find it easy to learn and implement. It has encompassed a huge user base that is constantly growing and this strength of Python can be understood from the fact that it is the most preferred programming language in companies such as Nokia, Google, YouTube, and even NASA for its easy syntax and short codes.

About the Book

This book is designed as a textbook to cater to the requirements of the first course in Python programming. It is suited for undergraduate degree students of computer science engineering and information technology as well as postgraduate students of computer applications. The objective of this book is to introduce the students to the fundamentals of computers and the concepts of Python programming language, and enable them to apply these concepts for solving real-world problems.

The book is organized into 12 chapters that provide comprehensive coverage of all the relevant topics using simple language. It also contains useful annexures to various chapters including for additional information. Case studies and appendices are also provided to supplement the text.

Programming skill is best developed by rigorous practice. Keeping this in mind, the book provides a number of programming examples that would help the reader learn how to write efficient programs. These programming examples have already been complied and tested using Python 3.4.1 version and can be also executed on Python 3.5 and 3.6 versions. To further enhance the understanding of the subject, there are numerous chapter-end exercises provided in the form of objective-type questions, review questions, and programming problems.

Key Features of the Book

The following are the important features of the book:

- Offers **simple** and **lucid** treatment of concepts supported with illustrations for easy understanding.
- Contains **separate chapters** on Strings, Files, Exception Handling, and Operator Overloading

- Provides **numerous programming examples** along with their outputs to help students master the art of writing efficient Python programs.
- Includes **notes** and **programming tips** to highlight the important concepts and help readers avoid common programming errors.
- Offers **rich chapter-end pedagogy** including plenty of objective-type questions (with answers), review questions, programming and debugging exercises to facilitate revision and practice of concepts learnt.
- Includes **7 Annexures** and **5 appendices** covering types of operating systems, differences between Python 2.x and 3.x, installing Python, debugging and testing, iterators, generators, getters, setters, @property, @deleter, Turtle graphics, plotting graphs, multi-threading, GUI and Web Programming provided to supplement the text. Exercises are also added at the end of several annexures and appendices.
- Provides **case studies** on creating calculator, calendar, hash files, compressing strings and files, tower of Hanoi, image processing, shuffling a deck of cards, and mail merge that are linked to various chapters to demonstrate the application of concepts.
- Point-wise **summary** and **glossary** of keyterms to aid quick recapitulation to concepts.

Organization of the Book

The book contains 12 chapters, 7 annexures, 8 case studies, and 5 appendices. The details of the book are presented as follows.

Chapter 1 provides an introduction to computer hardware and software. It covers the concept of memory and its storage units, application software, and system software. The chapter provides an insight into the different stages of software development life cycle and discusses the various strategies used for problem solving. Topics such as algorithms, flowcharts, and pseudocodes are discussed in this chapter.

Annexure 1 given after Chapter 1 discusses the classification of operating systems.

Chapter 2 discusses about programming languages and their evolution through generations. It describes different programming paradigms, features of OOP, and merits and demerits of object oriented programming languages. The chapter also gives a comparative study Python and other OOP languages, and highlights the applications of OOP paradigm.

Chapter 3 details the history, important features and applications of Python. It also presents the various building blocks (such as keywords, identifiers, constants variables, operators, expressions, statements and naming conventions) supported by the language.

The chapter is followed 3 annexures – ***Annexure 2*** provides instructions for installing Python. ***Annexure 3*** provides the comparison between Python 2.x and Python 3.x versions. ***Annexure 4*** discusses testing and debugging of Python programs using IDLE.

Chapter 4 deals with the different types of decision control statements such as selection/ conditional branching, iterative, break, continue, pass, and else statements.

Case studies 1 *and* ***2*** on simple calculator and generating a calendar show the implementation of concepts discussed in Chapters 3 and 4.

Chapter 5 provides a detailed explanation of defining and calling functions. It also explains the important concepts such as variable length arguments, recursive functions, modules, and packages in Python.

Annexure 5 explains how functions are objects in Python. ***Case studies 3*** and ***4*** on tower of Hanoi and shuffling a deck of cards demonstrates the concepts of functions as well as recursion.

Chapter 6 unleashes the concept of strings. The chapter lays special focus on the operators used with strings, slicing operation, built-in string methods and functions, comparing and iterating through strings, and the `string` module.

Chapter 7 discusses how data can be stored in files. The chapter deals with opening, processing (like reading, writing, appending, etc.), and closing of files though a Python program. These files are handled in text mode as well as binary mode for better clarity of the concepts. The chapter also explains the concept of file, directory, and the `os` module.

Case studies 5, 6, and *7* on creating a hash file, mail merge, and finding the resolution of an image demonstrate the applications of concepts related strings and file handling.

Chapter 8 details the different data structures (such as list, tuple, dictionary, sets, etc.) that are extensively used in Python. It deals with creating, accessing, cloning, ad updating of lists as well as list methods and functions. It also describes functional programming and creating, accessing, and updating tuples. It also includes the concepts related to sets, dictionaries, nested lists, nested tuples, nested sets, nested dictionaries, list comprehensions, and dictionary comprehensions.

Annexure 6 discusses the concepts of iterator and generator.

Chapter 9 introduces the concept of classes, objects, public and private classes, and instance variables. It also talks about special methods, built-in attributes, built-in methods, garbage collection, class method, and static method.

Annexure 7 discusses the `getter` and `setter` methods as well as `@property` and `@deleter` decorators facilitate data encapsulation in Python.

Chapter 10 introduces inheritance and its various forms. It gives a detail explanation on method overriding, containership, abstract class, interface, and metaclass.

Chapter 11 is all about overloading arithmetic and logical operators. It also discusses reverse adding and overriding `__getitem__()`, `__setitem__()`, and `__call__()` methods, `in` operator, as well as other miscellaneous functions.

Chapter 12 elucidates the concepts of exception handling that can be used to make your programs robust. Concepts such as try, except, and finally blocks, raising and re-raising exceptions, built-in and user-defined exceptions, assertions, and handling invoked functions, used for handling exceptions are demonstrated in this chapter.

Case study 8 shows how to compress strings and files using exception handling concepts.

The **5 appendices** included in the book discuss about multi-threading, GUI programming, usage of Turtle graphics, plotting graphs and web programming in Python.

Online Resources

For the benefit of faculty and students reading this book, additional resources are available online at india.oup.com/orcs//9780199480173

For Faculty

- Solutions manual (for programming exercises)
- Chapter-wise PPTs

For Students

- Lab exercises
- Test generator
- Projects
- Solutions to find the output and error exercises
- Model question papers
- Extra reading material on number systems, unit testing in Python, sorting and searching methods, network programming, event-driven programming and accessing databases using Python

Acknowledgements

The writing of this textbook was a mammoth task for which a lot of help was required from many people. Fortunately, I have had wholehearted support of my family, friends, and fellow members of the teaching staff and students at Shyama Prasad Mukherji College, New Delhi.

My special thanks would always go to my parents, Mr Janak Raj Thareja and Mrs Usha Thareja, and my siblings, Pallav, Kimi, and Rashi, who were a source of abiding inspiration and divine blessings for me. I am especially thankful to my son, Goransh, who has been very patient and cooperative in letting me realize my dreams. My sincere thanks go to my uncle, Mr B.L. Theraja, for his inspiration and guidance in writing this book.

I would like to acknowledge the technical assistance provided to me by Mr Mitul Kapoor. I would like to thank him for sparing out his precious time to help me to design and test the programs.

I would like to express my gratitude to the reviewers for their valuable suggestions and constructive feedback that helped in improving the book.

Prof. M V S V Kiranmai
University College of Engineering, JNTU Kakinada
Dr Nagender Kumar Suryadevara
Geethanjali College of Engineering and Technology, Hyderabad
Dr Vipul Kumar Mishra
School of Engineering, Bennett University, Greater Noida, U.P.
Dr G Shobha
R V College of Engineering, Bengaluru
Prof. Priyang P Bhatt
GH Patel College of Engineering and Technology, Vallabh Vidyanagar, Gujarat
Prof. Karthick Nanmaran
School of Computing, SRM University, Chennai

Last but not the least, I would like to thank the editorial team at Oxford University Press, India for their help and support over the past few years.

Comments and suggestions for the improvement of the book are welcome. Please send them to me at reemathareja@gmail.com.

