

# Computer Modeling Applications for Environmental Engineers

## Second Edition

**Isam Mohammed Abdel-Magid**  
**Mohammed Isam Mohammed Abdel-Magid**

# Computer Modeling Applications for Environmental Engineers



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

# Computer Modeling Applications for Environmental Engineers

## Second Edition

Isam Mohammed Abdel-Magid  
Mohammed Isam Mohammed Abdel-Magid



**CRC Press**

Taylor & Francis Group  
Boca Raton London New York

---

CRC Press is an imprint of the  
Taylor & Francis Group, an **informa** business



CRC Press  
Taylor & Francis Group  
6000 Broken Sound Parkway NW, Suite 300  
Boca Raton, FL 33487-2742

© 2017 by Taylor & Francis Group, LLC  
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed on acid-free paper

International Standard Book Number-13: 978-1-4987-7654-7 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access [www.copyright.com](http://www.copyright.com) (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

**Trademark Notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

---

#### Library of Congress Cataloging-in-Publication Data

---

Names: 'Abd al-Majid, 'Iṣām Muḥammad, author. | Abdel-Magid, Mohammed Isam Mohammed, author.  
Title: Computer modeling applications for environmental engineers / Isam Mohammed Abdel-Magid Ahmed and Mohammed Isam Mohammed Abdel-Magid.  
Other titles: Modeling methods for environmental engineers.  
Description: Second edition. | Boca Raton : Taylor & Francis, CRC Press, 2017. | Revised edition of: Modeling methods for environmental engineers / Isam Mohammed Abdel-Magid, Abdel-Wahid Hago Mohammed, Donald R. Rowe. 1997. | Includes bibliographical references.  
Identifiers: LCCN 2016054370 | ISBN 9781498776547 (print : alk. paper)  
Subjects: LCSH: Pollution control equipment--Design and construction--Data processing. | Pollution control equipment--Computer simulation. | Sanitary engineering--Data processing. | Environmental management--Data processing. | Sanitary engineering--Computer simulation.  
Classification: LCC TD192 .A25 2017 | DDC 628.0285--dc23  
LC record available at <https://lcn.loc.gov/2016054370>

---

Visit the Taylor & Francis Web site at  
<http://www.taylorandfrancis.com>

and the CRC Press Web site at  
<http://www.crcpress.com>

# *Dedication*

---

## **Dedications to the Second Edition**

*To their souls, Mother Umsalama Al-Tahir Ahmed, Father Mohammed  
Abdel-Magid Ahmed, and Wife Eng. Layla Salih Mahmoud*

*I. M. Abdel-Magid*

*To mom, till we meet again*

*M. I. M. Abdel-Magid*

## **Dedications to the First Edition**

*To all my former teachers and professors*

*I. M. Abdel-Magid*

*To all knowledge seekers*

*A. W. Hago*

*To Amjad and Nasra*

*D. R. Rowe*



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# Contents

List of Figures .....	xv
List of Tables .....	xix
List of Computer Programs .....	xxi
Appendix Contents .....	xxv
Preface .....	xxvii
Preface and Acknowledgment to First Edition .....	xxxix
Authors .....	xxxiii
Authors of First Edition .....	xxxv
List of Symbols and Abbreviations .....	xxxvii
List of Acronyms .....	xliii

<b>Chapter 1</b>	<b>Programming Concepts .....</b>	<b>1</b>
1.1	How It All Started .....	1
1.2	Components of a Computer .....	2
1.3	Types of Programs .....	5
1.3.1	OS (Supervisor Programs) .....	6
1.3.2	Utility Programs (User Programs) .....	6
1.4	Programming and Languages .....	6
1.5	The Open-Source Movement .....	10
1.6	About the .NET Framework .....	12
1.7	What Is VS.NET? .....	12
1.8	Fundamentals of the VB.NET Language .....	13
1.8.1	Variables and Constants .....	14
1.8.2	Operators .....	15
1.8.3	The REM Statement (Remark or the Single Quote, “ ’ ”) .....	15
1.8.4	The End Statement .....	15
1.8.5	The GoTo Statement (Transfer of Control and Unconditional Branching) .....	15
1.8.6	Intrinsic Functions (Library or Built-In Functions) .....	15
1.8.7	The IF . . . THEN Construct (Conditional Transfer) .....	16
1.8.8	The CASE Construct .....	16
1.8.9	The FOR . . . TO . . . NEXT Block (Building a Loop) .....	16
1.8.10	DO . . . LOOP .....	17
1.8.11	The STRING Class .....	17
1.8.12	The FORMAT Family of Functions .....	17
1.8.13	The Drawing Functions .....	17
1.8.14	The DIM Statement .....	17
1.8.15	Functions and Subroutines .....	18
1.8.16	The MSGBOX and INPUTBOX Functions .....	18
1.8.17	Objects and Their Properties .....	18
1.9	Programs on the Accompanying CD-ROM .....	19
1.10	Homework Problems in Programming Concepts .....	21
	References .....	21

<b>Chapter 2</b>	<b>Computer Modeling Applications for Water and Wastewater Properties .....</b>	<b>23</b>
2.1	Introduction .....	23
2.2	Properties of Water and Wastewater .....	23
2.3	Physical Properties .....	23
2.3.1	Temperature .....	23
2.3.2	Conductivity .....	27
2.3.3	Salinity .....	29
2.3.4	Solids Content .....	29

2.3.5	Total Dissolved Solids .....	31
2.3.6	Density, Specific Volume, Specific Weight, and Specific Gravity .....	31
2.3.7	Viscosity (Rheological Properties).....	34
2.3.8	Surface Tension and Capillary Rise .....	36
2.3.9	Bulk Modulus (Bulk Modulus of Elasticity) .....	41
2.4	Chemical Characteristics.....	43
2.4.1	Hydrogen Ion Concentration .....	43
2.4.2	Hardness .....	45
2.5	Biological Properties .....	51
2.5.1	Dissolved Oxygen.....	51
2.5.2	Biochemical Oxygen Demand.....	53
2.6	Radioactivity.....	60
2.7	Homework Problems in Water and Wastewater Properties.....	64
2.7.1	Discussion Problems .....	64
2.7.2	Specific Mathematical Problems.....	65
	References .....	66

<b>Chapter 3</b>	<b>Computer Modeling Applications for Water Resources, Usage, Groundwater, and Water Storage and Distribution.....</b>	<b>67</b>
3.1	Introduction .....	67
3.2	Water Sources and Footprints .....	67
3.2.1	Introduction .....	67
3.2.2	Water Footprint .....	67
3.2.3	Source Selection .....	68
3.3	Population Growth and Consumption Design Rates .....	69
3.4	Fire Demand.....	72
3.5	Groundwater Flow .....	75
3.6	Water Storage .....	86
3.6.1	Water Storage .....	86
3.6.2	Mass Curve (Rippl Diagram and <i>S</i> -Curve).....	87
3.7	Water Distribution .....	92
3.7.1	Introduction .....	92
3.7.2	Patterns of Pipelines in Water Networks.....	92
3.7.3	Viscous Flow in Closed Conduits .....	92
3.7.3.1	Fundamental Equations for an Incompressible Flow .....	92
3.7.4	Flow in Pipes.....	93
3.7.4.1	Laminar Flow (Hagen–Poiseuille Flow) .....	93
3.7.4.2	Turbulent Flow.....	94
3.7.4.3	Minor Losses .....	98
3.7.5	Pipes in Series .....	98
3.7.5.1	Equivalent-Velocity-Head Method.....	98
3.7.5.2	Equivalent-Length Method.....	100
3.7.6	Pipes in Parallel (Pipe Network).....	102
3.7.6.1	Disadvantages of the Hardy Cross Method .....	110
3.7.6.2	Analysis of Pipe Networks by the Finite Element Method .....	110
3.7.6.3	Advantages of the Finite Element Method .....	118
3.8	Structural Design of Storage Tanks.....	118
3.8.1	Analysis of Forces .....	118
3.8.1.1	Circular Tanks .....	118
3.8.1.2	Rectangular Tanks .....	121
3.8.2	Design of the Section .....	123
3.9	Homework Problems in Water Resources, Water Storage, and Water Distribution.....	127
3.9.1	Discussion Problems .....	127
3.9.2	Specific Mathematical Problems.....	128
	References .....	129

<b>Chapter 4</b>	<b>Computer Modeling Applications for Water Treatment.....</b>	<b>131</b>
4.1	Water Treatment Systems.....	131
4.2	Aims of Water Treatment.....	131
4.3	Screening.....	131
4.4	Sedimentation.....	132
4.4.1	Introduction .....	132
4.4.2	Types of Settling Phenomenon.....	132
4.4.3	Class I Settling .....	134
4.4.4	Settling Characteristics .....	138
4.4.5	The Ideal Sedimentation Basin .....	146
4.4.6	Elements That Reduce the Efficiency in the Performance of Sedimentation Basins.....	146
4.4.6.1	Turbulence .....	146
4.4.6.2	Bottom Scour.....	147
4.4.6.3	Nonuniform Velocity Distribution and Short-Circuiting .....	147
4.4.7	Design of the Settling Zone.....	149
4.4.8	Settling of Flocculent Particles .....	150
4.4.9	Analysis of Flocculent Settling .....	150
4.4.10	General Design Considerations .....	150
4.4.10.1	Rectangular Tanks.....	150
4.4.10.2	Circular Settling Tanks (Dorr Settling Tanks).....	151
4.5	Flocculation and Coagulation .....	155
4.5.1	Introduction .....	155
4.5.2	Electrokinetics of Coagulation.....	155
4.5.3	Design Parameters.....	157
4.5.4	Design of Flocculator .....	157
4.6	Aeration and Gas Transfer .....	161
4.7	Filtration.....	164
4.7.1	Introduction .....	164
4.7.2	Filtration Theory .....	167
4.7.3	Clogging of the Filter Bed.....	178
4.7.4	Backwashing a Rapid Sand Filter.....	179
4.8	Methods of Desalination .....	181
4.8.1	Introduction .....	181
4.8.2	Distillation.....	181
4.8.3	Osmosis .....	184
4.8.4	Electrodialysis .....	188
4.9	Disinfection Process.....	190
4.9.1	Introduction .....	190
4.9.2	Chlorination.....	190
4.10	Corrosion.....	192
4.10.1	Introduction .....	192
4.10.2	Corrosion Potential Indicators.....	192
4.10.2.1	Langelier Index.....	192
4.10.2.2	Ryznar Index .....	192
4.10.2.3	Aggressiveness Index .....	193
4.10.3	Rate of Corrosion .....	195
4.11	Homework Problems in Computer Modeling Applications for Water Treatment .....	198
4.11.1	Discussion Problems.....	198
4.11.2	Specific Mathematical Problems .....	198
	References .....	199



<b>Chapter 5</b>	<b>Computer Modeling Applications for Wastewater Collection System and Treatment Technology and Disposal .....</b>	<b>201</b>
5.1	Introduction .....	201
5.2	Sewers and Sewerage Systems .....	202
5.2.1	Advantages and Disadvantages of Sewer Systems .....	202
5.2.1.1	Separate System .....	202
5.2.1.2	Combined System .....	202
5.2.1.3	Other Considerations .....	203
5.2.2	Flow Rates of Sanitary Wastewater .....	203
5.2.3	Storm Water .....	203
5.2.4	Hydraulics of Sewers .....	206
5.2.5	Flow Friction Formulae .....	207
5.2.6	Design Computations .....	212
5.2.6.1	Capacity of Flow Estimates .....	212
5.2.6.2	Self-Cleansing Velocities .....	217
5.2.7	Summary of Sewer System Design .....	217
5.2.8	Corrosion in Sanitary Sewers .....	218
5.2.8.1	Sulfide Buildup Estimates .....	218
5.2.8.2	Filled Pipe Conditions .....	220
5.2.8.3	Partially Filled Pipe Conditions .....	220
5.3	Wastewater Disposal for Rural Inhabitants .....	220
5.3.1	Septic Tanks .....	220
5.3.1.1	Introduction .....	220
5.3.1.2	Designing Septic Tanks .....	221
5.3.1.3	Septic Tank Effluents .....	222
5.3.2	Imhoff Tanks .....	235
5.4	Sources and Evaluation of Wastewater Flow Rates .....	236
5.5	Concept of PE .....	239
5.6	Reasons for Treating Wastewater .....	241
5.7	Wastewater Treatment Unit Operations and Processes .....	241
5.8	Preliminary Treatment: Grit Removal .....	241
5.9	Secondary Treatment (Aerobic and Biological) .....	244
5.9.1	Introduction .....	244
5.9.2	Suspended Growth Systems (Aerobic Suspended Growth Process): Activated Sludge Process .....	245
5.9.3	Activated Sludge Process Kinetics .....	245
5.9.4	Factors Affecting the Activated Sludge Process .....	247
5.9.4.1	Volume and Sludge Loadings .....	247
5.9.4.2	Sludge Age, Mean Cell Residence Time, Solids Retention Time, or Cell Age .....	247
5.9.4.3	Effects of SVI (Mohlman Sludge Volume Index) .....	250
5.9.4.4	SDI or Donaldson Index .....	250
5.10	Attached Growth Treatment Processes: Trickling Filter .....	252
5.10.1	Introduction .....	252
5.10.2	Recirculation to the Trickling Filter .....	253
5.10.3	Efficiency of a Trickling Filter (BOD Removal Efficiency) .....	253
5.10.3.1	NRC Formula .....	253
5.10.3.2	Velz Formula .....	254
5.10.3.3	Rankin Formula .....	254
5.10.3.4	Rumpf Formula .....	254
5.10.3.5	Eckenfelder Equation .....	254
5.10.3.6	Galler and Gotaas Equation .....	255
5.10.4	Overall Treatment Plant Efficiency .....	255
5.10.5	Trickling Filter Clarifier .....	259
5.11	Combined Suspended and Attached Growth Systems WSP, Lagoon, or Oxidation Pond .....	260
5.11.1	Introduction .....	260

5.11.2	WSP Design.....	261
5.11.2.1	General .....	261
5.11.2.2	Completely Mixed Conditions.....	261
5.11.2.3	Maturation WSP .....	263
5.12	Sludge Treatment and Disposal.....	266
5.12.1	Sludge Digestion .....	266
5.12.2	Sludge Dewatering.....	268
5.12.2.1	Introduction .....	268
5.12.2.2	Filtration of Sludge .....	269
5.12.2.3	Centrifugation.....	275
5.13	Wastewater Disposal .....	278
5.13.1	Dilution.....	278
5.13.2	Disposal into Natural Waters.....	280
5.13.2.1	Introduction .....	280
5.13.2.2	Oxygen Renewal and Depletion in Rivers.....	280
5.13.2.3	Dissolved Oxygen Sag Curves in Rivers .....	281
5.13.3	Disposal into Lakes .....	285
5.13.4	Disposal in Estuaries .....	287
5.14	Homework Problems in Wastewater Collection System, Wastewater Treatment Technology, and Disposal.....	288
5.14.1	Discussion Problems.....	288
5.14.2	Specific Mathematical Problems .....	289
	References .....	290