Reema Thareja

Brief Contents

Preface *iv*
Detailed Contents *ix*

1 Introduction to Computers and Problem Solving Strategies 1
Annexure 1 — Types of Operating Systems *52*

2 Introduction to Object Oriented Programming (OOP) 63

3 Basics of Python Programming 83
Annexure 2 — Installing Python *125*
Annexure 3 — Comparison between Python 2.x and Python 3.x Versions *127*
Annexure 4 — Testing and Debugging *130*

4 Decision Control Statements 137
Case Study 1 — Simple Calculator *180*
Case Study 2 — Generating a Calendar *183*

5 Functions and Modules 185
Annexure 5 — Functions as Objects *237*
Case Study 3 — Tower of Hanoi *239*
Case Study 4 — Shuffling a Deck of Cards *241*

6 Python Strings Revisited 242

7 File Handling 289
Case Study 5 — Creating a Hash File (or a message digest of a file) *317*
Case Study 6 — Mail Merge Program *319*
Case Study 7 — Finding Resolution of an Image *321*

8 Data Structures 322
Annexure 6 — Iterator and Generator *392*

9 Classes and Objects 400
Annexure 7 — Getters, Setters, `@property`*, and* `@deleter` *432*

10 Inheritance 436

11 Operator Overloading 460

12 Error and Exception Handling 480
Case Study 8 — Compressing String and Files *505*

Appendix A — Multi-threading *508*
Appendix B — GUI Programming With `tkinter` *Package* *514*
Appendix C — Simple Graphics Using Turtle *524*
Appendix D — Plotting Graphs in Python *530*
Appendix E — CGI/Web Programming Using Python *536*

Detailed Contents

Preface iv
Brief Contents viii

1 Introduction to Computers and Problem Solving Strategies 1

1.1 Introduction 1
1.2 What is a Computer? 1
1.3 History of Computers 2
1.4 Characteristics of Computers 4
1.5 Classification of Computers 5
1.6 Basic Applications of Computers 9
1.7 Stored Program Concept 11
1.7.1 Types of Stored Program Computers 12
1.8 Components and Functions of a Computer System 12
1.9 Concept of Hardware and Software 14
1.9.1 Hardware 14
1.9.2 Software 14
1.10 Central Processing Unit (CPU): Basic Architecture 14
1.11 Input and Output devices 16
1.12 Computer Memory 17
1.12.1 Memory Hierarchy 18
1.12.2 Primary Memory 18
1.12.3 Secondary Storage Devices 20
1.13 Classification of Computer Software 20
1.13.1 System Software 22
1.13.2 Application Software 27
1.14 Representation of Data: Bits and Bytes 29
1.15 Problem Solving Strategies 31
1.16 Program Design Tools: Algorithms, Flowcharts, Pseudocodes 32
1.16.1 Algorithms 32
1.16.2 Flowcharts 37
1.16.3 Pseudocodes 39
1.17 Types of Errors 41
1.18 Testing and Debugging Approaches 42
Annexure 1 — Types of Operating Systems 52

2 Introduction to Object Oriented Programming (OOP) 63

2.1 Computer Programming and Programming Languages 63
2.2 Generations of Programming Languages 64
2.2.1 First Generation: Machine Language 64
2.2.2 Second Generation: Assembly Language 65
2.2.3 Third Generation: High-level Language 66
2.2.4 Fourth Generation: Very High-level Languages 67
2.2.5 Fifth Generation Programming Language 68
2.3 Programming Paradigms 69
2.3.1 Monolithic Programming 69
2.3.2 Procedural Programming 69
2.3.3 Structured Programming 70
2.3.4 Object Oriented Programming (OOP) 71
2.4 Features of Object Oriented Programming 72
2.4.1 Classes 72
2.4.2 Objects 73
2.4.3 Method and Message Passing 73
2.4.4 Inheritance 74
2.4.5 Polymorphism 75
2.4.6 Containership 75
2.4.7 Reusability 75
2.4.8 Delegation 76
2.4.9 Data Abstraction and Encapsulation 76
2.5 Merits and Demerits of Object Oriented Programming Language 77
2.6 Applications of Object Oriented Programming 77
2.7 Differences Between Popular Programming Languages 78

3 Basics of Python Programming 83

3.1 Features of Python 83
3.2 History of Python 85
3.3 The Future of Python 87
3.4 Writing and Executing First Python Program 87
3.5 Literal Constants 88
3.5.1 Numbers 88
3.5.2 Strings 90
3.6 Variables and Identifiers 94
3.7 Data Types 94
3.7.1 Assigning or Initializing Values to Variables 94
3.7.2 Multiple Assignment 96
3.7.3 Multiple Statements on a Single Line 97
3.7.4 Boolean 97
3.8 Input Operation 97
3.9 Comments 98
3.10 Reserved Words 98
3.11 Indentation 98
3.12 Operators and Expressions 99
3.12.1 Arithmetic Operators 99
3.13.2 Comparison Operators 100
3.12.3 Assignment and In-place or Shortcut Operators 101
3.12.4 Unary Operators 102
3.12.5 Bitwise Operators 102
3.12.6 Shift Operators 103
3.12.7 Logical Operators 103
3.12.8 Membership Operators 104
3.12.9 Identity Operators 104
3.12.10 Operators Precedence and Associativity 105
3.13 Expressions in Python 106
3.14 Operations on Strings 106
3.14.1 Concatenation 107
3.14.2 Multiplication (or String Repetition) 107
3.14.3 Slice a String 108
3.15 Other Data Types 109
3.15.1 Tuples 109
3.15.2 Lists 109
3.15.3 Dictionary 110
3.16 Type Conversion 110
Annexure 2 — Installing Python 125
Annexure 3 — Comparison between Python 2.x and Python 3.x Versions 127
Annexure 4 — Testing and Debugging 130

4 Decision Control Statements 137

4.1 Introduction to Decision Control Statements 137
4.2 Selection/Conditional Branching Statements 138
4.2.1 `if` Statement 138
4.2.2 `if-else` Statement 139
4.2.3 Nested `if` Statements 142
4.2.4 `if-elif-else` Statement 142
4.3 Basic Loop Structures/ Iterative Statements 147
4.3.1 `while` loop 147
4.3.2 `for` Loop 155
4.3.3 Selecting an appropriate loop 156
4.4 Nested Loops 164
4.5 The `break` Statement 167
4.6 The `continue` Statement 168
4.7 The `pass` Statement 171
4.8 The `else` Statement used with Loops 172
Case Study 1 — Simple Calculator 180
Case Study 2 — Generating a Calendar 183

5 Functions and Modules 185

5.1 Introduction 185
5.1.1 Need for Functions 186
5.2 Function Definition 187
5.3 Function Call 189
5.3.1 Function Parameters 189
5.4 Variable Scope and Lifetime 191
5.4.1 Local and Global Variables 192
5.4.2 Using the Global Statement 193
5.4.3 Resolution of Names 195
5.5 The `return` statement 196
5.6 More on Defining Functions 198
5.6.1 Required Arguments 198
5.6.2 Keyword Arguments 198
5.6.3 Default Arguments 199