<b>Chapter 6</b>	<b>Computer Modeling Applications for Municipal Solid Waste Classification, Quantities, Properties, Collection, Processing, Material Separation, and Cost Estimates .....</b>	<b>293</b>
6.1	Introduction .....	293
6.2	Physical Properties of Solid Waste.....	297
6.2.1	Moisture Content.....	297
6.2.2	Particle Size .....	298
6.2.3	Permeability of Compacted Waste.....	303
6.2.4	Apparent Density .....	303
6.2.5	Angle of Repose.....	304
6.2.6	Size of Reduction in Volume (Reduction Volume) .....	304
6.2.7	Material Abrasiveness.....	306
6.3	Chemical Properties of Solid Waste.....	306
6.3.1	Fusion Point of Ash (Ash Fusibility).....	306
6.3.2	Proximate Analysis .....	307
6.3.3	Ultimate Analysis .....	307
6.3.4	Volatile Solids .....	307
6.3.5	Heat Value of Refuse.....	307
6.4	Solid Waste Collection .....	320
6.5	Solid Waste Collection, Processing, and Material Separation .....	325
6.6	Sanitary Landfill .....	330
6.7	Biochemical Processes, Combustion, and Energy Recovery .....	335
6.8	Cost Estimates for Solid Waste Facilities.....	346
6.8.1	Capital Cost and Capital Recovery Factor.....	346
6.8.2	Present Worth Value and Present Worth Factor .....	348
6.8.3	Sinking Fund and Sinking Fund Factor.....	348
6.8.4	Total Cost .....	351
6.9	Homework Problems in Computer Modeling Applications for Municipal Solid Waste Classification, Quantities, Properties, Collection, Processing, Material Separation, and Cost Estimates .....	353
6.9.1	Discussion Problems .....	353
6.9.1.1	Solid Waste Research.....	353
6.9.1.2	Type of Waste.....	353

6.9.1.3	Solid Waste Types .....	353
6.9.1.4	Solid Waste Amount .....	354
6.9.1.5	Solid Waste Properties .....	354
6.9.1.6	Solid Waste Collection .....	354
6.9.1.7	Solid Waste Recycling and Reuse .....	355
6.9.1.8	Solid Waste Treatment .....	355
6.9.1.9	Solid Waste Finance and Costs .....	355
6.9.2	Specific Mathematical Problems .....	357
6.9.2.1	Amount of Solid Waste .....	357
6.9.2.2	Solid Waste Properties .....	357
6.9.2.3	Solid Waste Collection .....	358
6.9.2.4	Solid Waste Treatment and Disposal .....	358
6.9.2.5	Solid Waste Finance and Cost Analysis .....	359
	References .....	359

## **Chapter 7** Computer Modeling Applications for Air Pollution Control Technology..... 361

7.1	Introduction .....	361
7.2	Fundamental Concepts in Air Pollution.....	363
7.2.1	Units of Measurement .....	363
7.2.2	Mole and Mole Fraction .....	365
7.2.3	Basic Gas Laws .....	368
7.2.3.1	Boyle's Law .....	368
7.2.3.2	Charles's Law .....	370
7.2.3.3	Ideal Gas Law .....	374
7.2.4	Van der Waal's Equation.....	378
7.2.5	Dalton's Law of Partial Pressures .....	381
7.2.6	Henry's Law .....	382
7.2.7	Clausius–Clapeyron Equation.....	384
7.2.8	Reynolds Numbers .....	386
7.2.9	Stoke's Law .....	387
7.2.10	Particle Size Distribution—Log Normal .....	390
7.3	Air Pollution Control Technology .....	392
7.3.1	Settling Chambers.....	392
7.3.2	Cyclones .....	396
7.3.3	Electrostatic Precipitators .....	403
7.3.4	Venturi Scrubbers .....	408
7.3.5	Baghouse or Fabric Filters .....	413
7.3.6	Combustion Processes.....	417
7.3.7	Packed Column Absorption Towers.....	423
7.3.8	Adsorption.....	427
7.4	Air Quality Modeling.....	436
7.4.1	Introduction.....	436
7.4.2	Effective Stack Height .....	436
7.4.3	Dispersion Models .....	440
7.4.3.1	Basic Gaussian Dispersion Model.....	440
7.4.3.2	U.S. EPA Dispersion Models .....	447
7.4.3.3	SCREEN2 Dispersion Model.....	447
7.5	Homework Problems in Computer Modeling Applications for Air Pollution Control Technology.....	448
7.5.1	Discussion Problems .....	448
7.5.2	Specific Mathematical Problems .....	450
	References .....	452

## **Chapter 8** Computer Modeling Applications for Noise Pollution and Abatement ..... 455

8.1	Sound and Noise.....	455
8.2	Effects of Noise .....	459

8.3	Decibel Scale.....	459
8.4	Sound Fields from Different Sound Sources.....	461
8.5	Noise Measurement.....	465
8.6	Homework Problems in Computer Modeling Applications for Noise Pollution and Abatement .....	467
8.6.1	Discussion Problems.....	467
8.6.2	Special Mathematical Problems .....	467
	References .....	467
<b>Appendices.....</b>		<b>469</b>
<b>Index.....</b>		<b>479</b>



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# List of Figures

<b>Figure 1.1</b>	Components of a computing machine .....	2
<b>Figure 1.2</b>	Components of a computer (von Neumann model).....	2
<b>Figure 1.3</b>	The system bus. I/O, input/output .....	2
<b>Figure 1.4</b>	QWERTY keyboard layout .....	2
<b>Figure 1.5</b>	Structure of a computer mouse .....	2
<b>Figure 1.6</b>	Components of a PC system.....	3
<b>Figure 1.7</b>	Different levels of abstraction in the programming model .....	7
<b>Figure 1.8</b>	Steps of programming .....	9
<b>Figure 1.9</b>	Form code and design.....	14
<b>Figure 1.10</b>	Changing the value of a variable .....	14
<b>Figure 1.11</b>	Declaring an array .....	18
<b>Figure 2.1</b>	Types of solids .....	29
<b>Figure 2.2</b>	Total dissolved solids as related to water quality .....	31
<b>Figure 2.3</b>	Different types of fluids .....	34
<b>Figure 2.4</b>	Viscosity concept .....	35
<b>Figure 2.5</b>	Surface tension: (a) capillary rise (wetting liquid); (b) capillary depression (nonwetting liquid).....	36
<b>Figure 2.6</b>	Variation of BOD with time.....	53
<b>Figure 2.7</b>	Radioactive decay .....	60
<b>Figure 3.1</b>	Schematic diagram of the hydrological cycle.....	68
<b>Figure 3.2</b>	Source selection.....	68
<b>Figure 3.3</b>	Water use cycle.....	69
<b>Figure 3.4</b>	Moderate rate of growth method.....	70
<b>Figure 3.5</b>	A supply and an observation well .....	80
<b>Figure 3.6</b>	Solution to Example 3.4 .....	80
<b>Figure 3.7</b>	Steady unconfined flow .....	84
<b>Figure 3.8</b>	An elevated water storage tank .....	87
<b>Figure 3.9</b>	Mass diagram .....	88
<b>Figure 3.10</b>	Mass diagram for Example 3.6.....	89
<b>Figure 3.11</b>	Pipe flow velocity profile and shear stress distribution .....	93
<b>Figure 3.12</b>	Pipes in series .....	98
<b>Figure 3.13</b>	Pipes in parallel .....	104
<b>Figure 3.14</b>	Example 3.11.....	104
<b>Figure 3.15</b>	Pipe element .....	110
<b>Figure 3.16</b>	Pipe network.....	110



<b>Figure 4.1</b>	Sedimentation and flotation.....	132
<b>Figure 4.2</b>	Sedimentation tanks: (a) schematic diagram of a rectangular horizontal sedimentation tank; (b) schematic diagram of a circular sedimentation tank.....	133
<b>Figure 4.3</b>	Settling column test.....	138
<b>Figure 4.4</b>	Cumulative frequency distribution plot.....	139
<b>Figure 4.5</b>	Schematic diagram of an ideal settling basin .....	146
<b>Figure 4.6</b>	Round-the-end baffles in a sedimentation tank .....	147
<b>Figure 4.7</b>	Rectangular sedimentation tank.....	150
<b>Figure 4.8</b>	Radial flow sedimentation tank.....	152
<b>Figure 4.9</b>	Schematic diagram of a sand filter .....	167
<b>Figure 4.10</b>	Summary of various desalination methods. TDS, total dissolved solids.....	181
<b>Figure 4.11</b>	Pressure regulation with a multiple-effect distiller .....	182
<b>Figure 4.12</b>	Osmosis process .....	184
<b>Figure 4.13</b>	RO process.....	185
<b>Figure 4.14</b>	Schematic representation of an electrodialysis process.....	188
<b>Figure 4.15</b>	Types of corrosion .....	195
<b>Figure 4.16</b>	Classification of corrosion .....	196
<b>Figure 5.1</b>	Phases of projects of a sanitary sewer .....	203
<b>Figure 5.2</b>	Cross section of a sewer .....	212
<b>Figure 5.3</b>	Regions in a septic tank .....	221
<b>Figure 5.4</b>	A schematic diagram of a septic tank: (a) elevation; (b) side view; (c) plan .....	221
<b>Figure 5.5</b>	Activated sludge process .....	245
<b>Figure 5.6</b>	Schematic diagram of a trickling filter process.....	252
<b>Figure 5.7</b>	(a) High- and (b) low-rate trickling filters .....	253
<b>Figure 5.8</b>	Schematic diagram of a facultative WSP .....	261
<b>Figure 5.9</b>	Specific resistance apparatus.....	270
<b>Figure 5.10</b>	Wastewater disposal into a stream (dilution).....	278
<b>Figure 5.11</b>	Disposal into a lake .....	285
<b>Figure 6.1</b>	Angle of repose.....	304
<b>Figure 6.2</b>	Binary separator .....	327
<b>Figure 6.3</b>	Energy flow in a combustor.....	343
<b>Figure 7.1</b>	Settling chamber.....	396
<b>Figure 7.2</b>	(a) Typical dimension for single, high-efficiency cyclone. (b) Nomenclature for a single, high-efficiency cyclone.....	398
<b>Figure 7.3</b>	Cyclone efficiency versus particle size ratio.....	399
<b>Figure 7.4</b>	Schematic diagram of an ESP .....	403
<b>Figure 7.5</b>	High-voltage electrostatic precipitators: a) plate type precipitator and b) Tube type precipitator .....	404
<b>Figure 7.6</b>	Venturi wet collector: a) isometric view b) sectional view.....	409

<b>Figure 7.7</b>	Baghouse filter .....	414
<b>Figure 7.8</b>	The gaseous contaminants that can be removed by catalytic system at both room temperatures and elevated temperatures. (a) Teledyne's room-temperature catalyst at room temperature effectively removes carbon monoxide (CO), ozone (O <sub>3</sub> ), hydrogen sulfide (H <sub>2</sub> S), sulfur dioxide (SO <sub>2</sub> ), formaldehyde (HCHO), and nitrogen dioxide (NO <sub>2</sub> ). (b) Teledyne's room-temperature catalyst at elevated temperature effectively removes hydrogen cyanide (HCN), propane (C <sub>3</sub> H <sub>8</sub> ), methane (CH <sub>4</sub> ), and isobutane (C <sub>4</sub> H <sub>10</sub> ).....	418
<b>Figure 7.9</b>	(a) Countercurrent-packed column: a typical countercurrent-packed column absorber system. (b) Countercurrent-packed column. (c) Countercurrent-packed column: graphical presentation of the packed column .....	424
<b>Figure 7.10</b>	Straight-line form of the Freundlich adsorption isotherm.....	428
<b>Figure 7.11</b>	(a) Schematic of a regenerative adsorption-type system. (b) Schematic of a nonregenerative adsorption-type system .....	429
<b>Figure 7.12</b>	(a) An example of plume rise calculations. (b) An example of the reduced ground-level contaminant concentration due to doubling of the effective stack height .....	437
<b>Figure 7.13</b>	(a) Coordinate system showing Gaussian distribution in the horizontal and vertical directions. (b) Flat terrain for Gaussian distribution .....	441
<b>Figure 7.14</b>	(a) Horizontal dispersion coefficient, $\sigma_y$ , as a function of downwind distance from the source. (b) Vertical dispersion coefficient, $\sigma_z$ , as a function of downward distance from the source.....	441
<b>Figure 7.15</b>	(a) Schematic of ground-level concentration underneath the center line of the plume. (b) Schematic of contaminants generated under ground-level conditions .....	442
<b>Figure 8.1</b>	Sound wave travel in space.....	456
<b>Figure 8.2</b>	Examples of environmental noise on a decibel scale.....	456
<b>Figure 8.3</b>	Amplitude and frequency.....	458
<b>Figure 8.4</b>	Frequency, pitch, amplitude, and sound level .....	459
<b>Figure 8.5</b>	Schematic diagram of a sound level meter.....	466
<b>Figure 8.6</b>	Important means for reducing noise pollution .....	466



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# List of Tables

<b>Table 1.1</b>	Some of the mathematical functions used in VB (part of the <code>System.Math</code> namespace).....	16
<b>Table 1.2</b>	Relational (Boolean or logical) operators (Example column uses $A = 1$ and $B = 0$ ) .....	16
<b>Table 1.3</b>	Some of the common control properties .....	19
<b>Table 1.4</b>	Summary of VB.NET statements, functions, and commands .....	20
<b>Table 2.1</b>	Principal cations and anions associated with hardness.....	45
<b>Table 2.2</b>	Degrees of hardness .....	46
<b>Table 2.3</b>	Sewage strength in terms of BOD and COD .....	54
<b>Table 3.1</b>	Values of the coefficient $C_f$ .....	72
<b>Table 3.2</b>	Residential fire flows .....	72
<b>Table 3.3</b>	Fire hydrant spacing .....	72
<b>Table 3.4</b>	Fire flow duration .....	72
<b>Table 3.5</b>	Typical total porosities for selected materials.....	76
<b>Table 3.6</b>	Typical hydraulic conductivities for selected materials .....	76
<b>Table 3.7</b>	Turbulent and laminar flows.....	92
<b>Table 3.8</b>	Iterations.....	105
<b>Table 4.1</b>	Comparison of rapid and slow sand filters .....	165
<b>Table 4.2</b>	Dissolved solids content for different waters .....	181
<b>Table 4.3</b>	Activation energies ( $E'$ ) for aqueous chlorine .....	191
<b>Table 4.4</b>	Aggressiveness index .....	193
<b>Table 5.1</b>	Typical values of $c$ for the rational method .....	204
<b>Table 5.2</b>	Design computations for partial pipe flow .....	213
<b>Table 5.3</b>	Values of $b'$ .....	217
<b>Table 5.4</b>	Recommended minimum velocities .....	217
<b>Table 5.5</b>	Sulfide condition for different values of $f$ .....	218
<b>Table 5.6</b>	Water and wastewater treatment units.....	241
<b>Table 5.7</b>	Grit chamber typical design parameters .....	243
<b>Table 5.8</b>	SLR for certain treatment units.....	247
<b>Table 5.9</b>	Mean cell residence time for various modifications of the activated sludge process.....	248
<b>Table 5.10</b>	Activated sludge settleability according to SVI .....	250
<b>Table 5.11</b>	Design criteria for conventional activated sludge.....	252
<b>Table 5.12</b>	Typical design information for trickling filter.....	260
<b>Table 5.13</b>	Design information for WSPs.....	266
<b>Table 5.14</b>	Design information for a conventional anaerobic digester .....	267
<b>Table 5.15</b>	Sludge characteristics as related to specific resistance.....	270
<b>Table 5.16</b>	River classification patterns.....	280