5.6.4 Variable-length Arguments 201

5.7 Lambda Functions or Anonymous Functions 201

5.8 Documentation Strings 205

5.9 Good Programming Practices 206

5.10 Recursive Functions 211

5.10.1 Greatest Common Divisor 213

5.10.2 Finding Exponents 214

5.10.3 The Fibonacci Series 214

5.10.4 Recursion vs Iteration 216

5.11 Modules 217

5.11.1 The from…import statement 218

5.11.2 Name of Module 220

5.11.3 Making your own Modules 220

5.11.4 The `dir()` function 221

5.11.5 The Python Module 223

5.11.6 Modules and Namespaces 223

5.12 Packages in Python 225

5.13 Standard Library modules 226

5.14 `Globals()`, `Locals()`, and `Reload()` 227

5.15 Function Redefinition 228

Annexure 5 — Functions as Objects 237

Case Study 3 — Tower of Hanoi 239

Case Study 4 — Shuffling a Deck of Cards 241

6 Python Strings Revisited 242

Introduction 242

6.1 Concatenating, Appending, and Multiplying Strings 243

6.2 Strings are Immutable 245

6.3 String Formatting Operator 246

6.4 Built-in String Methods and Functions 249

6.5 Slice Operation 254

6.5.1 Specifying Stride While Slicing Strings 256

6.6 `ord()` and `chr()` Functions 257

6.7 `in` and `not in` operators 257

6.8 Comparing Strings 258

6.9 Iterating String 259

6.10 The String Module 265

6.11 Regular Expressions 268

6.11.1 The `match()` Function 268

6.11.2 The `search()` Function 269

6.11.3 The `sub()` Function 270

6.11.4 The `findall()` and `finditer()` Functions 270

6.11.5 Flag Options 271

6.12 Metacharacters in Regular Expression 272

6.12.1 Character Classes 273

6.12.2 Groups 276

6.12.3 Application of Regular Expression to Extract Email 277

7 File Handling 289

7.1 Introduction 289

7.2 File Path 289

7.3 Types of Files 290

7.3.1 ASCII Text Files 291

7.3.2 Binary Files 291

7.4 Opening and Closing Files 292

7.4.1 The `open()` Function 292

7.4.2 The File Object Attributes 293

7.4.3 The `close()` Method 294

7.5 Reading and Writing Files 295

7.5.1 `write()` and `writelines()` Methods 295

7.5.2 `append()` Method 295

7.5.3 The `read()` and `readline()` Methods 296

7.5.4 Opening Files using `with` Keyword 298

7.5.5 Splitting Words 299

7.5.6 Some Other Useful File Methods 300

7.6 File Positions 300

7.7. Renaming and Deleting Files 303

7.8 Directory Methods 304

7.8.1 Methods from the os Module 307

Case Study 5 — Creating a Hash File (or a message digest of a file) 317

Case Study 6 — Mail Merge Program 319

Case Study 7 — Finding Resolution of an Image 321

8 Data Structures 322

8.1 Sequence 322

8.2 Lists 322

8.2.1 Access Values in Lists 323

8.2.2 Updating Values in Lists 323
8.2.3 Nested Lists 325
8.2.4 Cloning Lists 326
8.2.5 Basic List Operations 326
8.2.6 List Methods 327
8.2.7 Using Lists as Stack 330
8.2.8 Using Lists as Queues 331
8.2.9 List Comprehensions 332
8.2.10 Looping in Lists 334
✓ 8.3 Functional Programming 335
8.3.1 `filter()` Function 335
8.3.2 `map()` Function 336
8.3.3 `reduce()` Function 337
✓ 8.4 Tuple 346
8.4.1 Creating Tuple 346
8.4.2 Utility of Tuples 347
8.4.3 Accessing Values in a Tuple 347
8.3.4 Updating Tuple 348
8.4.5 Deleting Elements in Tuple 348
8.4.6 Basic Tuple Operations 349
8.4.7 Tuple Assignment 349
8.4.8 Tuples for Returning Multiple Values 350
8.3.9 Nested Tuples 351
8.4.10 Checking the Index: `index()` method 351
8.4.11 Counting the Elements: `count()` Method 352
8.4.12 List Comprehension and Tuples 352
8.4.13 Variable-length Argument Tuples 353
8.4.14 The `zip()` Function 354
8.4.16 Advantages of Tuple over List 356
8.5 Sets 359
8.5.1 Creating a Set 359
8.6 Dictionaries 366
8.6.1 Creating a Dictionary 367
8.6.2 Accessing Values 368
8.6.2 Adding and Modifying an Item in a Dictionary 369
8.6.3 Modifying an Entry 369
8.6.4 Deleting Items 370
8.6.5 Sorting Items in a Dictionary 373
8.6.6 Looping over a Dictionary 373
8.6.7 Nested Dictionaries 373
8.6.8 Built-in Dictionary Functions and Methods 374
8.6.9 Difference between a List and a Dictionary 376
8.6.10 String Formatting with Dictionaries 377
8.6.11 When to use which Data Structure? 377
8.6.12 List vs Tuple vs Dictionary vs Set 377
Annexure 6 — Iterator and Generator 392
9 Classes and Objects 400
9.1 Introduction 400
9.2 Classes and Objects 400
9.2.1 Defining Classes 400
9.2.2 Creating Objects 401
9.2.3 Data Abstraction and Hiding through Classes 401
9.3 Class Method and `self` Argument 402
9.4 The `__init__()` Method (The Class Constructor) 403
9.5 Class Variables and Object Variables 403
9.6 The `__del__()` Method 406
9.7 Other Special Methods 407
9.8 Public and Private Data Members 408
9.9 Private Methods 409
9.10 Calling a Class Method from Another Class Method 410
9.11 Built-in Functions to Check, Get, Set, and Delete Class Attributes 412
9.12 Built-in Class Attributes 413
9.13 Garbage Collection (Destroying Objects) 414
9.14 Class Methods 425
9.15 Static Methods 426
Annexure 7 — Getters, Setters, `@property`, and `@deleter` 432
10 Inheritance 436
10.1 Introduction 436
10.2 Inheriting Classes in Python 437
10.2.1 Polymorphism and Method Overriding 438

10.3 Types of Inheritance *441*
- 10.3.1 Multiple Inheritance *441*
- 10.3.2 Multi-level Inheritance *442*
- 10.3.3 Multi-path Inheritance *443*

10.4 Composition or Containership or Complex Objects *444*

10.5 Abstract Classes and Interfaces *446*

10.6 Metaclass *447*

11 Operator Overloading 460

11.1 Introduction *460*
- 11.1.1 Concept Of Operator Overloading *460*
- 11.1.2 Advantage of Operator Overloading *461*

11.2 Implementing Operator Overloading *461*

11.3 Reverse Adding *470*

11.4 Overriding `__getitem__()` and `__setitem__()` Methods *471*

11.5 Overriding the `in` Operator *472*

11.6 Overloading Miscellaneous Functions *473*

11.7 Overriding the `__call__()` Method *474*

12 Error and Exception Handling 480

12.1 Introduction to Errors and Exceptions *480*
- 12.1.1 Syntax Errors *480*
- 12.1.2 Logic Error *481*
- 12.1.3 Exceptions *481*

12.2 Handling Exceptions *482*

12.3 Multiple Except Blocks *483*

12.4 Multiple Exceptions in a Single Block *484*

12.5 Except Block Without Exception *484*

12.6 The `else` Clause *485*

12.7 Raising Exceptions *486*

12.8 Instantiating Exceptions *486*

12.9 Handling Exceptions in Invoked Functions *487*

12.10 Built-in and User-defined Exceptions *489*

12.11 The `finally` Block *491*

12.12 Pre-defined Clean–up Action *493*

12.13 Re-raising Exception *494*

12.14 Assertions in Python *494*

Case Study 8 — Compressing String and Files 505

Appendix A — Multi-threading 508

Appendix B — GUI Programming with `tkinter` *Package 514*

Appendix C — Simple Graphics Using Turtle 524

Appendix D — Plotting Graphs in Python 530

Appendix E — CGI/Web Programming Using Python 536

CHAPTER 1

Introduction to Computers and Problem Solving Strategies

KEY Concepts

• Characteristic of computers • History of computer • Types of computers • Applications of computers • Components of a computer • Basic architecture and organization of computer • CPU • Input and Output devices • Computer memory • Generations of programming languages • Computer hardware and software • Translators, linkers, loaders, assemblers • Stored program concept • System software • Application software • Software development process languages • Pseudocodes • Flowcharts • Algorithms • Types of errors • Testing and debugging

1.1 INTRODUCTION

We all have seen computers in our homes, schools, or colleges. In fact, in today's scenario we find computers in most aspects of daily life, and for some it is hard to even imagine a world without them. A computer is basically a machine that takes instructions and performs computations based on those instructions.

Nowadays computers come in different sizes. Their size may vary from very small to very large. In the past, computers were extremely large in size and required an entire room for installation. These computers consumed enormous amounts of power and were too expensive to be used for commercial applications. Therefore, they were used only for limited tasks, such as computing trajectories for astronomical or military applications.

However, with technological advancements, the size of computers became smaller and their energy requirements lowered immensely. This opened the way for adoption of computers for commercial purposes.

These days, computers have become so prevalent in the market that all interactive devices such as cellular phones, global positioning system (GPS) units, portable organizers, automated teller machines (ATMs), and gas pumps work with computers.

1.2 WHAT IS A COMPUTER?

A computer is an electronic machine that takes instructions and performs computations based on those instructions. Before going into details, let us learn some key terms that are frequently used in computers.

Data Data is a collection of raw facts or figures.

Information Information comprises processed data to provide answers to *'who', 'what', 'where',* and *'when'* type of questions.

Knowledge Knowledge is the application of data and information to answer *'how'* part of the question (refer Figure 1.1).

Instructions Commands given to the computer that tells what it has to do are instructions.

Programs A set of instructions in computer language is called a program.

Software A set of programs is called software.

Hardware A computer and all its physical parts are known as hardware.

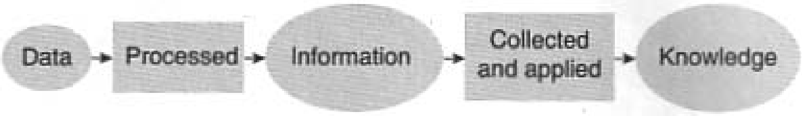

Figure 1.1 Data, information, and knowledge

1.3 HISTORY OF COMPUTERS

History of computers can be understood by looking into five generations. With each new generation of computers, there had been advancement in computer technology. The circuitry became smaller with enhanced speed, less consumption of power, and efficient memory.

Therefore, each generation of computer is characterized by a major technological development that has drastically changed the way in which computers operate.

First Generation (1942–1955)

Hardware Technology First generation computers were manufactured using thousands of vacuum tubes. Vacuum tube (as shown in Figure 1.2) is a device made of fragile glass.

Software Technology Programming was done in machine language or assembly language.

Used for Scientific applications

Examples ENIAC, EDVAC, EDSAC, UNIVAC I, IBM 701

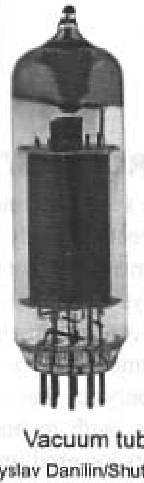

Figure 1.2 Vacuum tube
Source: Vladyslav Danilin/Shutterstock

Highlights

- They were the fastest calculating device of those times.
- Computers were too bulky and required a complete room for storage.
- Highly unreliable as vacuum tubes emitted a large amount of heat and burnt frequently.
- Required air-conditioned room for installation.
- Costly.
- Difficult to use.
- Required constant maintenance because vacuum tubes used filaments that had limited lifetime. Therefore, these computers were prone to frequent hardware failures.

Second Generation (1955–1964)

Hardware Technology Second generation computers were manufactured using transistors (as shown in Figure 1.3). Transistors were reliable, powerful, cheaper, smaller, and cooler than vacuum tubes.

Software Technology Programming was done in high-level programming language.

Used for Scientific and commercial applications

Examples Honeywell 400, IBM 7030, CDC 1604,

UNIVAC LARC

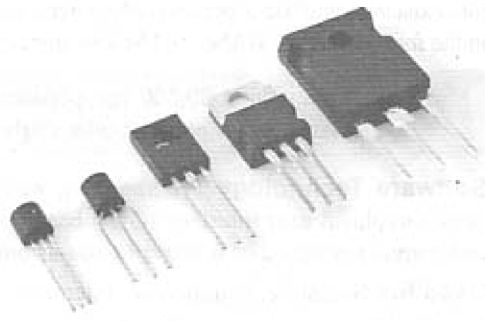

Figure 1.3 Transistors

Source: yurazaga/Shutterstock

Highlights

- Faster, smaller, cheaper, reliable, and easier to use than the first generation computers.
- Consumed 1/10th the power consumed by first generation computers.
- Bulky in size and required a complete room for its installation.
- Dissipated less heat than first generation computers but still required air-conditioned room.
- Costly.
- Difficult to use.

Note Initially, ICs contained 10–20 components. This technology was called Small Scale Integration (SSI). Later it was enhanced to contain about 100 components. This was called MSI (Medium Scale Integration).

Third Generation (1964–1975)

Hardware Technology Third generation computers were manufactured using integrated chips (ICs) as shown in Figure 1.4. ICs consist of several components such as transistors, capacitors, and resistors on a single chip to avoid wired interconnection between components. These computers used SSI and MSI technology. Minicomputers came into existence.

Figure 1.4 Integrated chip

Source: cooldesign/FreeDigitalPhotos.net

Software Technology Programming was done in high-level programming language such as FORTRAN, COBOL, Pascal, and BASIC.

Used for Scientific, commercial, and interactive online applications.

Examples IBM 360/370, PDP-8, PADP-11, CDC6600

Highlights

- Faster, smaller, cheaper, reliable, and easier to use than the second generation computers.
- They consumed less power than second generation computers.
- Bulky in size and required a complete room for its installation.
- Dissipated less heat than second generation computers but still required air-conditioned room.
- Costly.
- Easier to use and upgrade.

Figure 1.5 VLSI

Fourth Generation (1975–1989)

Hardware Technology Fourth generation computers were manufactured using ICs with LSI (Large Scale Integrated) and later with VLSI (Very Large Scale Integrated) technology as shown in Figure 1.5. Microcomputers came into existence, and use of personal computers became widespread during this period. High speed computer networks in the form of LANs, WANs, and MANs started growing. Besides mainframes, supercomputers were also used.

Note LSI contained 30,000 components on a single chip and VLSI technology had about 1 million electronic components on a single chip.

Software Technology Programming was done in high-level programming language such as C++ and Java. Graphical user interface (GUI) based operating system (like Windows) was introduced. It had icons and menus among other features to allow computers to be used as a general purpose machine by all users.

Used for Scientific, commercial, interactive online, and network applications.

Examples IBM PC, Apple II, TRS-80, VAX 9000, CRAY-1, CRAY-2, CRAY-X/MP

Highlights Faster, smaller, cheaper, powerful, reliable, and easier to use than the previous generation computers.

Fifth Generation (1989–Present)

Hardware Technology Fifth generation computers were manufactured using ICs with ULSI (Ultra Large Scale Integrated) technology as shown in Figure 1.6. Use of Internet became widespread. Very powerful mainframes, desktops, portable laptops, and smartphones are being used commonly. Super computers use parallel processing techniques.

Note ULSI contained about 10 million electronic components on a single chip.

Software Technology Programming was done in high-level programming language such as Java, Python, and C#.

Used for Scientific, commercial, interactive online, multimedia (graphics, audio, video), and network applications.

Examples IBM notebooks, Pentium PCs, SUM workstations, IBM SP/2, Param supercomputer.

Figure 1.6 ULSI

Highlights

- Faster, smaller, cheaper, powerful, reliable, and easier to use than the previous generation computers.
- Speed of microprocessors and the size of memory are growing rapidly.
- High-end features available on mainframe computers in the fourth generation are now available on the microprocessors.
- Consume less power than computers of prior generations.
- Air-conditioned rooms required for mainframes and supercomputers but not for microprocessors.

1.4 CHARACTERISTICS OF COMPUTERS

The important characteristics of a computer (as shown in Figure 1.7) are as follows.

Speed Computers can perform millions of operations in a single second. This means that a computer can process the data in blink of an eye which otherwise may take multiple days to complete. The speed of the computer is usually given in *nano second* and *pico second,* where

$$1 \text{ nano second} = 1 \times 10^{-9} \text{ second and } 1 \text{ pico second} = 1 \times 10^{-12} \text{ second}$$

Accuracy Computers are a reliable electronic device. It never makes mistakes. It always gives accurate results provided that correct data and set of instructions are input to it. So in the advent of an error, only the user who has fed the incorrect data/program is responsible. If the input data is wrong, then the output will also be erroneous. In computer terminology, it is known as garbage-in garbage-out (GIGO).

Automatic Besides being very fast and accurate, computers are automatic devices that can perform without any user intervention. The user just needs to assign the task to the computer after which the computer automatically controls different devices attached to it and executes the program instructions one by one.

Diligence Computers can never get tired as humans do. It can continually work for hours without creating any error. If a large number of executions have to be made then each and every execution will require the same amount of time and accuracy.

Versatile Versatile means flexible. Today, computers are being used in our daily lives in different fields. For example, they are used as personal computers (PCs) for home use, for business-oriented tasks, weather forecasting, space explorations, teaching, railways, banking, medicine, etc. On the PC that you use at home, you may play a game, compose and send e-mails, listen to music, etc. Therefore, computers are versatile devices as they can perform multiple tasks of different nature at the same time.

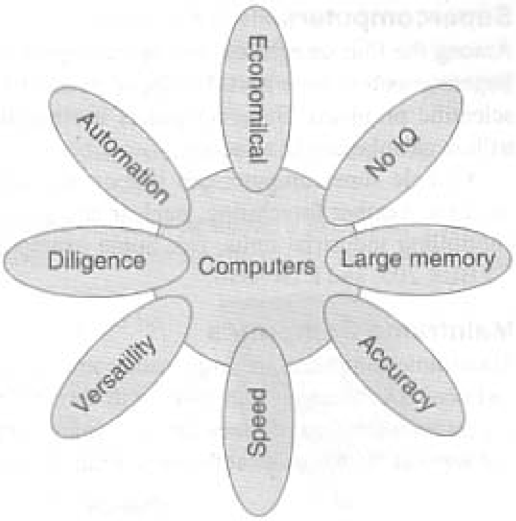

Figure 1.7 Characteristics of a computer

Memory Similar to humans, computers also have memory. Human beings cannot store everything in their memory and need secondary media, such as a notebook to record certain important things. Similarly, computers have internal memory (storage space) as well as external or secondary memory. While the internal memory of computers is very expensive and limited in size, the secondary storage is cheaper and bigger in size.

The computer stores a large amount of data and programs in the secondary storage space. The stored data and programs can be used whenever required. Secondary memory devices include CD, DVD, hard disk, pen drives, etc.