<b>Table 5.17</b>	Reaeration constants.....	280
<b>Table 6.1</b>	Sources and types of production units of solid waste .....	294
<b>Table 6.2</b>	Moisture content of uncompacted refuse components.....	298
<b>Table 6.3</b>	Ideal data for final analysis of components of a combustible MSW .....	307
<b>Table 6.4</b>	General overview of solid waste properties and characteristics of concern .....	320
<b>Table 6.5</b>	Summary of selected compounding factors.....	351
<b>Table 6.6</b>	Capital recovery factors for an interest rate of 6.125% .....	351
<b>Table 7.1</b>	National ambient air quality standards.....	362
<b>Table 7.2</b>	Air quality standards for selected pollutants in several countries ( $\mu\text{g}/\text{m}^3$ ) .....	362
<b>Table 7.3</b>	Sources of air pollutants in the United States, 1993 .....	363
<b>Table 7.4</b>	EPA recommended units of measurement .....	364
<b>Table 7.5</b>	Van der Waal's constants for some gases .....	378
<b>Table 7.6</b>	Henry's law constants for some gases in water at 25°C .....	382
<b>Table 7.7</b>	Values of $K$ for Calculating Pressure Drop through Cyclone.....	399
<b>Table 7.8</b>	Pasquill stability types .....	437
<b>Table 7.9</b>	Average values of wind profile power law exponents ( $k$ ) by stability class.....	438
<b>Table 8.1</b>	Speed of sound in different media at 0°C .....	456
<b>Table 8.2</b>	Speed of sound in air at different temperatures.....	456

---

# List of Computer Programs

2.1	(Chap2.1\Form1.vb): Converting temperature to different scales .....	24
2.2	(Chap2.2\Form1.vb): Electric conductivity for TDS value .....	27
2.3	(Chap2.3\Form1.vb): Solids content.....	29
2.4	(Chap2.4\Form1.vb): Density and specific gravity.....	32
2.5	(Chap2.5\Form1.vb): Kinematic viscosity from dynamic viscosity .....	35
2.6	(Chap2.6\Form1.vb): Capillary heights in surface tension .....	37
2.7	(Chap2.7\Form1.vb): Bulk modulus .....	42
2.8	(Chap2.8\Form1.vb): Hydrogen ion concentration ascertaining .....	44
2.9	(Chap2.9\Form1.vb): Hardness estimation.....	47
2.10	(Chap2.10\Form1.vb): Dissolved oxygen concentration.....	52
2.11	(Chap2.11\Form1.vb): BOD computations .....	55
2.12	(Chap2.12\Form1.vb): BOD determination according to Thomas .....	58
2.13	(Chap2.13\Form1.vb): Radioactivity rate of decay.....	61
3.1	(Chap3.1\Form1.vb): Population estimates.....	71
3.2	(Chap3.2\Form1.vb): Fire demand approximation .....	73
3.3	(Chap3.3\Form1.vb): Flow velocity within aquifers .....	77
3.4	(Chap3.4\Form1.vb): Depressions in observation wells .....	81
3.5	(Chap3.5\Form1.vb): Steady unconfined aquifer drawdown computations .....	85
3.6	(Chap3.6\Form1.vb): Reservoir capacity.....	89
3.7	(Chap3.7\Form1.vb): Viscous flow in pipes.....	93
3.8	(Chap3.8\Form1.vb): Head loss due to friction in pipes.....	95
3.9	(Chap3.9\Form1.vb): Equivalent-velocity-head method.....	98
3.10	(Chap3.10\Form1.vb): Equivalent-length method.....	101
3.11	(Chap3.11\Form1.vb): Hardy cross method .....	106
3.12	(Chap3.12\Form1.vb): FEM analysis of pipe networks .....	113
3.13	(Chap3.13\Form1.vb): Analysis of circular tanks.....	120
3.14	(Chap3.14\Form1.vb): Analysis of rectangular tanks .....	122
3.15	(Chap3.15\Form1.vb): Structural design of tanks .....	124
4.1	(Chap4.1\Form1.vb): Settling velocity for spherical discrete particles .....	135
4.2	(Chap4.2\Form1.vb): Suspended solids in a settling tank effluent.....	140
4.3	(Chap4.3\Form1.vb): Stability and effects of turbulence in a settling tank.....	148
4.4	(Chap4.4\Form1.vb): Design a circular sedimentation tank .....	153
4.5	(Chap4.5\Form1.vb): Electrokinetics of coagulation .....	156
4.6	(Chap4.6\Form1.vb): Power needed for a flocculator system .....	158
4.7	(Chap4.7\Form1.vb): Efficiency coefficient of a cascade aerator.....	162



<b>4.8</b>	(Chap4.8\Form1.vb): Number of filters, filtration area, and filter areas .....	166
<b>4.9</b>	(Chap4.9\Form1.vb): Head loss in filters.....	168
<b>4.10</b>	(Chap4.10\Form1.vb): Filtration rate for a given head loss .....	174
<b>4.11</b>	(Chap4.11\Form1.vb): Concentration of impurities of effluents of a filter .....	179
<b>4.12</b>	(Chap4.12\Form1.vb): Depth of bed expansion.....	180
<b>4.13</b>	(Chap4.13\Form1.vb): Temperature difference in multiphase still .....	183
<b>4.14</b>	(Chap4.14\Form1.vb): Osmotic pressure per unit volume for salt solutions.....	185
<b>4.15</b>	(Chap4.15\Form1.vb): Difference in osmotic pressure across membranes.....	186
<b>4.16</b>	(Chap4.16\Form1.vb): Electrodialysis maximum current density.....	189
<b>4.17</b>	(Chap4.17\Form1.vb): Chlorination time of kill.....	191
<b>4.18</b>	(Chap4.18\Form1.vb): Corrosion potential indicators .....	193
<b>4.19</b>	(Chap4.19\Form1.vb): Rate of corrosion of a metallic surface .....	196
<b>5.1</b>	(Chap5.1\Form1.vb): Wastewater collection .....	204
<b>5.2</b>	(Chap5.2\Form1.vb): Velocity of flow in pipes .....	208
<b>5.3</b>	(Chap5.3\Form1.vb): Capacity of flow estimates .....	215
<b>5.4</b>	(Chap5.4\Form1.vb): Sulfide build up indicator .....	219
<b>5.5</b>	(Chap5.5\Form1.vb): Design of septic tanks.....	223
<b>5.6</b>	(Chap5.6\Form1.vb): Dry weather flow, DWF .....	237
<b>5.7</b>	(Chap5.7\Form1.vb): Different per capita wastewater flows .....	238
<b>5.8</b>	(Chap5.8\Form1.vb): Population equivalent of industrial establishments.....	240
<b>5.9</b>	(Chap5.9\Form1.vb): Designing a grit chamber .....	243
<b>5.10</b>	(Chap5.10\Form1.vb): Determination of growth rate of microorganisms.....	246
<b>5.11</b>	(Chap5.11\Form1.vb): Activated sludge process.....	249
<b>5.12</b>	(Chap5.12\Form1.vb): Sludge volume index.....	251
<b>5.13</b>	(Chap5.13\Form1.vb): Trickling filter efficiency .....	256
<b>5.14</b>	(Chap5.14\Form1.vb): Design of a waste stabilization pond .....	263
<b>5.15</b>	(Chap5.15\Form1.vb): Retention time and volume of a pond.....	265
<b>5.16</b>	(Chap5.16\Form1.vb): Volumetric gas production rate in digesters .....	267
<b>5.17</b>	(Chap5.17\Form1.vb): Specific resistance of sludge .....	271
<b>5.18</b>	(Chap5.18\Form1.vb): Compressibility of sludges.....	273
<b>5.19</b>	(Chap5.19\Form1.vb): Sludge dewatering by centrifugation.....	276
<b>5.20</b>	(Chap5.20\Form1.vb): Concentration of pollutants in rivers .....	279
<b>5.21</b>	(Chap5.21\Form1.vb): DO and BOD of river–wastewater mixtures .....	282
<b>5.22</b>	(Chap5.22\Form1.vb): Concentration of pollutants in lakes .....	286
<b>5.23</b>	(Chap5.23\Form1.vb): Waste concentration in estuaries.....	287
<b>6.1</b>	(Chap6.1\Form1.vb): Diversion of solid waste .....	295
<b>6.2</b>	(Chap6.2\Form1.vb): Moisture concentration of solid waste .....	299
<b>6.3</b>	(Chap6.3\Form1.vb): Volume reduction of compacted solid waste .....	305

<b>6.4</b>	(Chap6.4\Form1.vb): Chemical composition and energy content of solid waste.....	309
<b>6.5</b>	(Chap6.5\Form1.vb): Heat value .....	317
<b>6.6</b>	(Chap6.6\Form1.vb): Number of cans and compacted blocks produced by a family .....	321
<b>6.7</b>	(Chap6.7\Form1.vb): Number of collection vehicles needed by a community .....	323
<b>6.8</b>	(Chap6.8\Form1.vb): Capacity of a crushing cylinder .....	326
<b>6.9</b>	(Chap6.9\Form1.vb): Recoveries and the effectiveness of separation of a binary separator.....	329
<b>6.10</b>	(Chap6.10\Form1.vb): Peak gas production from a landfill .....	331
<b>6.12</b>	(Chap6.12\Form1.vb): Vertical flow .....	334
<b>6.13</b>	(Chap6.13\Form1.vb): Production of CO <sub>2</sub> and CH <sub>4</sub> .....	335
<b>6.14</b>	(Chap6.14\Form1.vb): Amount of water to be added to MSW to obtain the desired moisture content .....	338
<b>6.15</b>	(Chap6.15\Form1.vb): Stoichiometric oxygen and air required for combustion of a gas .....	339
<b>6.16</b>	(Chap6.16\Form1.vb): Temperature of stack gases .....	344
<b>6.17</b>	(Chap6.17\Form1.vb): Capital cost, capital recovery factor, and annual installments .....	347
<b>6.18</b>	(Chap6.18\Form1.vb): Amount of investment needed for a number of years .....	349
<b>6.19</b>	(Chap6.19\Form1.vb): Necessary funds to reach a certain value .....	350
<b>6.20</b>	(Chap6.20\Form1.vb): Refuse collection truck's costs.....	352
<b>7.1</b>	(Chap7.1\Form1.vb): Concentration of gaseous air contaminants .....	364
<b>7.4</b>	(Chap7.4\Form1.vb): Moles and mole fractions in a gaseous mixture.....	366
<b>7.5</b>	(Chap7.5\Form1.vb): Boyle's law.....	368
<b>7.6</b>	(Chap7.6\Form1.vb): Boyle's and Charles' laws.....	371
<b>7.8</b>	(Chap7.8\Form1.vb): Ideal gas law.....	375
<b>7.9</b>	(Chap7.9\Form1.vb): Van der Waal's equation .....	379
<b>7.11</b>	(Chap7.11\Form1.vb): Henry's law .....	382
<b>7.13</b>	(Chap7.13\Form1.vb): Clausis–Clapeyron equation .....	385
<b>7.14</b>	(Chap7.14\Form1.vb): Reynolds number .....	386
<b>7.15</b>	(Chap7.15\Form1.vb): Terminal velocity and Stoke's law .....	389
<b>7.17</b>	(Chap7.17\Form1.vb): Particle size distribution.....	392
<b>7.19</b>	(Chap7.19\Form1.vb): Settling chamber .....	397
<b>7.21</b>	(Chap7.21\Form1.vb): Efficiency of cyclone.....	400
<b>7.23</b>	(Chap7.23\Form1.vb): Electrostatic precipitator .....	405
<b>7.25</b>	(Chap7.25\Form1.vb): Venturi scrubber .....	410
<b>7.27</b>	(Chap7.27\Form1.vb): Design and operation of a baghouse .....	415
<b>7.28</b>	(Chap7.28\Form1.vb): Catalytic system .....	419
<b>7.29</b>	(Chap7.29\Form1.vb): Multiple catalysts .....	421
<b>7.31</b>	(Chap7.31\Form1.vb): Packed columns absorption towers .....	426
<b>7.33</b>	(Chap7.33\Form1.vb): Freundlich adsorption isotherm .....	430
<b>7.35</b>	(Chap7.35\Form1.vb): Breakthrough working capacity of an adsorption bed .....	433
<b>7.37</b>	(Chap7.37\Form1.vb): Adsorbent's service time before regeneration.....	434

<b>7.39</b>	(Chap7.39\Form1.vb): Effective stack height .....	<a href="#">439</a>
<b>7.41</b>	(Chap7.41\Form1.vb): Downwind ground-level concentration.....	<a href="#">443</a>
<b>8.1</b>	(Chap8.1\Form1.vb): Sound velocity and wavelength.....	<a href="#">457</a>
<b>8.2</b>	(Chap8.2\Form1.vb): Sound pressure level .....	<a href="#">460</a>
<b>8.3</b>	(Chap8.3\Form1.vb): Net sound pressure level .....	<a href="#">461</a>
<b>8.4</b>	(Chap8.4\Form1.vb): Weekly average of noise exposures.....	<a href="#">463</a>
<b>8.5</b>	(Chap8.5\Form1.vb): Sound level at a distance .....	<a href="#">464</a>

---

# Appendix Contents

<b>A1</b>	Physical properties of water .....	<b>469</b>
<b>A2</b>	Saturation values of dissolved oxygen in water exposed to water-saturated air containing 20.9% oxygen under a pressure of 760 mmHg.....	<b>470</b>
<b>A3</b>	The periodic table of elements .....	<b>471</b>
<b>A4</b>	Hydraulic elements graph for circular sewers.....	<b>472</b>
<b>A5</b>	Alignment chart for flow in pipes .....	<b>473</b>
<b>A6</b>	Moody's diagram for Reynolds number versus relative roughness and friction factor .....	<b>474</b>
<b>A7</b>	Typical specific weight and moisture content data for residential, commercial, industrial, and agricultural wastes.....	<b>474</b>
<b>A8</b>	Conversion table.....	<b>476</b>



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# Preface

Books and text material in the field of environmental, public health, and sanitary engineering are numerous for addressing the principles, fundamentals, methods, models, concepts, designs, remedial activities, and science-related ideas. Nonetheless, rare are the books that deal with practical problems, problem solving in environmental engineering, and fundamental utilization of computer programming perceptions and applications.

The broad objectives of this book are to facilitate teaching, improve learning, shoulder training, and assist research in the application of computer programming based on well-established models, equations, formulae, and procedures used in the environmental engineering domain. Visual Basic .NET was selected and used as the programming language of choice throughout this book for its simplicity, comprehensive usage, and easiness. It is a language that was traditionally considered a model language of teaching purposes, not to undermine its wide range of applicability; it has been introduced into scripting, application programming (VBA), databases (VB and SQL), and web scripting (VBScript), to mention but a few uses.

Section I of first edition has been rewritten from scratch to reflect the advancements in computer modeling and software programming that took place over the past two decades. The first edition was written in Quick BASIC 4.5, a widely used and popular model programming language of the time. As BASIC faded away and gave the stage to its descendants Visual Basic and Visual Basic .NET, programming models and methods have changed vastly, new programming concepts emerged, and graphical user interface (GUI) systems took over command-line environments. Object-oriented programming (OOP) became the *de facto* programming model, and a whole set of new programming languages has erupted to exploit these concepts.

In addition, severe competition emerged between *Closed Source* and *Open Source* camps. Companies fighting to protect their royalties and copyrights by defending the *Closed Source* regime and closing arms with the emerging and ever-growing *Open Source* movement, which has novel concepts such as the freedom of source code and software *copylefting* (a pun on the word *copyright* to reflect the exact opposite meaning, that is, freedom of information). These concepts are changing the face of contemporary programming, more details on which can be found in Section I of the first edition.