Note When data and programs have to be used they are copied from the secondary memory into the internal memory (often known as RAM).

No IQ Although the trend today is to make computers intelligent by inducing artificial intelligence (AI) in them, they do not have any decision-making abilities of their own, that is, their IQ level is zero. They need guidance to perform various tasks.

Economical Today, computers are considered as short-term investment for achieving long-term gain. Using computers also reduces manpower requirements and leads to an elegant and efficient way for doing tasks. Hence, computers save time, energy, and money. When compared to other systems, computers can do more work in lesser time. For example, using the conventional postal system to send an important document takes at least 2–3 days, whereas the same information when sent using the Internet (e-mail) will be delivered instantaneously.

1.5 CLASSIFICATION OF COMPUTERS

Computers can be broadly classified into four categories based on their speed, amount of data that they can process, and price (refer to Figure 1.8). These categories are as follows:

- Supercomputers
- Mainframe computers
- Minicomputers
- Microcomputers

Supercomputers

Among the four categories, the supercomputer is the fastest, most powerful, and most expensive computer. Supercomputers were first developed in the 1980s to process large amounts of data and to solve complex scientific problems. Supercomputers use parallel processing technology and can perform more than one trillion calculations in a second.

A single supercomputer can support thousands of users at the same time. Such computers are mainly used for weather forecasting, nuclear energy research, aircraft design, automotive design, online banking, controlling industrial units, etc. Some examples of supercomputers are CRAY-1, CRAY-2, Control Data CYBER 205, and ETA A-10.

Mainframe Computers

Mainframe computers are large-scale computers (but smaller than supercomputers). These are very expensive and need a very large clean room with air conditioning, thereby making them very costly to deploy. As with supercomputers, mainframes can also support multiple processors. For example, the IBM S/390 mainframe can support 50,000 users at the same time. Users can access mainframes by either using terminals or via PCs.

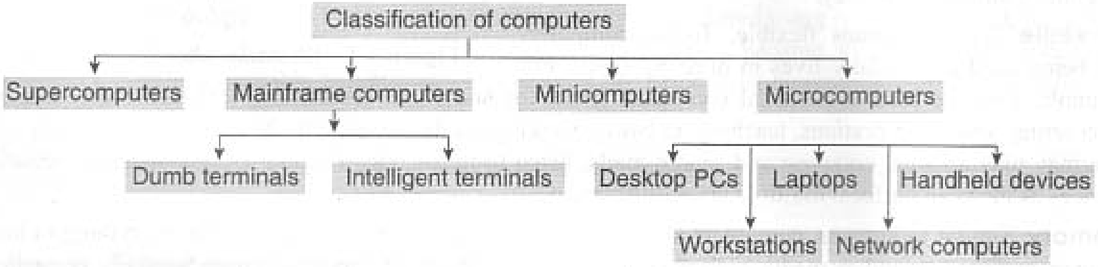

Figure 1.8 Classification of computers

Mainframe computers are typically used as servers on the World Wide Web. They are also used in organizations such as banks, airline companies, and universities, where a large number of users frequently access the data stored in their databases. IBM is the major manufacturer of mainframe computers. Some examples of mainframe computers include IBM S/390, Control Data CYBER 176, and Amdahl 580.

Minicomputers

As the name suggests, minicomputers are smaller, cheaper, and slower than mainframes. They are called minicomputers because they were the smallest computer of their times. Also known as *midrange computers*, the capabilities of minicomputers fall between mainframe and personal computers.

Minicomputers are widely used in business, education, hospitals, government organizations, etc. While some minicomputers can be used only by a single user, others are specifically designed to handle multiple users simultaneously. Usually, single-user minicomputers are used for performing complex design tasks.

As with mainframes, minicomputers can also be used as servers in a networked environment, and hundreds of PCs can be connected to it.

The first minicomputer was introduced by Digital Equipment Corporation (DEC) in the mid-1960s. Other manufacturers of minicomputers include IBM Corporation (AS/400 computers), Data General Corporation, and Prime Computer.

Microcomputers

Microcomputers, commonly known as PCs, are very small and cheap. The first microcomputer was designed by IBM in 1981 and was named IBM-PC. Later on, many computer hardware companies copied this design and termed their microcomputers *PC-compatible*, which refers to any PC that is based on the original IBM PC design.

Another type of popular PC is designed by Apple. PCs designed by IBM and other PC-compatible computers have a different architecture from that of Apple computers. Moreover, PCs and PC-compatible computers commonly use the Windows operating system, while Apple computers use the Macintosh operating system (MacOS). PCs can be classified into the following categories:

Desktop PCs A desktop PC is the most popular model of PCs. The system unit of the desktop PC can be placed flat on a desk or table. It is widely used in homes and offices.

Laptops Laptops (Figure 1.9) are small microcomputers that can easily fit inside a briefcase. They are very handy and can easily be carried from one place to another. They may also be placed on the user's lap (thus the name). Hence, laptops are very useful, especially when going on long journeys. Laptops operate on a battery and do not always have to be plugged in like desktop computers.

The memory and storage capacity of a laptop is almost equivalent to that of a desktop computer. As with desktop computers, laptops also have hard disk drives, USB drives, etc. For input, laptops have a built-in keyboard and a trackball/touchpad, which is used as a pointing device (as a mouse is used for a desktop PC).

Today, laptops have the same features and processing speed as the most powerful PCs. However, a drawback is that laptops are generally more expensive than desktop computers. These computers are very popular among business travellers.

Figure 1.9 Laptop

Source: You can more/Shutterstock

Workstations Workstations are single-user computers that have the same features as PCs, but their processing speed matches that of a minicomputer or mainframe computer. Workstation computers have advanced processors, and more RAM and storage capacity than PCs. Therefore, they are more expensive and powerful than a normal desktop computer.

Although workstation computers are widely used as powerful single-user computers by scientists, engineers, architects, and graphic designers, they can also be used as servers in a networked environment.

Network Computers Network computers have less processing power, memory, and storage than a desktop computer. These are specially designed to be used as terminals in a networked environment. For example, some network computers are specifically designed to access data stored on a network (including the Internet and intranet).

Some network computers do not have any storage space and merely rely on the network's server for data storage and processing tasks. The concept of network computers had become popular in the mid-1990s when several variations of computers such as Windows terminals, NetPCs, and diskless workstations were widely used.

Network computers that are specifically designed to access only the Internet or intranet are often known as Internet PCs or Internet boxes. Some network computers used in homes do not even have a monitor. Such computers may be connected to a television, which serves as the output device. The most common example of a home-based network computer is Web TV, which enables the user to connect a television to the Internet. The other reason for the popularity of network computers is that they are cheaper to purchase and maintain than PCs.

Handheld Computers The mid-1990s witnessed a range of small personal computing devices that are commonly known as handheld computers, or mobile computers. These computers are called handheld computers because they can fit in one hand, while users can use the other hand to operate them. Handheld computers are very small in size, and hence they have small-sized screens and keyboards. These computers are preferred by business travellers and mobile employees whose jobs require them to move from place to place.

Some examples of handheld computers are as follows:

- Smartphones
- Tablet PCs
- Phablets

Smartphones These days, cellular phones are web-enabled telephones that have features of both analog and digital devices. Such phones are also known as smartphones because, in addition to basic phone capabilities, they also facilitate the users to access the Internet and send e-mails and faxes.

Tablet PCs A tablet PC (refer Figure 1.10) is a computing device that is smaller than a laptop, but bigger than a smartphone. Features such as user-friendly interface, portability, and touch screen have made them very popular in the last few years. These days, a wide range of high-performance tablets are available in the market. While all of them look similar from outside, they may differ in features such as operating system, speed of data connectivity, camera specifications, size of the screen, processing power, battery life, and storage capability.

Some operating systems that are used in tablets are Android Jellybean (an open-source operating system built by Google), Windows 8, and iOS (developed by Apple).

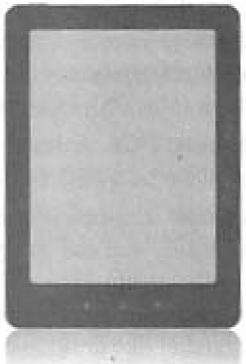

Figure 1.10 Tablet

Source: bloomua/Shutterstock/OUP Picture Bank

While users can easily type directly on the surface of a tablet, some users prefer a wireless or bluetooth-connected keyboard. These days, tablets also offer an optional docking station with keyboards that transforms the tablet into a full-featured netbook.

Uses The following are the uses of tablet PCs:

- Viewing presentations
- Video conferencing
- Reading e-books, e-newspaper
- Watching movies
- Playing games
- Sharing pictures, video, songs, documents, etc.
- Browsing the Internet
- Keeping in touch with friends and family on popular social networks, sending emails
- Business people use them to perform tasks such as editing documents, exchanging documents, taking notes, and giving presentations
- Tablets are best used in crowded places such as airports and coffee shops, where size and portability become more important.

Note Tablets may replace laptops if users don't have to perform heavy processing tasks and do not require a CD or DVD player.

Phablet (Phone + Tablet) Phablet is a class of mobile device that combines the functions of a smartphone and tablet. Usually, mobile devices with screen size 4–5 inch are termed as smartphones and those with size ranging from 7–10 inch are known as tablets. A phablet fills the void between the two types of devices. Therefore, phablet is a half-smartphone and half-tablet mobile device (refer Figure 1.11). A phablet can be easily held and used in one hand. These days, phablets support 3G as well as 4G networks for cellular calls and are Wi-Fi-enabled.

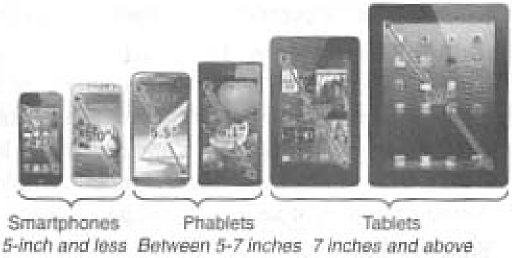

Figure 1.11 Comparison between Smartphone, Phablet, and Tablet

The trend of phablet started with Samsung's Galaxy Note in 2011 and its popularity grew dramatically in 2012 due to the falling costs and increasing power efficiency of smartphone display. Following the competition, other smartphone manufacturers, including Lenovo, LG, HTC, Huawei, Micromax, and Sony came up with their models of phablets.

Raspberry Pi

Raspberry Pi is a credit card sized computer which was originally designed for education. It is a low-cost device that has been specifically created to improve programming skills and hardware understanding at the pre-university level. Raspberry Pi was originally intended to be a microcomputer to teach children coding. But later on its scope was expanded and it has now become a very popular device.

Although the Raspberry Pi is slower than a modern laptop or desktop, it is a complete Linux computer that provides all the expected abilities at a low-power consumption level. Raspberry Pi is being widely used around the world either as a desktop computer or as a device to build smart devices. For example, some common use of Raspberry Pi include:

- Teach coding
- Used as a desktop
- Create a retro gaming console
- Make a world clock or an FM radio with the Pi Zero
- Make a media center with Rasplex or an always-on downloading machine
- Build a motion capture security camera

1.6 BASIC APPLICATIONS OF COMPUTERS

When the first computers were developed, they were used only in the fields of mathematics and science. In fact, the first effective utilization of computers was for decoding messages in military applications. Later on, computers were used in real-time control systems, such as for landing on the moon. However, with the advancement of technology, the cost of a computer and its maintenance declined. This opened the way for computers extensively being used in business and commercial sector for information processing. Today, computers are widely used in different fields as discussed below.

Communication Internet which connects computers all over the world. Internet gives you access to enormous amount of information, much more than you could have in a library. Then using electronic mail you can communicate in seconds with a person who is thousands of miles away. The chat software enables you to chat with another person in real-time (irrespective of the physical location of that person). Then, video conferencing tools are becoming popular for conducting meetings with people who are unable to be present at a particular place.

Desktop Publishing Desktop publishing software enables you to create page layouts for entire books.

Government Computers are used to keep records on legislative actions, Internal Revenue Service records, etc.

Traffic Control It is used by governments for city planning and traffic control.

Legal System Computers are being used by lawyers to shorten the time required to conduct legal precedent and case research. Lawyers use computers to look through millions of individual cases and find whether similar or parallel cases were approved, denied, criticized, or overruled. This enables the lawyers to formulate strategies based on past case decisions. Moreover, computers are also used to keep track of appointments and prepare legal documents and briefs in time for filling cases.

Retail Business Computers are used in retail shops to enter the order, calculate the cost, and print a receipt. They are also used to keep an inventory of the products available and a complete description about them.

Sports In sports, computers are used to compile statistics, identify weak players and strong players by analyzing statistics, sell tickets, create training programs and diets for athletes, and suggest game plan strategies based on the competitor's past performance. Computers are also used to generate most of the graphic art displays flashed on scoreboards.

Computers are used in the control room to display action replays and insert commercial breaks on schedule. Moreover, sports shoes manufacturing companies, like NIKE, use computers for designing footwears. They calculate stress points and then create the style and shape that offer maximum support for the foot.

Music Computers are used to generate a variety of sounds. Moreover, the background music in movies, television shows, and commercials are all generated electronically using computers.

Movies Computers are used to create sets, special effects, animations, cartoons, imaginary characters, videos, and commercials.

Travel and Tourism Computers are used to prepare ticket, monitor the train's or airplane's route, or guide the plane to a safe landing. They are also used to know about hotels in an area, reserve room, or rent a car.

Business and Industry In business and industry, computers are used mainly for entering and analysing data, pay roll processing, personnel record keeping, inventory management, etc.

Hospitals Hospitals use computers to record every information about a patient from the time of his admission till his exit. For example the date, time, reason of admit, the doctor being consulted, all prescribed medications, doctor visits, other hospital services, bill, etc. are all stored in computers. Moreover, computer-controlled devices are widely used to monitor pulse rate, blood pressure, and other vital signs of the patient and in an emergency situation an alarm is used to notify the nurses and other attendants.

Moreover, computers are used as an aid to physically handicapped people. For example, computers are used to develop more effective artificial limbs for amputees.

Simulation Computers enable the engineers to design aircraft models and simulate the effects that winds and other environmental forces might have on those designs. Even the astronauts at NASA are trained using computer-simulated problems that could be encountered during launch, in space, or upon return to Earth.

Geology Civil engineers use computers to evaluate the effects of an earthquake on the structure of buildings based on age, proximity to the fault, soil type, size, shape, and construction material.

Astronomy Spacecrafts are usually monitored using computers which not only keep a continuous record of the voyage and the records of the speed, direction, fuel, temperature, and such performance but also suggests a corrective action if the vehicle makes any mistake. The remote stations on the earth compares all these quantities with the desired values and in case these values need to be modified to enhance the performance of the space craft, signals are immediately sent which set in motion the mechanics to rectify the situation. With the help of computers, these are done within a fraction of seconds.

Weather Forecasting When computers are fed with mathematical equations along with data about air pressure, temperature, humidity, and other values, the solution of these equations gives an accurate prediction of weather in a particular area. For example, a Crax XMP Supercomputer installed at Mausam Bhavan in New Delhi is used to predict weather and climatic changes in the Indian subcontinent.

Education A computer is a powerful teaching aid and acts as another teacher in the classroom. Teachers use computers to develop instructional material. They may use pictures, graphs, and graphical presentations to easily illustrate an otherwise difficult concept. Moreover, teachers at all levels can use computers to administer assignments and keep track of grades of the students. Besides teachers, students also prefer to learn from an E-learning software rather than learning from a book. Students can also give online exams and get instant results.

Online Banking The world today is moving towards a cashless society, where you need not have money in your pocket to purchase anything. You can just have your credit card or debit card with you. The ATM machines (Automated Teller Machine) provides a 24 × 7 service and allows you to draw cash, check the balance in your account, and order a product.

Industry and Engineering Computers are found in all kinds of industries like thermal power plant, oil refineries, chemical industries, etc. for process control, computer aided designing, and computer aided manufacturing.

Computerized process control (with or without human intervention) is used to enhance efficiency in applications such as production of various chemical products, oil refining, paper manufacture, rolling and cutting steel to customer requirements, etc.

In Computer Aided Design (CAD) the computers are used for automating the design and drafting process. It helps an engineer to design a part, analyse its characteristics, and then subject it to simulated stresses. In case a part fails the stress test, its specifications can be modified on the computer and retested. The final design specifications are released for production only when the engineer is satisfied that the part meets strength and other quality considerations.

Computer-aided manufacturing (CAM) phase comes up where CAD leaves off. In this phase, the metal or other materials are manufactured while complying with their specification. For this computer-controlled manufacturing tools are used to produce high-quality products.

Robots Robots are computer-controlled machines mainly used in manufacturing process in extreme conditions where humans cannot work. For example, in high temperature, high pressure conditions, or in processes that demand very high level of accuracy.

Decision Support Systems Computers help managers to analyse their organization's data to understand the present scenario of their business, view the trends in the market, and predict the future of their products. Managers also use decision support systems to analyse market research data, to size up the competition, and to plan effective strategies for penetrating their markets.