Lastly, there are more operating systems (OSs) in active use today than when the first edition of the book was published. Microsoft™ and Apple® were once the only horses in the OS arena (not to undermine UNIX, but it was majorly a teaching OS used by universities and big institutions, not by regular users on their home portable computers); nowadays, other systems such as Linux, Solaris, Android, and iOS are actively contributing to the OS industry. As such, programmers are being faced with the problems of *system*

*compatibility* (ensuring software runs on different OSs) and *architecture compatibility* (ensuring software runs on different platforms such as x86 and x86\_64).

In the face of all these intermingled points, the authors were faced with the question of which programming language to use in writing the examples of this text:

1. Using a cross-platform language such as Java would seem logical. This ensures all the examples will work on the widest possible range of target machines without recompilation. In fact, this would have been the best choice had the intent been commercial software production, not teaching. Java is an OOP language by nature; and the performance hit of using the intermediate Java Virtual Machine (JVM) is negligible in most cases. This is all good, but using Java means the need to discuss OOP in depth, having a few sections on the JVM, the garbage collector, Java packages and built-in classes, and so on. This will shift the focus of the text from modeling computer applications for solving environmental engineering problems to a text on Java and OOP.
2. Using a simple procedural language such as C is perfect in a teaching setting like this, as it will remove all the headache of OOP, virtual machines, intermediate compilers, garbage collectors, and other things that are not primal to the teaching process. The reader will be able to focus on the specific modeling techniques under discussion. In addition, pure C is text based by nature, which is an important feature when programming console programs (such as Linux terminal programs or MS-DOS prompt-like programs). However, unfortunately, console programs have fallen out of favor to GUI programs, and most users (except power users and system administrators) feel uncomfortable when faced with the black screen and its blinking cursor. As such, using C alone is defective. One would usually combine C with a GUI toolkit; the most popular combination is using C++ with Qt (pronounced “cute”; stands for the Q toolkit). However, again, this will make it obligatory to include a whole chapter on Qt, its history, structure, object classes, and its uses, not to mention the mandatory chapter on core C programming. This will undoubtedly unnecessarily inflate the book out of scope.
3. Use of a traditional teaching language such as Pascal or Basic. This ensures that the focus of the text will remain on problem solving and application modeling. The chapter on programming itself will be supplementary, and habituated programmers can skip it and start directly on the text proper. This is the solution we opted to use throughout this text.



That being said, there are important points and topics pertaining to computer programming in general, and to modeling engineering applications specifically, which need to be discussed, or to the least, pointed to. These topics have been handled in Section I of the first edition.

The more specified objectives are proposed to accomplish the main learning outcomes for students enrolled in environmental engineering courses, which are as follows:

1. To help the reader in the design of treatment and disposal facilities
2. To introduce the basic concepts of computer design related to environmental engineering
3. To aid in teaching of environmental engineering unit operations and processes design
4. To improve the understanding by students and researchers in environmental engineering field
5. To improve the understanding of relevant physical, chemical, and biological processes, and their mutual relationships within various sanitation components
6. To promote environmental protection, improvement, and sustainability, enhancing the overall quality of life
7. To demonstrate effective problem-solving abilities
8. To utilize the current technology of waste collection, handling, treatment, final disposal, and management
9. To apply appropriate regulations and guidelines in safeguarding environmental conditions
10. To be able to design and apply necessary procedures and precautions to establish sound environmental engineering systems

Recently, the Department of Environmental Engineering of the College of Engineering at the Imam Abdulrahman Bin Faisal University, Kingdom of Saudi Arabia (KSA), sought the Accreditation Board for Engineering and Technology Accreditation Commission (<http://www.abet.org/>). As such, student outcomes were adhered to as per requirements. The program must have documented student outcomes that prepare graduates to attain the program educational objectives. Student outcomes are outcomes (a) through (k) plus any additional outcomes that may be articulated by the program are as follows:

- a. An ability to apply the knowledge of mathematics, science, and engineering
- b. An ability to design and conduct experiments, as well as to analyze and interpret data
- c. An ability to design a system, component, or process to meet the desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

- d. An ability to function on multidisciplinary teams
- e. An ability to identify, formulate, and solve engineering problems
- f. An understanding of professional and ethical responsibility
- g. An ability to communicate effectively
- h. The broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- i. A recognition of the need for, and an ability to engage in, lifelong learning
- j. A knowledge of contemporary issues
- k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

*Computer-Aided Modeling in Environmental Engineering* is chiefly intended as a supplement and a complementary guide to basic principles of environmental engineering. Nonetheless, it can be sourced as a stand-alone problem-solving text in the field. The book targets university students taking first-degree courses in environmental, construction, civil, mechanical, biomedical, and chemical engineering or related fields. The manuscript is estimated to have valuable benefits to postgraduate students and professional sanitary and environmental engineers. Equally, it is anticipated that the book will stimulate problem-solving learning and accelerate self-teaching. By writing such a script, it is hoped that the included worked examples, set problems, and compiled computer programs will warrant that the booklet is a treasured support to student-centered learning. To accomplish such goals, great attention was paid to offer solutions to selected problems in a well-defined, clear, and discrete layout exercising step-by-step procedures and explanation of the related solution employing crucial procedures, methods, approaches, equations, data, figures, and calculations.

**Chapter 6** has been added to the format of the book to address solid waste engineering and management. The first part deals with municipal solid waste classification, quantities, and properties stressing on introductory items, and physical and chemical properties of solid waste. The second part deals with solid waste collection, processing, material separation, and financing facilities. Treatment methods tackle incorporated sanitary landfill, biochemical processes, and combustion and energy recovery. This chapter highlights financing solid waste facilities. Likewise, this part of the book incorporates homework problems shouldering both discussion and special mathematical types of practical questions to be encountered. Computer programs were written for every stage discussed in this chapter.

The authors hope this work will contribute toward the achievement of United Nations Sustainable Development

Goals (UN SDGs<sup>\*</sup>) numbers 1 (no poverty), 3 (good health and decent work and economic growth), 4 (quality education), 6 (clean water and sanitation), 8 (decent work and economic growth), 9 (industry, innovation, and infrastructure), 11 (make cities inclusive, safe, resilient, and sustainable, i.e., sustainable cities and communities), 12 (Reasonable consumption and production), 13 (climate action), 14 (life under water), 15 (life on land), and 17 (partnership for the goals).

Users of this book will be able to access the contents of Appendix 9 (*A User's Manual and Programs Screenshots*), the source codes and executables of all example programs,

along with the solutions manual on the downloads tab of the book's CRC Press site: <https://www.crcpress.com/Computer-Modeling-Applications-for-Environmental-Engineers-Second-Edition/Ahmed-Abdel-Magid-Abdel-Magid/p/book/9781498776547>.

**Prof. Dr. CEng. Isam Mohammed Abdel-Magid Sr.**

**Dr. Mohammed Isam Mohammed Abdel-Magid Jr.**  
*Dammam, Doha, 2017*

---

\* UN SDGs include (1) end poverty in all its forms everywhere, that is, no poverty; (2) end hunger, achieve food security and improved nutrition, and promote sustainable agriculture, that is, zero hunger; (3) ensure healthy lives and promote well-being for all at all ages, that is, good health and welfare; (4) ensure inclusive and quality education for all and promote lifelong learning, that is, quality education; (5) gender equality; (6) ensure access to water and sanitation for all, that is, clean water and sanitation; (7) affordable and clean energy; (8) promote inclusive and sustainable economic growth, employment, and decent work for all, that is, decent work and economic growth; (9) industry, innovation, and infrastructure; (10) reduced inequalities; (11) make cities inclusive, safe, resilient, and sustainable, that is, sustainable cities and communities; (12) responsible consumption and production; (13) climate action; (14) life below water; (15) life on land; (16) peace, justice, and strong institutions; (17) partnership for the goals. (Source: United Nations, 2015, <http://www.un.org/sustainabledevelopment/sustainable-development-goals/#>.)



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# Preface and Acknowledgment to First Edition

Environmental engineering is the branch of engineering dealing with activities in the areas of air, water, wastewater, solid waste, radiation, and hazardous materials. Many textbooks have been published in all of these areas; however, most of them lack the fundamental utilization and application of computer programming. Computer programming is now of paramount importance not only for the students but also for the practicing engineers. This book presents computer programs in the air, water, wastewater, and solid waste fields that can be used to evaluate and assist in the design of environmental control systems. This book will be of value to environmental engineers, sanitary engineers, civil engineers, town planners, industrial engineers, agricultural engineers, treatment plant designers, consultation firms, and researchers. The BASIC programming language was selected for its simplicity and used throughout this book.

The broad objectives of this book are to facilitate improving teaching, learning, and research in the application of computer programming based on well-established models, equations, formulae, and procedures used in the environmental engineering field.

The more specified objectives are as follows:

1. To help the reader in the design of treatment and disposal facilities
2. To introduce the basic concepts of computer design
3. To aid in teaching of environmental engineering unit operations and processes design
4. To improve the understanding by students and researchers in the environmental engineering field

This book contains many illustrative figures, schematic diagrams, solved practical examples, design problems, and tutorial homework problems. A wide range and spectrum of environmental engineering subjects are covered. Many computer programs are included, and they cover both the environmental design principles and the structural design parameters using different materials. This book is written in a clear, concise manner in order to enhance understandability. SI units are used throughout this book.

The format of the book is divided mainly into four sections. Section I deals with an introductory material that contains functions and layout of the computer, programming and modeling concepts, and the basic outlines for the BASIC language, which is discussed in [Chapter 1](#).

Section II covers water sources and resources, and is divided into three chapters. [Chapter 2](#) deals with the characteristics of water and wastewater, supported by mathematical equations in order to help programming, water resources and sources, and groundwater flow. Emphasis is laid on water abstraction from wells with the consequent drawdown relationships. [Chapter 3](#) discusses water treatment unit operations/

processes and water storage and distribution. It outlines the design concepts adopted in treatment plant unit operations (e.g., screens, sedimentation and flotation, and aeration) and treatment unit processes (e.g., coagulation and flocculation, filtration, desalinization, and water stabilization). Each treatment process is briefly introduced and emphasis is given to the design of the units using appropriate equations, assumptions, and design tools. [Chapter 4](#) deals with water storage and distribution. It highlights the design of water storage tanks (communa and, dwelling) and the methods used for both urban and rural water distribution.

Section III deals with wastewater engineering, measurement, collection, treatment units, and disposal. It is divided into two main chapters. [Chapter 5](#) focuses on wastewater collection systems for rural areas and urban fringes. Sewerage system design is introduced within the context of this chapter. [Chapter 5](#) also discusses the selected methods for wastewater disposal for rural inhabitants. [Chapter 6](#) is introduced to aid the design of wastewater treatment and disposal units. More stress is placed on in-depth design considerations and parameters for selected systems universally adopted for both rural and urban regions. This chapter emphasizes preliminary, primary, tertiary, and sludge treatment methods and the disposal of treated wastewater to rivers, lakes, or oceans.

Section IV deals with air pollution control technology. It presents a brief overview of the air pollution field followed by the fundamental concepts that are necessary for calculations dealing with air pollution control. It also includes computer programs based on mathematical equations and models related to these fundamental concepts. The most commonly used air pollution control devices for gaseous and particulate pollutants are presented (e.g., absorption, adsorption, and combustion for gaseous contaminants and settling chambers, cyclones, electrostatic precipitators, venturi scrubbers, and baghouse filters for controlling particulate emissions). Each control device or technique is accompanied by a computer program that can aid in the design or evaluation of air pollution control equipment. This section concludes with mathematical models that can be used for the determination of effective stack heights (plume rise) as well as dispersion models that can help estimate the concentrations of air pollutants dispersed in the atmosphere. Computer programs for both stack height calculations and dispersion models are included. All computer programs in the book are included on diskettes.

The authors gratefully acknowledge the support, patience, encouragement, and understanding of their families while they were engaged in the enjoyable task of writing this book.

CRC Press/Lewis editors are acknowledged for their forbearance and patience during all stages of the manuscript preparation, review, editing, and publishing. Particular thanks

and appreciation go to Andrea Demby, Jan Boyle, and Victoria Held for their contributions and outstanding efforts to bring this book into existence.

The authors are indebted to a number of institutions, organizations, and individuals for permission to reproduce and draw published material, and these are acknowledged in the references. Thanks are also extended to the reviewers of the manuscript for their helpful comments and numerous constructive suggestions.

The authors extend their thanks to all who contributed in different ways toward the production of this book. In the air pollution control section, special acknowledgment goes to Asa Raymond for typing, reviewing, and developing the computer programs. Also a special thanks goes to Gary Whittle for his significant contribution in locating pertinent literature in the air pollution control field. Special thanks are also extended to Majid Soud Rashid Al-Khanjri and Taiseer Marhoon Al-Riyami for redrawing and tracing Figures 7.16, 7.17, 7.32, and 7.33.

The authors trust that the book will serve the needs of the profession and provide help and stimulation for candidates to develop their own computer models. They invite and welcome any comments, suggestions, improvements, and constructive criticism from users of the text. Although diligent efforts have been made by the authors, reviewers, editors, and publisher to produce an error-free book, nonetheless, this may not have been achieved and some errors may have escaped the scrutiny of all. The authors would appreciate if any omissions or errors found be brought to their attention so that corrections can be made, which will enhance the usefulness of this publication.

**Isam Mohammed Abdel-Magid**

**Abdel-Wahid Hago Mohammed**

**Donald R. Rowe**

ISBN 1-56670-172-4

© 1997 by CRC Press, Inc.

---

# Authors



**Isam Mohammed Abdel-Magid**, BSc, PDH, DSE, PhD, FSES, CSEC, MSECS, is the professor of water resources and environmental engineering. He received his BSc in civil engineering from the University of Khartoum, Khartoum, Sudan; diploma in hydrology from the University of Padova, Padova, Italy; MSc

in sanitary engineering from Delft University of Technology, Delft, the Netherlands; and PhD in public health engineering from the University of Strathclyde, Glasgow, in 1977, 1978, 1979, and 1982, respectively.

Professor Abdel-Magid authored or coauthored many papers, publications, scientific text and reference books, technical reports, and lecture notes in the areas of water supply; wastewater disposal, reuse, and reclamation; solid waste disposal; water resources and industrial wastes; and slow sand filtration in both English and Arabic. He has participated in several workshops, symposia, seminars, and conferences. He has edited and coedited many conference proceedings and college bulletins.

The Sudan Engineering Society awarded him the prize for the Best Project in Civil Engineering in 1977. In 1986, the Khartoum University Press awarded him Honourly Scarf for Enrichment of Knowledge. Ministry of Irrigation (MoI) awarded him the prize for the Second Best Performance in fourth-year civil engineering. The Sudanese Press Council of Ministry of Information and Culture (MoIC) awarded him the best book of the year. ALECSO awarded him the prize for a book in engineering.

Professor Abdel-Magid has taught environmental engineering subjects and supervised research projects for graduate and postgraduate students at the University of Khartoum, United Arab Emirates University (Al Ain, the United Arab Emirates), Sultan Qaboos University (Muscat, Oman), Sudan University for Science and Technology (Khartoum, Sudan), Juba University (Juba, Sudan), Industrial Research and Consultancy Centre (Khartoum, Sudan), Sudan Academy for Sciences (Khartoum, Sudan), King Faisal University (Dammam, KSA), University of Dammam and Imam Abdulrahman Bin Faisal University (Dammam, KSA). Professor Abdel-Magid acted as an external examiner for different institutions and a founder of some institutions, centers, and refereed journals.