Expert System Expert systems are used to automate the decision-making process in a specific area like analysing the credit histories for loan approval and diagnosing a patient's condition for prescribing an appropriate treatment. Expert systems analyse the available data in depth to recommend a course of action. A medical expert system can provide the most likely diagnosis of a patient's condition.

Others Adding more to it, in today's scenario computers are used to find jobs on the Internet, find a suitable match for a boy or girl, read news and articles online, find one's batchmates, send and receive greetings pertaining to different occasions, etc.

1.7 STORED PROGRAM CONCEPT

All digital computers are based on the principle of stored program concept, which was introduced by Sir John von Neumann in the late 1940s. The following are the key characteristic features of this concept:

- Before any data is processed, instructions are read into memory.
- Instructions are stored in the computer's memory for execution.
- Instructions are stored in binary form (using binary numbers—only 0s and 1s).
- Processing starts with the first instruction in the program, which is copied into a control unit circuit. The control unit executes the instructions.
- Instructions written by the users are performed sequentially until there is a break in the current flow.
- Input/output and processing operations are performed simultaneously. While data is being read/written, the central processing unit (CPU) executes another program in the memory that is ready for execution.

Note A stored program architecture is a fundamental computer architecture wherein the computer executes the instructions that are stored in its memory.

A stored program architecture is a fundamental computer architecture wherein the computer executes the instructions that are stored in its memory. John W. Mauchly, an American physicist, and J. Presper Eckert, an American engineer, further contributed to the stored program concept to make digital computers much more flexible and powerful. As a result, engineers in England built the first stored-program computer, Manchester Mark I, in the year 1949. They were shortly followed by the Americans who designed EDVAC in the very same year.

Today, a CPU chip can handle billions of instructions per second. It executes instructions provided both the data and instructions are valid. In case either one of them or both are not valid, the computer stops the processing of instructions.

1.7.1 Types of Stored Program Computers

A computer with a Von Neumann architecture stores data and instructions in the same memory (refer Figure 1.12(a)). There is a serial machine in which data and instructions are selected one at a time. Data and instructions are transferred to and from memory through a shared data bus. Since there is a single bus to carry data and instructions, process execution becomes slower.

Later Harvard University proposed a stored program concept in which there was a separate memory to store data and instructions (refer Figure 1.12(b)). Instructions are selected serially from the instruction memory and executed in the processor. When an instruction needs data, it is selected from the data memory. Since there are separate memories, execution becomes faster.

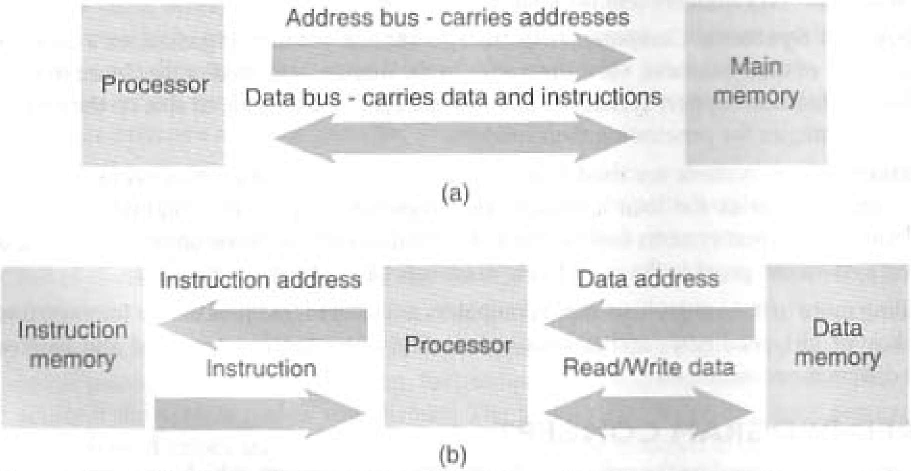

Figure 1.12 (a) Von Neumann architecture—Shared memory for instructions and data (b) Harvard architecture—Separate memories for instructions and data

1.8 COMPONENTS AND FUNCTIONS OF A COMPUTER SYSTEM

A computer is an electronic device which basically performs five major operations, which are as follows:

1. accepting data or instructions (input)
2. storing data
3. processing data
4. displaying results (output) and
5. controlling and coordinating all operations inside a computer

In this section, we will discuss all these functions and see how one component of a computer interacts with another unit to perform these operations using the block diagram of a computer as shown in Figure 1.13.

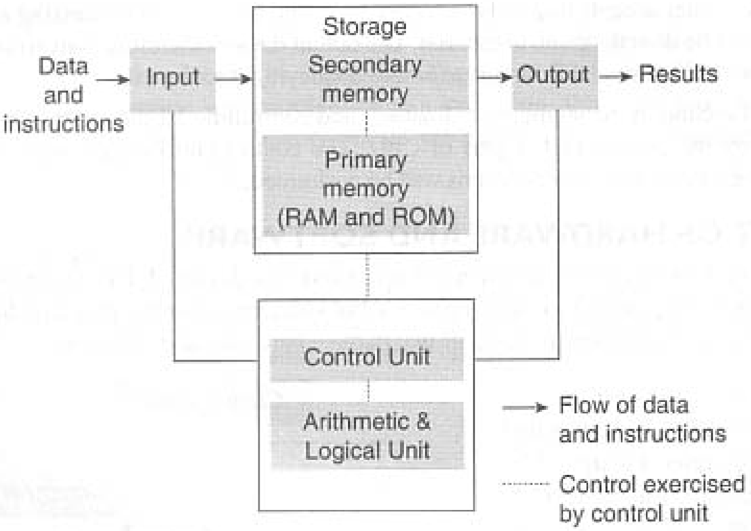
Data and instructions
Input
Storage
Secondary memory
Primary memory (RAM and ROM)
Output
Results
Control Unit
Arithmetic & Logical Unit
Flow of data and instructions
Control exercised by control unit

Figure 1.13 Block diagram of a computer

Input This is the process of entering data and instructions (also known as programs) into the computer system. The data and instructions can be entered into the computer system by using different input devices such as keyboard, mouse, scanner, trackball, etc.

> **Note** Computers understand binary language which consists of only two symbols (0s and 1s).Therefore, it is the responsibility of the input devices to convert the input data into binary codes.

Storage Storage is the process of saving data and instructions permanently in the computer so that it can be used for processing. The computer storage space stores not only the data and programs but also the intermediate results and the final results of processing. A computer has two types of storage areas:

Primary Storage Primary storage also known as the main memory is that storage area which is directly accessible by the CPU at a very fast speed. It is used to store the data and program, the intermediate results of processing, and the recently generated results. The primary storage is very expensive and therefore limited in capacity. Another drawback of main memory is that it is volatile in nature, that is, as soon as the computer is switched off, the information stored in it gets erased. Hence, it cannot be used as a permanent storage of useful data and programs for future use. For example, RAM (Random Access Memory).

Secondary Storage Also known as the secondary memory or auxiliary memory is just the opposite of primary memory. It basically overcomes all the drawbacks of the primary storage. It is cheaper, non-volatile, and used to permanently store data and programs of those jobs which are not being currently executed by the CPU. Secondary memory supplements the limited storage capacity of the primary memory. For example, using a magnetic disk you can store your data in C drive, D drive, etc. for future use.

Processing The process of performing operations on the data as per the instructions specified by the user (program) is called processing. Data processing is an activity that involves handling or manipulating data in some way to assign meaning to it. The main aim of processing is to transform data into information. Data and instructions are taken from the primary memory and are transferred to the Arithmetic and Logical Unit (ALU), a part of CPU, which performs all sorts of calculations. When the processing completes, the final result is transferred to the main memory.

Output Output is the reverse of input. It is the process of giving the result of data processing to the outside world (external to the computer system). The results are given through output devices like monitor, printer, etc. Now that the computer accepts data only in binary form and the result of processing is also in the binary form, the result cannot be directly given to the user. The output devices therefore convert the results available in binary codes into a human-readable language before displaying it to the user.

Controlling The function of managing, coordinating, and controlling all the components of the computer system is handled by the control unit, a part of CPU. The control unit decides the manner in which the instructions will be executed and the operations will be performed.

1.9 CONCEPT OF HARDWARE AND SOFTWARE

You have a TV at home. When you purchase a TV, it is a box like device. A TV can be used only when it is able to display different programs. You can touch a TV but you cannot touch a program. Same is the concept in a computer. A computer system is made up of two parts— hardware and software.

1.9.1 Hardware

All the physical parts that can be touched are called hardware (refer Figure 1.14). For example, all input and output devices, and memory devices form the hardware part of the computer.