The current address of Prof. Abdel-Magid is as follows: Chair Development and Training Unit of Postgraduate Studies Deanship, Head Proofreading and Revision Department of the Center of Scientific Publications and member of its Council, and Professor of Environmental Engineering and Water Resources at Environmental Engineering Department of the College of Engineering, Imam Abdulrahman Bin Faisal University, Box 1982, Dammam 31451, KSA. Fax: +96638584331, phone: +966530310018, +96633331686, e-mail: iahmed@uod.edu.sa and isam.abdelmagid@gmail.com. Web sites: <http://www/sites.google.com/site/isamabdelmagid/>; <http://www.isam-abdelmagid.net/>. Twitter: [twitter.com/IsamAbdelmagid](https://twitter.com/IsamAbdelmagid). Facebook: <https://www.facebook.com/isam.m.abdelmagid>. Researchgate: [https://www.researchgate.net/profile/Isam\\_Abel-Magid](https://www.researchgate.net/profile/Isam_Abel-Magid). Google scholar: <https://www.facebook.com/isam.m.abdelmagid>. LinkedIn: <https://www.linkedin.com/nhome/?trk=Isam%20Abdel-Magid>. Amazon: <https://author-central.amazon.com/author/isamabdelmagid>.



**Mohammed Isam Mohammed Abdel-Magid**, MBBS, BLS, ALS, MRCP(UK), PgDip in Diabetes, is a graduate of the College of Medicine, the University of Khartoum, Khartoum, Sudan, in 2008. He completed basic training with the Ministry of Health, Khartoum, Sudan, then worked as a physician in the department of internal medicine,

Ribat University Hospital, Khartoum, Sudan, and the Ministry of Health, Sultanate of Oman. He is currently working with the Health Assistance Medical Services, Doha, Qatar.

He completed his higher training with the membership of the Royal Colleges of Physicians of the United Kingdom and accomplished postgraduate diploma in diabetes from the University of South Wales, Wales.

He tutored in problem-based learning teaching sessions in the department of Internal Medicine, Sudan International University, Sudan.

He is a registered practicing physician with the Sudan Medical Council, the Health Authority of Abu-Dhabi, and the Saudi Commission of Health Specialties. He is a full member of the Society for Acute Medicine (Edinburgh, UK), the European Society for Emergency Medicine (Brussels, Belgium), and the European Respiratory Society (H.Q. Lausanne, Switzerland).

He is a peer reviewer of the *Science Journal of Medicine & Clinical Trial* and the *Pan-African Journal of Medical Sciences*.

He is a qualified Linux system administrator, as he obtained the Linux Foundation Certified SysAdmin qualification in 2015. He is an experienced computer programmer with more than 15 years of programming, with a special focus on C/C++ and Visual Basic. He is the maintainer of several Fedora GNU/Linux packages and is an active contributor to the GNU Project (with his GnuDOS and Fontopia packages). He is a member of the Free Software Foundation (Boston, MA). He has designed and packaged several fonts, all of which can be accessed and downloaded from the Open Font Library (<http://fontlibrary.org>) or from the author's Web site (<http://sites.google.com/site/mohammedisam2000>).



# Authors of First Edition



**Isam Mohammed Abdel-Magid Ahmed** received his BSc degree in civil engineering from the University of Khartoum, Khartoum Sudan; diploma in hydrology from the University of Padova, Padova, Italy; MSc in sanitary engineering from Delft University of Technology, Delft, the Netherlands; and PhD in public health engineering from the University of Strathclyde, Glasgow, U.K. in 1977, 1978, 1979, and 1982, respectively.

Abdel-Magid has authored more than 28 publications, 10 scientific textbooks, and numerous technical reports, and lecture notes in the areas dealing with water supply and resources; wastewater treatment, disposal, reclamation, and reuse; industrial wastes; and slow sand filtration, in both English and Arabic. He has participated in several workshops, symposia, seminars, and conferences. He has edited and coedited many conference proceedings and college bulletins.

The Sudan Engineering Society awarded him the prize for the Best Project in civil engineering in 1977. In 1986, the Khartoum University Press awarded him Honourly Scarf and Badge for Enrichment of Knowledge.

Abdel-Magid has taught at the University of Khartoum; United Arab Emirates University, Al Ain, Abu Dhabi, the United Arab Emirates; Sultan Qaboos University, Muscat, Oman; and Omdurman Islamic University, Omdurman, Sudan.



**Abdel-Wahid Hago Mohammed** received his BSc in civil engineering from the University of Khartoum, Khartoum, Sudan in 1976. He received his PhD degree from Glasgow University, Scotland in 1982. Hago has authored more than 13 publications, numerous reports, lecture notes, and more than 95 translated articles. He headed the

Design Office of Newtech Industrial and Engineering Group, Sudan for 3 years, where he contributed in the design of buildings and factories in different parts of that country. In 1986, he acted as a Senior Analyst for Associated Consultants and Partners, Khartoum, Sudan for the design of the ElObied town water distribution system.

Hago has participated in several conferences, and was a member of the organizing committee of the First National Conference on Technology of Buildings organized by the Sudan Engineering Society in 1984. Hago is a member of the

Sudan Engineering Society. He has taught both undergraduate and postgraduate courses at the University of Khartoum and Sultan Qaboos University, Muscat, Oman. He has supervised and cosupervised many MSc and PhD students.



**Donald R. Rowe, PhD\*** is president of D. R. Rowe Engineering Services, Inc., Bowling Green, Kentucky.

He received his MS (1962) and PhD (1965) from the University of Texas at Austin, and his BSc degree (1948) in civil engineering from the University of Saskatchewan, Saskatoon, Canada.

From 1964 to 1969 he was associate professor, Department of Civil Engineering, Tulane University, New Orleans, Louisiana; from 1969 to 1982, he was associate professor, Department of Engineering Technology, Western Kentucky University, Bowling Green, Kentucky; and from 1982 to 1988 he was professor, Department of Civil Engineering, King Saud University, Saudi Arabia. From 1971 to 1993, he was vice president of the Larox Research Corporation, Kentucky and from 1990 to the present, he has been president of his own company.

Rowe has authored or coauthored more than 70 reports and publications in the areas of air pollution, solid waste, water treatment, and wastewater reclamation and reuse. He also coauthored *Environmental Engineering* (McGraw-Hill Publishers) and the *Handbook of Wastewater Reclamation and Reuse* (CRC/Lewis Publishers).

Rowe holds, with coinventors, more than 10 patents on catalytic conversion removal processes of air contaminants in the United States, Canada, Great Britain, and Japan.

In 1980, he was recipient of a Fulbright-Hays Award to Ege University in Izmir, Turkey. Rowe also carried out research on wastewater reclamation and reuse at King Saud University, Riyadh, Saudi Arabia.

From 1984 to the present, he has served on the Peer Review Evaluation Committee for research projects funded by King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia.

\* Professor Donald Richard Rowe, 82, died on May 21, 2009, at his residence. The Late Donald R. Rowe owned and operated D. R. Rowe Engineering Services, Inc., Bowling Green, Kentucky, until his passing. This Vantage, Saskatchewan, Canada native was born in October 12, 1926. He moved from Saskatchewan to Austin, Texas, in the 1950s and married Nora Alice Crittenden in 1954. His passions in life were traveling and his farm situated on Cape Breton Island in Nova Scotia. He was son of the late William Hazen Rowe and Donalda McIntyre.





**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

# List of Symbols and Abbreviations

Symbol	Description	Unit	Symbol	Description	Unit
$A$	Area	m <sup>2</sup>	$C_v$	Volumetric concentration of particles (volume of particles divided by the total volume of the suspension)	Dimensionless
AI	Aggressiveness index		$C_w$	Concentration of pollutant in wastewater	kg/m <sup>3</sup>
Alk	Total alkalinity	mg/L CaCO <sub>3</sub>	$C$	Coefficient, constant	
$A_i$	Area of heat exchanger number $i$	m <sup>2</sup>	$C$	Velocity of the sound in a given medium	m/s
$A_n$	Area of an individual unit	m <sup>2</sup>	ch	Chlorinity	g/kg
$A_t$	Total area	m <sup>2</sup>	$c_g$	Gas concentration in gas phase	g/m <sup>3</sup>
$A$	Constant, coefficient, factor, fraction				
$a'$	Effective sulfide flux coefficient	m/h	$D$	Cumulative passing weight fraction	%
$a_o$	Percentage open area for a clogged screen	%	$D$	Diameter	m
$a_c$	Cell constant		$D$	Disintegrating organic material	
$a_o$	Percentage open area for a clean screen	%	$D$	Cylinder diameter	m
$B$	Width, thickness	m	$D_{60}$	Particle (sieve) size where 60% of the particles are smaller than that size	
BOD	Biochemical oxygen demand	mg/L	$D_{10}$	Particle (sieve) size where 10% of the particles are smaller than that size	
BOD <sub>e</sub>	Effective biochemical oxygen demand	mg/L	Diff	Molecular diffusion coefficient	m <sup>3</sup> /s
BOD <sub>load</sub>	BOD load	kg/m <sup>3</sup>	$D_m$	Mean hydraulic depth	ft
BOD <sub>s</sub>	BOD of standard sewage	gBOD <sub>5</sub> /c/day	DO <sub>c</sub>	Critical oxygen deficit	mg/L
BOD <sub>t</sub>	BOD that has been exerted in time interval $t$	mg/L	$D_1$	Dielectric constant	
BOD <sub>5</sub>	5-Day BOD of wastewater	mg/L	DO <sub>o</sub>	Initial Oxygen deficit at the point of waste discharge, at time $t = 0$	mg/L
$b'$	Constant		DO <sub>t</sub>	Oxygen deficit at time $t$	mg/L
$b$	Slope of the straight line of $t/V$ versus $V$	s/m <sup>6</sup>	DR	Dilution rate	Dimensionless
$C$	Concentration	mg/L	DWF	Dry weather flow	L/day
$C$	Idle resistance factor	Dimensionless	$d$	Depth, diameter	m
$C$	Efficiency factor		$d$	Particle size	
$C$	Total amount percolating within the top layer of soil	mm/year	$d'$ and $n$	Characteristic of the material studied	
$C$	Capacity	tons/h	$d_o$	Maximum particle size	
CA	Sludge age	day	$dP$	Differential change in pressure	Pa
Cd	Constant or shape factor	Dimensionless	du/dy	Velocity gradient (rate of angular deformation, rate of shear)	s <sup>-1</sup>
CON	Conductance	Ohm <sup>-1</sup>	$dV$	Differential change in volume	m <sup>3</sup>
$C'$	Solubility of oxygen at barometric pressure $P$ and given temperature	mg/L	$d_{max}$	Maximum depth	m
$C_D$	Newton's drag coefficient	Dimensionless	$d_p$	The distance between blades or the scroll pitch	m
$C_e$	Concentration in effluent	mg/L, g/m <sup>3</sup>			
$C_f$	Friction coefficient, coefficient of DeChezy	m <sup>0.5</sup> /s	$E$	Coefficient of Eddy diffusion, or turbulent mixing	m <sup>2</sup> /s
$C_m$	Concentration of the pollutant in mixture of river water and wastewater discharge, or concentration of pollutant in the river downstream point of discharge	mg/L	$E$	Evapotranspiration	mm/year
$C_o$	Concentration at time $t = 0$ (initial)	mg/L	EC	Electrical conductivity	Ohm/m
$C_p$	Carbon concentration in the mixture prior to composting, as the percentage of total wet mass of mixture	%	Eff	Efficiency	%
$C_s$	Saturation concentration	mg/L	EM	Electrophoretic mobility	m/s/V/m
$C_s$	Carbon concentration in the sludge, as the percentage of total wet sludge mass	%	$E_b$	Bulk modulus	N/m <sup>2</sup>
$C_r$	Concentration of the same pollutant in river upstream discharge point	mg/L	$E_v$	Rate of evaporation	L/day
$C_r$	Carbon concentration in the refuse, as the percentage of total wet refuse mass	%	$E'$	Activation energy	calories
			$E_e$	Electrochemical equivalent	g/Coulomb
			$e_e$	Expanded bed porosity	Dimensionless
			$e$	Porosity	Dimensionless

(Continued)

Symbol	Description	Unit	Symbol	Description	Unit
$F$	Collection frequency, number of collections per week		$k_p$	Removal rate constant for waste stabilization pond	day <sup>-1</sup>
$F$	Speed factor	Dimensionless	$k_t$	Decay constant for the particular decay reaction	day <sup>-1</sup>
$F$	Volume of reduction		$k'''$	Reaeration constant	day <sup>-1</sup>
Far	Faraday's constant	Ampere*h	$k'$	First-order reaction rate constant	day <sup>-1</sup>
F/M	Food-to-microorganisms ratio	day <sup>-1</sup>	$k_j$	Bacterial die-away rate	day <sup>-1</sup>
Fr	Froude number	Dimensionless	$L$	Length of conveyor belt	ft
$F_s$	Shear force	N	$L$	Length of cylinder	m
$F_1, F_2$	Recirculation factor	Dimensionless	$L_b$	BOD <sub>5</sub> <sup>20</sup> of benthal deposit	g/kg volatile matter
$f$	Factor, function, coefficient		$L_e$	Effluent BOD	mg/L
$f$	Frequency	cycle/s (Hertz)	$L_i$	Influent BOD	mg/L
$G$	Food waste, percentage by weight of total MSW, on a dry basis	%	$L_m$	Maximum daily benthal oxygen demand	g/m <sup>2</sup>
$G$	Velocity gradient	s <sup>-1</sup>	$L_o$	Methane generation potential	volume/mass of waste
$g$	Gravitational acceleration	m/s <sup>2</sup>	$L_o$	BOD remaining at time $t = 0$ (total or ultimate first stage BOD initially present)	mg/L
$H$	Lift	ft	$L_t$	Amount of first-stage BOD remaining in the sample at time $t$	mg/L
$H$	Total energy (total head, energy head)	m	$L_{EP}d$	Daily personal noise exposure level	dB
Hard	Hardness	mg equivalent CaCO <sub>3</sub> /L	$L_{EP}w$	Weekly average of the daily personal noise exposure	dB(A)
HCa	Calcium hardness	mg/L CaCO <sub>3</sub>	$(L_{EP}d)_i$	Daily personal noise exposure of day $i$	dB(A)
HE <sub><math>i</math></sub>	Heat exchanged in effect number $i$	J	$l$	Length, depth	m
H <sub><math>o</math></sub>	Initial resistance for a clean screen	m	$l_e$	Expanded bed depth	m
H <sub><math>s</math></sub>	Resistance for a clogged screen	m of water	$l_e$	Equivalent pipe length	m
[H <sup>+</sup> ]	Concentration of hydrogen ions	mol/L	$M$	Moisture content, percentage (on a wet basis)	%
$h$	Height, depth, thickness	m	$M$	Mass of the material	lb, tons, kg
$h_f$	Friction head	m	MCRT	Mean cell residence time	day
$h_l$	Head loss	m	MLVSS	Mixed liquor suspended solids	mg/L
$h_t$	Height of proposed tank	m	MW	Molecular weight of the gas	g
$I$	Mean rainfall intensity for a duration equal to the time of concentration	mm/h	$M_a$	Moisture in solids as in shredded and screened refuse	%
$I_r$	Average infiltration into the sewer owing to poor joints or pervious material	L/day	$M_d$	Moisture content, percentage (on a dry basis)	%
$i$	Annual interest rate	fraction	$M_i$	Mass of wet waste, placed at time $i$	
$i$	Current	A	$M_p$	Moisture in mixed pile (heap) ready to begin process of composting	%
$i$	Current density	A/cm <sup>2</sup>	$m$	Mass	kg
$j$	Slope, gradient	m/m	$N$	Number	Dimensionless
$K$	Constant, factor, coefficient		$N$	Number of conveyor leads or number of blades wrapped around conveyor hub	
$K$	Coefficient of permeability		$N$	Number of sound sources	
$K$	Hydraulic conductivity of the drainage layer	ft/day	$N$	Number of collection vehicles needed	
$K$	Intrinsic permeability or specific permeability		$N$	Number of working days taken into account during the week	
$K_b$	Bunsen absorption coefficient	g/J	ND	Nondecomposable material	
$K_D$	Distribution coefficient		$N_e$	Effluent bacterial number, number of bacteria	/100 mL
$K_s$	Half-velocity constant	mg/L	$N_i$	Influent bacterial number, number of bacteria	/100 mL
$k$	Coefficient, constant		$N_o$	Number of viable microorganisms of one type at time $t = 0$	/100 mL
$k$	Constant	Dimensionless			
$k$	Exponent	Dimensionless			
$k$	Landfill gas emission constant	Time <sup>-1</sup>			
$k_H$	Henry's constant	g/m <sup>3</sup> *Pa (g/J)			
$k_l$	Loss coefficient				
$k_n$	Kinetic coefficient	Dimensionless			