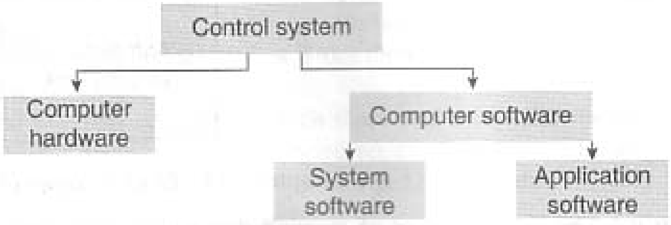

Figure 1.14 Parts of a computer system

If we think of computer as a living being, then the hardware would be the body that does things like seeing with eyes, lifting objects, and filling the lungs with air; the software would be the intelligence that helps in interpreting the images that come through the eyes, instructing the arms how to lift objects, and forcing the body to fill the lungs with air.

Since the computer hardware is a part of a machine, it can only understand two basic concepts: 'on' and 'off'. The 'on' and 'off' concept is called *binary*. Computer software was developed to tell the computer hardware what to do.

1.9.2 Software

The computer hardware cannot think and make decisions on its own. So, it cannot be used to analyse a given set of data and find a solution on its own. The hardware needs a software (a set of programs) to instruct what has to be done. A program is a set of instructions that is arranged in a sequence to guide a computer to find a solution for the given problem. The process of writing a program is called *programming*.

Let us now discuss the CPU and the other hardware components of a computer system in the following sections.

1.10 CENTRAL PROCESSING UNIT (CPU): BASIC ARCHITECTURE

Central Processing Unit can be called the brain of the computer system because the entire processing of data and execution of instructions is done here. It is made up of one or more than one microprocessors which consist of two main parts—Arithmetic and Logical Unit (ALU) and Control Unit (CU). It also contains registers and a Bus Interface Unit (BIU) of shown in Figure 1.15.

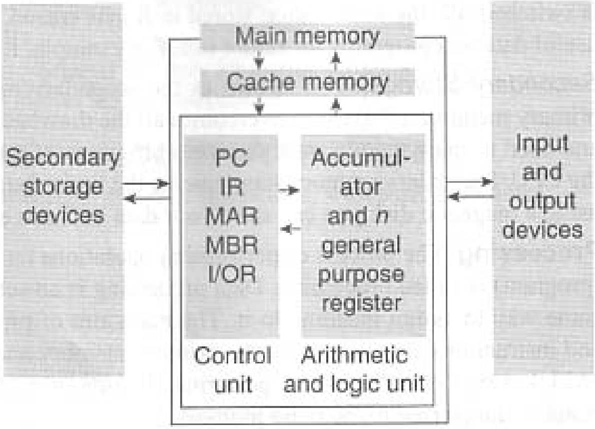
Main memory
Cache memory
Secondary storage devices
PC
IR
MAR
MBR
I/OR
Accumul-ator and n general purpose register
Input and output devices
Control unit
Arithmetic and logic unit

Figure 1.15 Basic computer organization

Arithmetic and Logical Unit

The ALU performs all kinds of calculations, such as arithmetic (add, subtract, multiply, divide, etc.), comparison (less than, greater than, or equal to), and other operations. The intermediate results of processing may be stored in the main memory, as they might be required again. When the processing completes, the final result is then transferred to the main memory. Hence, the data may move from main memory to the ALU multiple times before the processing is over.

Control Unit

The main function of the CU is to direct and coordinate the computer operations. It interprets the instructions (program) and initiates action to execute them. The CU controls the flow of data through the computer system and directs the ALU, input/output (I/O) devices, and other units. It is, therefore, called the central nervous system of the computer system. In addition, the CU is responsible for fetching, decoding, executing instructions, and storing results.

Registers

A processor register is a computer memory that provides quick access to the data currently being used for processing. The ALU stores all temporary results and the final result in the processor registers. As mentioned earlier, registers are at the top of memory hierarchy and are always preferred to speed up program execution.

Registers are also used to store the instructions of the program currently being executed. There are different types of registers, each with a specific storage function.

Accumulator and general-purpose registers These are frequently used to store the data brought from the main memory and the intermediate results during program execution. The number of general-purpose registers present varies from processor to processor. When program execution is complete, the result of processing is transferred from the accumulator to the memory through the memory buffer register (MBR).

Special-purpose registers These include the following:

- The memory address register (MAR) stores the address of the data or instruction to be fetched from the main memory. The value stored in the MAR is copied from the program counter.
- The MBR stores the data or instruction fetched from the main memory (Figure 1.16). If an instruction is fetched from the memory, then the contents of the MBR are copied into the instruction register (IR). If a data is fetched from the memory, the contents are either transferred to the accumulator or to the I/O register. The MBR is also used while writing contents in the main memory. In this case, the processor first transfers the contents to the MBR, which then writes them into the memory.

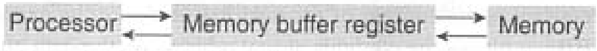

Figure 1.16 Data to and from memory comes from and to processor through the MBR

- The IR stores the instructions currently being executed. In general, an instruction consists of two parts—operation and address of the data on which the operation has to be performed. When the IR is loaded with an instruction, the address of the data is transferred to the MAR and the operation part is given to the CU, which interprets it and executes it.
- The I/O register is used to transfer data or instructions to or from an I/O device. An input device transfers data to the I/O register for processing. Correspondingly, any data to be sent to the output device is written in this register.
- The program counter stores the address of the next instruction to be executed.

The size of a register is usually specified by the number of bits it can store. For example, a register can be of 8 bits, 16 bits, 32 bits, or 64 bits. Higher the register size, more the data that can be stored in it.

Instruction cycle To execute an instruction, a processor normally follows a set of basic operations that are together known as an instruction cycle (Figure 1.17). The operations performed in an instruction cycle involve the following:

Fetch Retrieving an instruction or a data from memory.

Decode Interpreting the instruction.

Execute Running the corresponding commands to process the data.

Store Writing the results of processing into memory.

This instruction cycle is repeated continuously until the power is turned off.

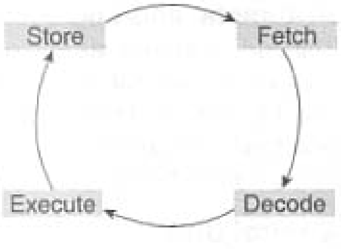

Figure 1.17 Instruction cycle

Bus Interface Unit

The BIU provides functions for transferring data between the execution unit of the CPU and other components of the computer system that lie outside the CPU. Every computer system has three different types of busses to carry information from one part to the other. These are the data bus, control bus, and address bus (Figure 1.18).

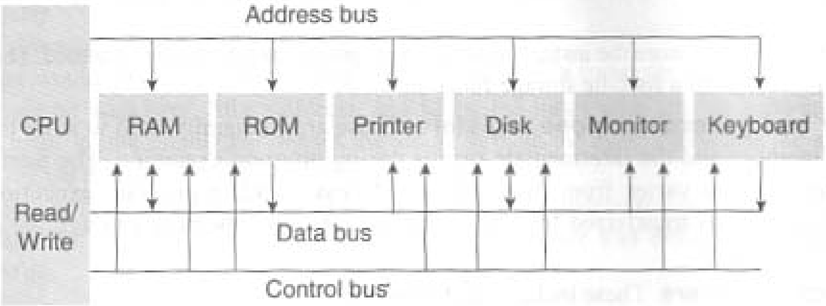

Figure 1.18 Buses with a computer system

The BIU puts the contents of the program counter on the address bus. Note that the content of the program counter is the address of the next instruction to be executed. Once the memory receives an address from the BIU, it places the contents at that address on the data bus, which is then transferred to the IR of the processor through the MBR. At this time, the contents of the program counter are modified (e.g., incremented by 1) so that it now stores the address of the next instruction.

1.11 INPUT AND OUTPUT DEVICES

An input device is used to feed data and instructions into the computer. In the absence of an *input device*, a computer would have only been a display device. Correspondingly, any device that outputs/gives information from a computer is called an *output device*. Refer to Figure 1.19 which shows some basic I/O devices that are generally connected with our computer system.

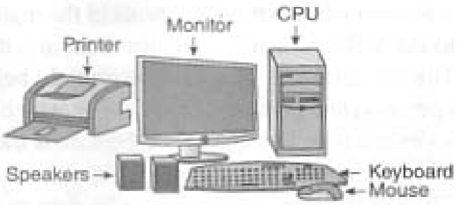

Figure 1.19 Basic I/O device computer system

Input Devices

Some of the input devices that are widely used by computer users to feed data or instruction to the computer are keyboard, mouse, trackball, joystick, stylus, touch screen, barcode reader, optical character recognition (OCR) device, optical mark recognition (OMR), MICR, web and digital cameras, etc.