Symbol	Description	Unit	Symbol	Description	Unit
$n$	Number	Dimensionless	$Q_R$	Recirculated flow	m <sup>3</sup> /s
$n$	Number of installments		$Q_T$	Total gas emission rate from a landfill	volume/time
$n$	Number of years		$OR_{x,y}$	Overall recovery for components $x$ and $y$	
$N$	Roughness factor, Manning and Kutter factor	m <sup>1/6</sup>	$Q_w$	Waste sludge flow	m <sup>3</sup> /day
$N$	Sound intensity level	dB	$q^+$	Charge on the colloid	Coulomb
$N$	Total time periods of waste placement		$q$	Flow that can be applied per unit area	L/m <sup>2</sup> *day
			$q$	Vertical inflow (infiltration)	ft/day
OL	Organic loading rate of the sewage effluent	g/L	$R$	Plastics, percentage by weight of total MSW, on a dry basis	
			$R$	Rotational speed of screw	rpm
$P$	Distance between collection pipes	ft	$R$	Universal gas constant	J/K*mol, L*atm/mol*K, J/kg*K
$P$	Level of sound intensity to be measured	W			
$P$	Number of people served by the sewer	Dimensionless			
$P$	Paper, percentage by weight of total MSW, on a dry basis		Re	Reynolds number	Dimensionless
$P$	Precipitation	mm/year	RES	Resistance of a conductor	W
$P$	Pressure	mmHg, N/m <sup>2</sup> (Pa)	RES <sub>s</sub>	Specific resistance of the suspension	W*cm
			RI	Ryzner index	
$P$	Pressure of sound wave	Pa	$R_m$	Resistance of filter medium	m <sup>-1</sup>
$P$	Pulley friction horsepower		$R_{mr}$	Rate of corrosion	
PA( $t$ )	Time-varying value of A-weighted instantaneous sound pressure	Pa	$R_s$	Specific resistance	W <sup>-1</sup>
$P_{ref}$	Some reference sound pressure	Pa	$R_T$	Total removal	%
PE	Population equivalent	Dimensionless	$R_u$	Recycling ratio	Dimensionless
POW	Power	J/s, W	$R_{x1}$	Recovery of component $x$ in the first output stream (1)	
$P_g$	Partial pressure of the respective gas in the gas phase	Pa, atm	$r$	Coefficient of runoff (can be estimated for different types of soil)	
$P_{osm}$	Osmotic pressure	atm	$r$	Radius	m
pOH	Negative logarithm of the hydroxyl ion concentration		$r_D$	Rate of deoxygenation	
			$r_H$	Hydraulic radius	m
$p_w$	Pressure of saturated water vapor	mmHg, N/m <sup>2</sup>	$r_r$	Rate of reaeration	
$P_o$	A reference level, usually the intensity of a note of the same frequency at the threshold of audibility = intensity of the least audible sound	W	$r_s$	Specific resistance of sludge cake	m/kg
			$r_s'$	Constant	
$P_{ref}$	Some reference sound pressure		$S$	Salinity correction term	
$P_{x1}$	Purity of the first output stream in terms of $x$	%	$S$	Storage in the soil or solid waste	mm/year
			$S$	Speed of belt	ft/min
pAlk	Negative logarithm of total alkalinity	Equivalent CaCO <sub>3</sub> /L	$S$	Total number of customers serviced	
			SA	Sludge age	days
pCa <sup>2+</sup>	Negative logarithm of calcium ion	mol/L	SDI	Sludge density index	g/mL
Pf	Peaking factor	Dimensionless	SLR	Sludge loading rate	day <sup>-1</sup>
pH	Negative logarithm of the hydrogen ion concentration		SRT	Solids retention time	m
			SS	Suspended solids	mg/L
pH <sub>a</sub>	Actual [measured] pH of the water		SVI	Sludge volume index	mL/g
pH <sub>s</sub>	Saturation pH		SWL	sound power level	dB
			SPL	Sound pressure level	dB
			SPLN	Net sound pressure level	dB
$Q$	Rate of flow, fire demand	m <sup>3</sup> /s	SLP <sub>A</sub>	Sound level at distance $D_A$ from the noise source	m
$Q$	Delivery of refuse	m <sup>3</sup> /min			
$Q_a$	Average flow	m <sup>3</sup> /s	SLP <sub>B</sub>	Sound level at distance $D_B$ from the noise source	m
$Q_i$	Total inflow volume during a specified period	m <sup>3</sup> /s			
			SPL <sub>i</sub>	Sound pressure level from source $i$	dB
$Q_{max}$	Maximum flow rate	m <sup>3</sup> /s	$S_c$	Specific capacity of well	m <sup>3</sup> /day*m
$Q_{min}$	Minimum wastewater flow	m <sup>3</sup> /day	$S_u$	Sulfide concentration	mg/L*h
$Q_o$	Total outflow volume during a specified period	m <sup>3</sup> /s	$S_{udis}$	Dissolved sulfide concentration in wastewater	mg/L
$Q_p$	Peak flow	m <sup>3</sup> /s			

(Continued)

Symbol	Description	Unit	Symbol	Description	Unit
$s$	Distance of separation between cylinders (discs)	m	$v_{sco}$	Scour velocity	m/s
sa	Salinity	g/kg	$v_w$	The difference in the rotational velocity between the bowl and the conveyor	rad/s
s.g.	Specific gravity of settling particle	Dimensionless	$v_z$	Updraft velocity	m/s
$s^*$	Growth-limiting substrate concentration in solution	mg/L	$W$	Loading rate	kg/day
$T$	Capacity	tons/h	$W$	Number of workdays per week	
$T$	Temperature	°C, °F, °K	$W$	Organic loading of the trickling filter	g BOD/day
TDS	Total dissolved solids	mg/L	$W$	Power of sound wave	W
TOR	Net torque input	dyne/cm	$W$	Water, percentage by weight, on a dry basis	%
TR	Transmissibility of aquifer	m <sup>3</sup> /day*m	$W$	Weight of SW to be buried	
$T_c$	Temperature correction factor	Dimensionless	$Wt$	Weight	kg
$T_w$	Average trade waste discharge	L/day	$W_w$	Initial (wet) weight of sample	
$T_e$	Duration of the person's personal exposure to noise		$W_d$	Final (dry) weight of the sample	
$T_o$	Length of the working day	h	$W_o$	Some reference sound power	W (pW)
$t$	Time, detention time	days	$w_p$	Wetted perimeter	m
$t_c$	Critical time	days	$X$	Amount borrowed, amount available every year	
$t_{1/2}$	Half-life of the particular nuclide	s, days	$X$	Number of customers a single truck can service per day	
$t_{25}$	Time required for water surface to fall 25 mm	s	$X_T$	Total removal	%
$t_i$	Age of the $i$ th section of waste	Time	$X_s$	Total mass of sludge, wet tons per day	
U	Velocity at free plate surface	LT <sup>-1</sup>	$X_r$	Total mass of refuse, wet tons per day	
UC	Uniformity coefficient		$x_1$	First component emerging of the first output stream (1)	Mass/time
$u_a$	Average velocity	m/s	$x_o$	$x$ Component entering to the binary separator	Mass/time
$u_i$	Heat transfer coefficient for heat exchanger number $i$	J	$x_g$	Mole fraction of gas	Dimensionless
$V$	Volume	m <sup>3</sup>	$x_c$	Critical distance	m
$V$	Volume of refuse between each pitch	m <sup>3</sup>	$x$	Distance, drawdown	m
$V$	Volume of sanitary landfill area		$x$	Percentage of compressed SW volume	%
$V_r$	Volume of a layer of coverage required		$x_a$	Mass of solids	wet tons
VO	Volatile organics		$x_s$	Mass of sludge or other source of water	tons
$V_o$	Original size (initial)		$Y$	Distance from a fixed plate	m
$V_c$	Volume after compaction		$Y$	Installment cost, amount that has to be invested	
VOL	Volumetric organic loading rate	kg/m <sup>3</sup> *s	$Y_{max}$	Maximum saturated depth over the liner	ft
VS	Concentration of volatile solids	kg/m <sup>3</sup>	$Y_t$	Ultimate gas yield	m <sup>3</sup> gas/kg VS added
$V_A$	Volume of material (A)		$y$	Distance, depth, width	m
$V_B$	Volume of material (B)		$Z$	Depth, elevation	m
$V_g$	Volumetric gas production rate (specific yield)	m <sup>3</sup> gas/m <sup>3</sup> digester/day	ZP	Zeta potential	V
$V_s$	Settled volume of sludge in a 1000 mL graduated cylinder in 30 minutes	mL/L, %	$\alpha$	Constant	
$v$	Velocity	m/s	$\alpha$	Inclination of liner from horizontal	°
$v$	Speed of cylinders	rpm	$\beta$	Beta function for a centrifuge	
$v'$	Hindered settling velocity	m/s	$\sigma$	Surface tension	N/m
$v_a$	Approach, actual velocity	m/s	$\phi$	Angle	°
$v_c$	Centerline velocity	m/s	$\phi$	Particle shape factor	Dimensionless
$v_f$	Filtration velocity	m/s	$\phi_{sw}$	Hydrogen sulfide flux to pipe wall	g/m <sup>2</sup> *h
$v_H$	Velocity of horizontal water movement	m/s	$\gamma$	Specific weight	N/m <sup>3</sup>
$v_r$	Rotational velocity of the bowl	rad/s	$\gamma$	Specific weight of water	
$v_s$	Settling rate, displacement velocity, loading rate, overflow rate	m/s	$\tau$	Shear stress	N/m <sup>2</sup>
$v_{sc}$	Self-cleansing velocity	m/s			

Symbol	Description	Unit	Symbol	Description	Unit
$\Sigma$	Parameter related to the characteristics of centrifuge		$\mu$	Dynamic (absolute) viscosity	N*s/m <sup>2</sup>
$\Sigma F$	Sum of all forces acting on fluid contained between two cross sections	N	$\mu$	Dynamic viscosity of water	
$\rho$	Density	kg/m <sup>3</sup>	$\xi$	Ionic strength	
$\rho$	Average density of solid waste and garbage		$\theta$	Temperature coefficient	Dimensionless
$\rho$	Density of the material	g/cm <sup>3</sup>	$\varepsilon$	Roughness of the pipe wall	m
$\rho_o$	Original apparent density		$\lambda_o$	Filtration coefficient	m <sup>-1</sup>
$\rho_c$	Apparent density after compaction		$\lambda$	wave length	m
$\rho_c = \rho_{A+B}$	Bulk density of the mixture of material (A) and material (B)		$\omega$	Impeller rotations	Number of rotations/min
$\rho_A$	Bulk density of material (A)		$\chi$	Constant	
$\rho_B$	Bulk density of material (B)		$\mu_s$	Growth rate of microorganism	day <sup>-1</sup>
$\delta$	Thickness of laminar layer	m	$(\mu_s)_{\max}$	Maximum specific growth rate of microorganisms	day <sup>-1</sup>
$\Psi$	Rate of deposition	m <sup>2</sup> /s	$\nu$	Specific volume of the fluid	m <sup>3</sup> /kg
$\nu$	Kinematic viscosity	m <sup>2</sup> /s	$\Delta T$	Change in temperature	°C
$\zeta$	Eddy viscosity	N*s/m <sup>2</sup>	$\Delta S$	Change in storage volume during specified interval	
			$\Delta P$	Pressure difference	Pa



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# List of Acronyms

Abbreviation	Meaning
ALGOL	algorithmic language
ALU	arithmetic and logic unit
AMD	Advanced Micro Devices, Inc.
BASIC	beginner's all-purpose symbolic instruction code
BIOS	basic input/output system
BMP	bitmapped picture
CD	compact disc
CIL	common intermediate language
CLI	command line interface
COBOL	common business-oriented language
CPU	central processing unit
CRT	cathode ray tube
DVD	digital versatile disc, or digital video disc
ENIAC	electronic numerical integrator and computer
FORTRAN	formula translation
FOSS	free and open source software
FSF	free software foundation
GNU	GNU's not Unix
GPL	GNU general public license
GUI	graphical user interface
IBM	International Business Machines Corporation
IDE	integrated developer environment
IoT	internet of things
JIT	just-in-time translator
JPEG	joint of picture experts group
JVM	Java virtual machine
LAN	local area network

Abbreviation	Meaning
LCD	liquid crystal display
LED	light emitting diodes
MMIO	memory mapped input/output
MS-DOS	microsoft disk operating system
MSDN	microsoft developer network
OCR	optical character recognition
OOP	object-oriented programming
OS	operating system
OSI	open source initiative
PC	personal computer
PDF	portable document format
PNG	portable network graphics
PROLOG	programming logic
QWERTY	The first six letters appearing from the left-hand side of a keyboard
QWERTZ	same as QWERTY, German version
RAM	random access memory
ROM	read-only memory
SQL	structured query language
TUI	text-user interface
USB	universal serial bus
VB.NET	Visual Basic.NET
VBA	VB for applications
VS	Visual Studio
WINE	Windows Emulator

---





**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# 1 Programming Concepts

This chapter presents an introduction to the basic functions, layout, and components of a computer workstation. Computer programming languages are briefly outlined and summarized, delineating their history, types, and uses, with an emphasis being placed on Visual Basic.NET (VB.NET) language as is the language of choice to this book.

The management of water resources, wastewater collection, treatment, and disposal, solid waste collection and ultimate recycling, reuse and disposal as well as air pollution and noise abatement and control involves complex processes. This is due to many interacting parameters, some of which are difficult in presenting a straightforward mathematical model, equation, or formula. Mathematical modeling techniques can be used in predicting the quality and sequence of relationships that can help management in an effort to solve potential environmental problems. A model can be regarded as an assembly of concepts in the form of one or more mathematical equations that approximate the behavior of a natural system or phenomenon (Ji 2008). Models can be divided into simulation, optimization, and computer-aided design models (Chapra 2008). *Simulation models* address the presentation of a mathematical model that simulates a specific situation, with the formulation of a mathematical relationship and solution through a structured and valid process. *Optimization models* use mathematical techniques to achieve a reasonable solution from a range of possibilities. *Computer-aided design models* help in the preparation of design drawings and computation of quantities.

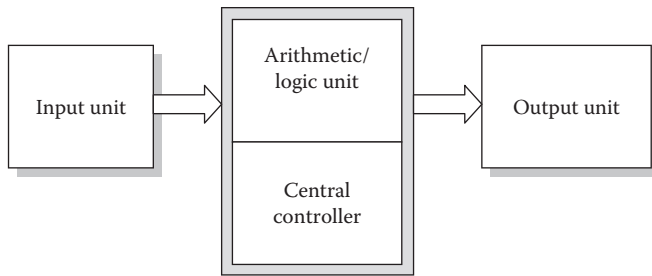
Modeling methods rely on computers for their validation, testing, speediness, and efficient use. The basic concepts of mathematical modeling and formulation of a well-written operative computer program form the central theme of this book. The aim is to train individuals to write, develop, validate, and use their own environmental engineering control programs and simulation models.

In many of the examples tackled in this book, the best *teaching* method was adhered to. This is not always the best *working* or the best method with regards to its real-world *performance*. Sometimes the teaching method is longer, slower, or even more complex than what a production-level model should be, but the authors opted to make all examples, presented throughout the manuscript, as readable and understandable as possible. As such the reader is free to edit any selected examples to be more performance efficient if so he or she desires. A live example is editing controls'

properties in the form `_load()` function of each form. In reality, the programmer (including oneself for that matter) will edit these properties in the design view. This will not only reduce code clutter, but it also helps separate design from code and reduces the work that the program has to do when loading each new instance of the form to memory upon execution. In this book, however, everything is placed inside code blocks to reduce the complexity of the text. As such, the reader is able to follow easily the code and understand exactly what it does without really looking into the form design. Suffice it to know that there are five labels, five textboxes, and one button in the form. Every other detail about these 11 controls is found in one place inside the `form_load()` function. Again, this design choice is not suitable for production-level software as it is solely intended for teaching models like the ones presented in this script. Furthermore, error checking has been dropped in some examples to keep the programs as direct, explicit, clear, and simple as possible.

## 1.1 HOW IT ALL STARTED

Long before computers came to life, machines have been developed to perform certain tasks. These ranged from ancient and simple devices like the Greek Abacus, which was used for counting (around 50 BC), to the Pascaline (a mechanical calculator), which was named after its inventor, *Blaise Pascal*, in the seventeenth century, to help his father do his tax work. It functioned mainly to add numbers. The first real calculator that could do the four basic calculations (addition, subtraction, multiplication, and division) was developed by the Austrian *Antonius Braun* (1727). Later on, in 1822, *Charles Babbage* (1791–1871), known as the father of computers, designed a machine that would do complex calculations and print the results. This was a revolutionary concept in his time. Although he never actually built the machine due to the limitations present in the technology at that time, which was not precise enough to match his designs, the design and functions he set are present in nearly all today's computers (see [Figure 1.1](#)). During World War II, German submarines were receiving encrypted messages from the bases that were coded using a coding machine called the *Enigma*. Around that time, in the U.S. Army, a machine for computing tables for ballistic trajectories, named electronic numerical integrator and computer, was being developed (Murdocca and Heuring 1999).



**FIGURE 1.1** Components of a computing machine.

## 1.2 COMPONENTS OF A COMPUTER

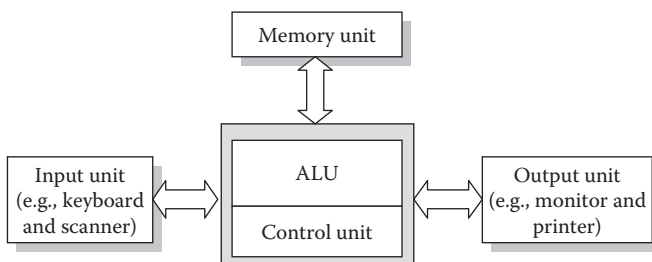
Essentially, a computer is a machine that can perform complex, long, and repetitive sequences of operations at very high speeds (Welsh and Elder 1988). It captures data in the form of instructions or electronic signals, processes it, and then supplies the results in the form of information or electronic signals to control some other device or process (Schneider 2013). Theoretically speaking, conventional computers have five major components (as designed by von Neumann and others, often referred to as *the von Neumann model*) (see Figure 1.2) (Murdocca and Heuring 1999):

1. An input unit
2. An arithmetic and logic unit (ALU)
3. An output unit
4. A memory unit
5. A control unit

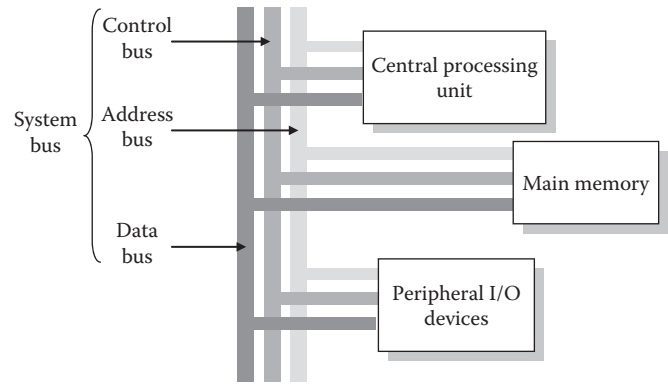
A typical personal computer (PC) includes four hardware components (in analogy to the von Neumann model): a processor (central processing unit [CPU], which contains both the control unit and the ALU), input devices (a keyboard and a mouse), output devices (video display devices such as a monitor or a projector, printers, and other peripherals), and storage media (memory, hard disks, optical media, and flash memory).

The main components of a PC system can be detailed as follows (see Figures 1.2 through 1.6):

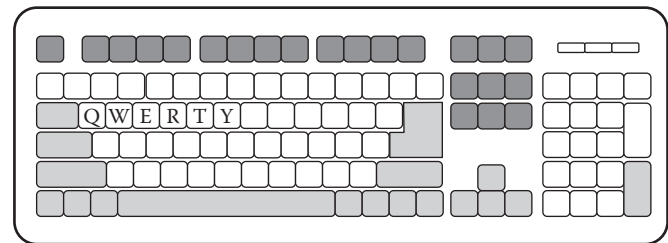
1. *The CPU*: This component represents the brain or the control center of the computer. It carries out the sequence of operations specified by a program, and



**FIGURE 1.2** Components of a computer (von Neumann model).

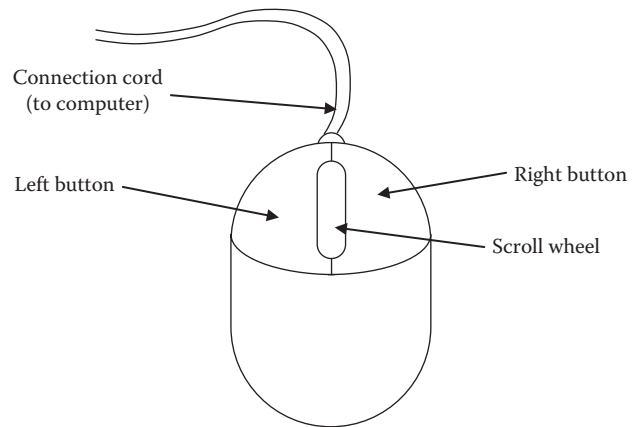


**FIGURE 1.3** The system bus. I/O, input/output.



- = Function keys
- = Meta keys and other special keys
- [ ] = Alphanumeric set

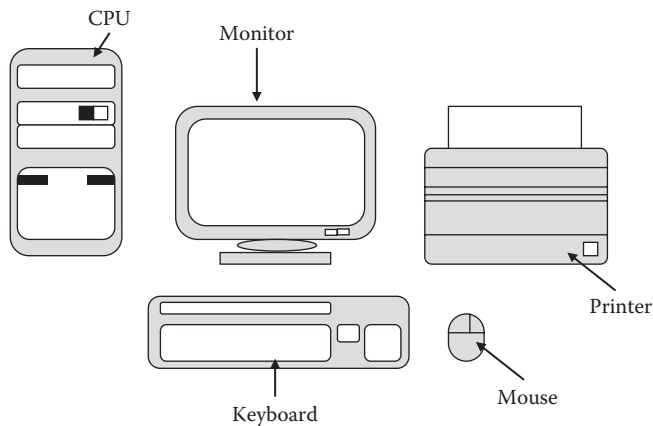
**FIGURE 1.4** QWERTY keyboard layout.



**FIGURE 1.5** Structure of a computer mouse.

it controls all computer activities. The functions of the CPU can be divided as follows (Hager and Wellein 2010):

- a. The control unit controls electronic signals passing through the processor, and it monitors the operation of the whole system. It also controls the flow of code and data through the processor, on system buses, to main memory and other destinations (such as hard disks, Universal Serial Bus [USB] disks, and output devices such as printers).



**FIGURE 1.6** Components of a PC system.

- b. The ALU executes arithmetic and logical operations. These are of paramount importance, as most of the commands performed by the processor involve data moving or operations on data. As the processor's golden currency is time, this must be handled as efficiently as possible. As such, modern processors apply very complex logical operations to determine the best path to take, how to handle data, and what operation to do next, and when to do it.

As the CPU is the maestro of the orchestra, it must have a method of contacting the other components under its control (see [Figure 1.2](#)). It negotiates tasks with other components of the system using a common channel known as the system bus (see [Figure 1.3](#)), which has three components (Murdocca and Heuring 1999):

- *The control bus:* It carries information about the operation being requested, for example, read from memory and write to disk.
  - *The address bus:* It carries the address of the destination (or source) of the data, be it in the main memory or saved on the disk.
  - *The data bus:* It carries the data being read/written.
2. *Input devices:* There are several programmable input devices such as the keyboard, digital pen, joystick, mouse, microphone, and document scanner. These input devices function much the same as human sensory organs in that they provide input from different sources into the computer:
    - a. The keyboard is by far the primary and mostly used input device for entering commands and data. It transfers characters to a special “keyboard buffer” in memory in response to each key event. Key events include key press, key release, and “sticky” or persistent key presses. Each key event corresponds to a special and well-defined “scan code” (Brouwer 2009), which is a special byte number indicating the key that generated

the event and the event itself (press, release, repeat). Most key events generate one to three scan codes. In addition to the scan codes, most keyboards generate special control codes that are used to convey messages to the host computer (e.g., the presence of key input) or to respond to commands from the host computer. A typical PC keyboard has 101 keys (some Windows-specific keyboards have 104). The keys generate scan codes as described previously. The operating system (OS) does the job of transforming each key event to its respective key mapping (e.g., the user pressed the key with “A” printed on it. This might result in “a” printed on the screen, or “A” if the user pressed Shift simultaneously, or has turned Caps Lock on beforehand, or maybe it prints a whole different letter in another language if the user is using international language mapping). The keys on a typical keyboard can be divided into: (1) alphabetical letters and numbers (also called the alphanumeric set), (2) special symbols and shapes, and (3) function and control keys. As each scan code is 1 byte in length, the keyboard driver can generate 256 different scan codes (a byte is 8 bits, thus  $2^8$  different values can be represented in a single byte). This is not an awful lot, but remembering that each key event can generate one to three different scan codes means that all key events (press, release, repeat) can be generated by using multiple scan codes for individual key events. There are different ways of arranging the keys on a keyboard, which is termed the keyboard “layout” (see [Figure 1.4](#)). The most commonly used layout is the QWERTY layout, which corresponds to the first six letters appearing from the left hand side. This layout was adopted from early typewriters. Other layouts exist, such as QWERTZ (a German layout) and others.

- b. The mouse is a positioning device that is used to navigate graphical user interfaces (GUIs) by controlling a cursor (commonly shaped as an arrow with changing sizes, shapes, and colors). It eases user interaction with the system by providing a means of working with files, operating on the disk file system, launching programs, and other functions, by the user clicking on one (or more) of its two to three buttons. Older mice used to have three buttons by default. More current mice dropped the middle button (but can usually be emulated by pressing both buttons concurrently) and substituted it for the scrolling wheel (see [Figure 1.5](#)). Although the mouse is commonly associated with GUI environments, a lot of contemporary OSs provide some (albeit sometimes rudimentary) mouse support in text-based environments (text–user interface; also

called command-line interface). For example, most Linux distributions provide mouse support by default on their virtual terminals.

- c. The digital pen is a type of input device that can be used to convert analog data (like user handwriting or hand drawing) to digital data that can be accepted, processed, and stored by a computer system. A popular use is in sending handwritten articles (by authors or journalists, for example) to the computer, after which it can be run through a special software known as optical character recognition (OCR) software, which can analyze the digital data, perform complex calculations on it, and ultimately convert the raw pixel data it received into distinct language characters, thus recovering the original piece of handwritten article. Text can then be edited by a word processor to add visual effects and can thereafter be printed or emailed.
  - d. Document scanners are used to scan pages and save them in an image format (such as JPEG, PNG, BMP, and others) or in the portable document format (PDF). The data can then be handled as is or processed through the same type of OCR software described previously to retrieve the data contained in it.
3. *Output devices* use different means to display the output of calculations, the results of different computations, whole documents, spreadsheets, or even audio and video.
- a. *The monitor (visual display, display screen):* This output unit permits the display of input information as it is entered and allows the display of the output produced by computer software. Many types of monitors are found on the market, with different vendors, sizes, and capabilities. Current monitors are usually light-emitting diode displays, which use an array of light-emitting diodes to present the pixels on screen. Some other monitors are using the inferior liquid crystal display technique, which takes advantage of the light-modulating capabilities of liquid crystals. The old cathode ray tube displays are rendered extinct by these new technologies. The contents of the display are governed by data stored in a special area of the memory known as the video memory. The video memory is typically built into the monitor itself to allow for fast transfer of images from the memory to the screen proper. The OS usually either writes directly to the video memory or channels the graphics (or text) it wants output on the screen to a special area of the main memory, known as memory mapped input/output (MMIO), which is connected to the main video memory. Writing to this special MMIO area is akin to writing

the main video memory, and changes in one are directly reflected on the other. This technique is not exclusively used by video memory. In fact, most (though not all) modern I/O devices use some sort of MMIO in their communication with the system. Some vendors provide special capabilities with their video cards to enable programs with higher visual needs to be run seamlessly, a feature that is very important in video game programming (Hill and Pilkgtton 1990).

- b. *Printers and plotters:* A printer allows the generation of a permanent record or a hard copy of the output. Printers can be broadly classified according to the following:
  - Users connected to the printer:
    - Personal printer: It is connected to one host computer. It is usually office or home based. It is easy to install, operate, and maintain, and is suitable for personal or small business use.
    - Shared or multiuser printer: It is connected to a network (usually local area network) of users. Typically, it is found in corporate offices and is shared and used simultaneously by different users at the same time. This type must be durable and highly maintained to withstand the workload expected.
  - Physical versus virtual: Physical printers are real-world printers like the one sitting on top of one's own desk. A virtual printer is a piece of software that acts and behaves as if it was a real printer. From the application's point of view, the virtual printer is a printer, and all data that were destined to be printed will be sent to the virtual printer as such. A typical example is the indispensable "generic" PDF printer, which is installed by default on almost all modern OSs.
  - Dimensions of the printed document: Typical, everyday use printers are two dimensional. With the advent of three-dimensional printers (3D), a new era of printing has emerged, and it may not take long for 3D printing to be integrated into our daily life activities and routine.
  - Mechanisms of character formation:
    - Dot matrix printers form individual characters by using a series of dots, one line at a time. Although it is considered legacy technology, dot matrix printers are still used in special situations where low cost, low quality, or large volume printing is required, for example, in cash registers.

- Solid font mechanisms produce fully formed characters via the formation of sharp images. They are to be preferred to dot matrix printers when letter-quality output is needed.
- Method of transfer of data, or print, to page:
  - Impact printers form characters by striking the type against an inked ribbon that presses the image onto the paper. This is obsolete nowadays.
  - Nonimpact printers do not involve physical contact but instead transfer the image to the paper using ink spray, heat, xerography, or laser. They are usually quieter and faster than impact printers. Most modern printers are nonimpact printers.

A plotter allows the sequential drawing of objects rather than using the raster-scan principle adopted for printers. The plotter produces professional quality drawings on different paper sizes and allows the production of final, multiple-colored drawings.

4. *Data storage media*: Storage of programs and data can be made on (fixed) internal hard disks, optical media (including compact discs [CDs] and DVDs), memory sticks (USB or flash drives), and external hard disk stores. Memory is considered storage medium, albeit temporary, as all the information stored therein is lost whenever the electrical current is lost; hence, it is often called “volatile memory.”

Generally, storage media are used to store information, either permanently (like CDs and hard disks) or temporarily (like memory). They may be divided as follows:

- *Floppy disks (diskettes)*: These portable media are made of flexible plastics that can be magnetized. Data are stored on the disks as magnetized spots or concentric tracks on its magnetic coating by the head of the disk drive. They are largely obsolete and are rarely used today as the other technologies have superseded them in terms of efficiency, reliability, storage size, and cost effectiveness.
- *Hard disks (Winchester disks)*: These are made of inflexible metal and are installed and fixed in the computer. They are more reliable with larger storage capacities and faster access times than the floppies. Each computer system usually contains at least one hard disk, which is fixed inside the computer case. This serves as the primary storage medium for the system. Otherwise, disks can be used as a large medium for transferring data between systems or as backup storage for data recovery in case of system failure.

- CDs are made of polycarbonate plastic with a thin layer of aluminum (the reflective surface). Data are stored on the surface of the disc as a series of microscopic indentations; this is read later using a laser beam. These discs can be formatted to store media (e.g., music CDs), data (e.g., software CDs), and video (e.g., movie CDs).
- The primary memory is volatile, that is, when the computer is turned off, the information is lost (also known as the main store, random access memory [RAM], or read/write memory). It is used by the processor as a temporary information storage device. The information can be an input to the system, or it may be generated from internal calculations. The RAM is also used to store and retrieve computer programs during program execution. The basic measurement unit of computer memory is denoted as a byte (corresponds to 8 bits, a bit being the smallest unit of information handled by a computer and either 1 or 0 in the binary number system). Bytes are grouped into kilobytes (1 kB = 1024 or  $2^{10}$  bytes), megabytes (1 MB = 1024 kB), giga bytes (1 GB = 1024 MB), and tera bytes (1 TB = 1024 GB). The immutable memory is called read-only memory (ROM) and contains the basic input/output system (BIOS). The BIOS controls the major I/O devices in the system. ROM is not a volatile memory, it is built into the computer, and this information is retained even when the computer is off.

Figure 1.6 shows the main components of a computer. The inside of the computer contains the main board, power supply, adapter cards, disk drives, processor, RAM, ROM, math coprocessor, and support chips. For effective, trouble-free operation, routine corrective and preventive maintenance should be performed regularly for the computer. Computer failures can be caused by excessive heat, dust buildup, noise interference, powerline problems, corrosion, and the presence of magnetic flux (Andrews 2013).

### 1.3 TYPES OF PROGRAMS

A program is an orderly collection of instructions directed to the computer, instructing it to perform specific tasks. Programs collectively available to the computer to be performed are termed software. This is in contrast with the hardware, which is the solid component of a computer system, described in Section 1.2. Software is needed to direct the computer to perform the required tasks. Hardware without software is just a collection of assembled material. Likewise, software has no purpose to do if there is no hardware to control. Software can be divided into the following categories:



### 1.3.1 OS (SUPERVISOR PROGRAMS)

This is a group of programs that efficiently control the operation of the computer. It is usually the first program to load into memory, immediately after the BIOS finishes inspecting the hardware, checks memory for errors, does some household work, and then looks for a suitable OS to run. The BIOS looks into its list of bootable media (editable through BIOS setup on all modern computers and laptops). For example, if the first bootable medium is the CD ROM, the BIOS looks into the controller to see if there is a CD in it. If there is, it inspects the disk to see if it is bootable. If so, it looks for the OS executable and loads it into a special address of the main memory and handles it control of the system. If the CD is not bootable (or no media is inserted in the drive), the BIOS checks the next bootable medium, typically one of the hard disks, and if it finds an OS installed there, it loads it and passes control to it. The part of the OS that forms the core of the system and is loaded by the BIOS is called the OS *kernel*. The kernel permanently resides in the computer memory throughout the system operation. It continuously supervises and coordinates other programs running on the machine and monitors device activities. Many OSs have been, and will continue to be, developed, such as Mac OS by Apple, Microsoft Windows by Microsoft Corporation, and Unix and its successor Linux by Bell Laboratories.

Linux was originally developed by Linus Torvalds (started as a hobby project back in 1991). Since then, Linux became a pillar of its own. It expanded vigorously, and with many contributions from the GNU project (<http://www.gnu.org>), GNU/Linux has become an important and respected player in today's OS arena. There are many systems built around the Linux kernel, which are called "distributions," or shortly "distros." Such distros include Fedora Linux, SuSE Linux, and Gentoo Linux.

### 1.3.2 UTILITY PROGRAMS (USER PROGRAMS)

These programs are designed by manufacturers or software companies to perform specific tasks. Examples of utility programs include text editors, debugging aids, disk formatters, file copiers, data sorters, and data file mergers. The major difference between user and supervisor programs is that user programs run in user mode, which means they are not granted direct control of the system except through special requests to the kernel. For example, when the user opens a document in his/her word processor, the user program (the word processor) asks the OS (the kernel) to do the operation of opening and reading the document. The process of shielding the user programs and preventing them from doing such important actions on the system is an important concept. Consider what happens if that same word processor had frozen because of a bug in the program logic, or even worse, due to malware or viral infection. If the word processor was given liberty while handling the disk and the files on it, a disaster could have happened when the program is frozen: When a program goes down, it will take the whole system down with it. This was the case in the old Microsoft Disk OS (MS-DOS) era,

when system crash was common, and every user knew the life-saving Ctrl-Alt-Del combination to reboot the system and start anew. In modern OSs however, user programs must ask the kernel to do almost every task that involves interaction with the computer system or its peripherals. As such, when a program freezes for whatever reason, the system remains intact, and the user can simply click the "kill this program" button to get rid of the offending software, leaving the rest of the system out of harms' way.

## 1.4 PROGRAMMING AND LANGUAGES

As computers became more and more complex, so did the functions performed by a computer. Old-fashioned computers were general purpose, whereas most of today's microprocessors serve specific goals, such as driving a user's smartphone, smart watch, or iPad, or even controlling a smart TV or a smart fridge; the interconnection of all these objects together has led to the term "Internet of Things." With the universal explosion of information technology that resulted in virtually each person having at least a personal laptop, a desktop PC at work, and a smartphone in his or her pocket, focus has shifted from the older PC era, when the old-fashioned desktop PC was the center of the programming world, and every effort was made to suit the programming process to ensure compatibility with the specific system vendor's specifications and needs.

Complexities in programs and their goals have led to the development of new programming languages, some of which are specific, sometimes highly specific to the extent that one programming language might be associated with just programming into another program (scripting). As such, there are different ways of looking at the programming languages and grouping them into different categories:

- *Object-oriented versus procedural programming:* Almost every programming language in today's world is object oriented (OOP). In this paradigm, everything is an object: you, your car, your dog, and every other thing in your world. Every object belongs to a class of objects, which defines in general terms what are the common features held by objects of its type. Every class has *members* that define its characteristics (such as color, model, and number of doors a car has) and *methods* that define how the outer world interacts with it (gas pedal to accelerate the car, user does not need to know how it actually works internally). As such, every object contains two things: data (*members* or *fields*) and code (*methods*) that works on this data. This programming model is different from the old way in which programs were written, which is known as procedural programming. In procedural programming, the program is written as one clump of code, which is read and executed by the processor sequentially until the commands are finished. This method of programming is considered retrospective in most today's programming aspects, but it still has

its important applications in selected cases. The user is directed to the references where more information about OOP is provided (Buyya et al. 2009).

- *Event-driven environments*: In an OOP world, everything is event driven; nothing happens on its own. For example, the word processor program will continue staring back at the user unless he or she strikes a key, clicks a mouse button, or touches a touch screen, instructing the program to do some action. As such, every event (key press, mouse click, touch on a touch screen) has an event handler (special part of the program's code that determines how to respond to certain events in specific ways). Contrast this with procedural programming where the only event needed was the user invoking the program's executable from the command line, after which the program will continue running from the top down in a predetermined and well-known path. Each and every time the program is run, it will run in the exact same way. It is worth mentioning that reference concerns the output (or results), and not focuses on the *steps* taken to reach these results. In an event-driven environment, how the program runs exactly depends on the events generated by the user, which are different every time unless the user generates the exact set of events every time he or she runs the program.
- *High-, intermediate-, and low-level programming* (see Figure 1.7) (Dandamudi 2005): As more basic functionality is hidden from the user (termed *encapsulation* [Buyya et al. 2009] under the OOP umbrella), as more the language becomes high level. High-level programmers usually program desktop application software; they do not need to know exactly how the OS or the underlying hardware works. Examples of high-level languages are VB, (Visual) C and C++, Java, structured query language (SQL), and many more. On the contrary, OS programmers need to have a detailed knowledge of how the monitor paints pixels to display images, how the keyboard sends key strokes in the form of scan codes, how the hard disk retrieves a sector from the disk to read part of a file, and so on.

In essence, these programmers need to know how the system behaves under the hood. As such, they need a language that does not have all the ornaments of a high-level language but is actually more close to the understanding of the underlying processor. This is known as the intermediate language (or the *assembly* language, which differs slightly according to the type of the underlying processor architecture). The lowest level of functionality is the bare bones, actual machine language, which is not human readable, and it translates to the letter what the processor does, usually expressed in the form of hexadecimal (letters and numbers) words.

- *General- and special-purpose languages*: Languages used to program desktop software, games, general web applications, and servers are general-purpose languages (most of the commonly used programming languages fall into this category, including VB, C/C++, Java, among others). Special-purpose languages are tailored toward more specific tasks, for example, database handling, program scripting, or programming logic, and artificial intelligence (e.g., SQL for database handling, Visual Basic for Applications, JavaScript for web scripting, and PROLOG [Programming in Logic] for artificial intelligence programs).
- *Compiled and translated languages*: Traditionally, every programming language was compiled. Compilation means transforming a program's source code (usually written in a high-level language code) into a machine-specific low-level (binary) code. The compiler is a program that understands both the high-level and machine languages for the specific platform, something akin to the human language translators in international meetings, and as such, each language has its own compiler, being specific for each and every processor. This is not a good scheme for certain conditions, for example, web scripting. As the web page needs to be fast loading and small in size, having a pre-compiled program is not always the good answer.

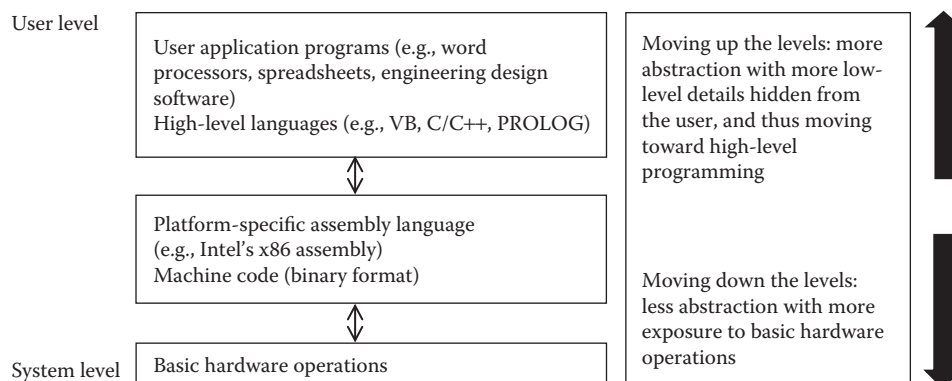


FIGURE 1.7 Different levels of abstraction in the programming model.



Some languages were developed for this very purpose, being resident as a source code, which will be executed by a mediocre program (the translator) at execution time (e.g., JavaScript for web scripting). This scheme involves another step in program's life: writing program source, distributing it, and letting another program (the translator) read it and tell the processor what to do. Some languages took it even further, as there arose the problem of system compatibility. A program written for MS Windows will not work on Unix or Linux. A program written for Intel's x86 architecture will not work on Motorola AMD processor, at least not natively and easily. Therefore, the compatibility problem has two facets: *OS* and *architecture* compatibility. Some languages such as Java solved this problem by inventing the Java virtual machine (JVM). Java source code is compiled into an executable containing intermediate Java binary code (similar, but not identical to, machine code). This executable needs a small program (the JVM) to run it. The JVM is like a processor inside a processor; it reads the binary code and tells the real processor what to do. Thus, programmers need not worry about distributing their original source code (a headache for many corporations and programmers), and users only need to install a small-sized virtual machine that is specific to their systems (e.g., JVM for x86 under Linux, JVM for AMD under Windows). Every program written in Java will work on the fly (provided the JVM is installed on the target system), with the price of some slowness of execution compared to a native machine-coded program.

Writing one's own programs has many benefits over buying and using packaged programs. These advantages include the following:

- Solve specific problems, as the new program will be written specifically to answer the question at hand
- Speed, performance, and interpretation of calculations, compared to manual interpretation of data. Computers are better than humans in doing complex calculations, provided the correct input is given
- Enhancing graphical work, especially with the advent of new hardware that is specially tailored toward graphical design and the continuous improvements in graphical design and modeling software
- Formulating cost-effective design and operation of products. Certainly, a company that invents its own software solutions manages to save more money than the one that relies on presold software
- Computer-aided design and manufacturing is a continuously growing market
- Performing preliminary testing enables preemption of bugs and early correction of problems before deploying software solutions to users
- Controlling manufacturing and assembly operations
- Control and automation of processes
- Effective business and management operations
- Communicating technical results in a reliable and effective way
- Educational and training programs. Virtual tutors are becoming a common practice in many educational institutions and corporate training schedules

Writing one's own software can have some drawbacks and problems also as follows:

- Reinventing the wheel, which is the most fatal mistake a programmer might do. If there is some problem that needs solving, one might better do some search around, looking at some programming forums, running through a specialized website (like the Microsoft Developer Network—<http://msdn.microsoft.com>), or even asking Google or Bing: someone might have had the answer in a more effective and elegant way, so there is no need to waste time and brain power.
- Not knowing where to start. Sitting there with a void mind and an empty source file, staring at the blinking cursor with no notion of what code to write. Therefore, good programming design and preemptive thinking are paramount before grasping the keyboard and starting on writing code.
- Not knowing which language to use. Having a look at the examples of programming languages and their uses as discussed previously helps make this decision.
- Working single handedly. Usually as programs grow bigger, team work becomes more efficient and effective. Sometimes assigning team members with specific tasks help speed up the process and make it more effective and reliable, such as having one software programmer, one debugger, and one testing person.

Once a program has been written, it needs to be verified for its accuracy through the use of specific data. This results in a cycle of checking and correcting the program to obtain required accuracy and specifications. This process is called debugging, and it is carried out by running the program on a computer with suitable test data to eliminate (almost) all compile time, runtime, and logical errors. Figure 1.8 shows the steps to be followed in order to obtain a *virtually* error-free program (we say *virtually* because it is impossible to write a 100% error-free program. Each and every program—except the most trivial ones—will fail at some point in some way under the wrong—or right?—conditions). The process of debugging is simply a trial to reduce these errors to a minimum and to ensure that the program runs as expected under most *normal* conditions.

High-level programming languages are machine independent; after all, this is why they were developed in the first