

**Technical Publications**

# **Problem Solving & Object Oriented Programming**

*A Conceptual Approach*



**Seema Kedar**



**TECHNICAL  
PUBLICATIONS**

*An Up-Thrust for Knowledge*

[technical@vtubooks.com](mailto:technical@vtubooks.com)

[www.vtubooks.com](http://www.vtubooks.com)

# **Problem Solving and Object Oriented Programming**





# PROBLEM SOLVING AND OBJECT ORIENTED PROGRAMMING

**Prof. Seema V. Kedar**

M.E.(Computer)  
HOD & Assistant Professor in I.T. Dept.,  
Rajarshi Shahu College of Engg.,  
Pune



**TECHNICAL<sup>TM</sup>  
PUBLICATIONS**  
*An Up-Thrust for Knowledge*

Pune Nashik Bangalore Chennai Hyderabad  
Ahmedabad Bhopal Lucknow Jaipur Delhi

# PROBLEM SOLVING AND OBJECT ORIENTED PROGRAMMING

ISBN 9789350993651 (Printed Book)



First Edition : January 2014

Reprint : March 2019

© Copyright with Author

All publishing rights (printed and ebook version) reserved with Technical Publications. No part of this book should be reproduced in any form, Electronic, Mechanical, Photocopy or any information storage and retrieval system without prior permission in writing, from Technical Publications, Pune.

Published by :



**TECHNICAL  
PUBLICATIONS**

Amit Residency, Office No.1, 412, Shaniwar Peth, Pune - 411030, M.S. INDIA.  
Ph.: +91-020-24495496/97, Telefax : +91-020-24495497  
Email : technical@vtubooks.com Website : www.vtubooks.com

Printer :

Repro India Ltd.  
50/2 TTC, MIDC,  
Mahape, Navi Mumbai

**Price ₹ 260/-**

The importance of **Problem Solving and Object Oriented Programming** is well known in various engineering fields. Overwhelming response to my books on various subjects inspired me to write this book. The book is structured to cover the key aspects of the subject **Problem Solving and Object Oriented Programming** .

The book uses plain, lucid language to explain fundamentals of this subject. The book provides logical method of explaining various complicated concepts and stepwise methods to explain the important topics. Each chapter is well supported with necessary illustrations, practical examples and solved problems. All the chapters in the book are arranged in a proper sequence that permits each topic to build upon earlier studies. All care has been taken to make students comfortable in understanding the basic concepts of the subject.

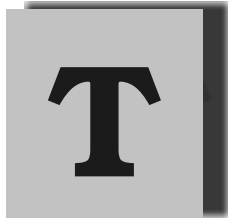
The book not only covers the entire scope of the subject but explains the philosophy of the subject. This makes the understanding of this subject more clear and makes it more interesting. The book will be very useful not only to the students but also to the subject teachers. The students have to omit nothing and possibly have to cover nothing more.

I wish to express my profound thanks to all those who helped in making this book a reality. Much needed moral support and encouragement is provided on numerous occasions by my whole family. I wish to thank the **Publisher** and the entire team of **Technical Publications** who have taken immense pain to get this book in time with quality printing.

Any suggestion for the improvement of the book will be acknowledged and well appreciated.

*Author*  
*Seema Kedar*

*Dedicated to God.*



# Table of Contents

---

<b>Unit - I</b>	<b>Problem Solving Concepts</b>	<b>(1 - 1) to (1 - 28)</b>
1.1	General Problem Solving Concepts .....	1 - 2
1.1.1	Types of Problems.....	1 - 3
1.1.2	Problem Solving with Computers.....	1 - 4
1.1.3	Difficulties with Problem Solving .....	1 - 4
1.1.4	Problem Solving Aspects .....	1 - 5
1.2	Problem Solving Concepts for Computer .....	1 - 6
1.2.1	Constants and Variables.....	1 - 6
1.2.2	Data Types.....	1 - 7
1.2.3	Functions.....	1 - 11
1.2.4	Operators .....	1 - 12
1.2.5	Expressions and Equations.....	1 - 15
1.3	Programming Concepts .....	1 - 17
1.3.1	Communicating with the Computers.....	1 - 17
1.3.2	Organizing the Problem .....	1 - 17
1.3.3	Using the Tools.....	1 - 27
1.3.4	Testing the Solution .....	1 - 27
1.3.5	Coding the Program .....	1 - 27
1.4	Top-down Design .....	1 - 27
	Review Questions.....	1 - 28
<hr/>		
<b>Unit - II</b>	<b>Problem Solving with Logic Structures</b>	<b>(2 - 1) to (2 - 52)</b>
2.1	Programming Structure .....	2 - 2
2.1.1	The Modules and their Functions .....	2 - 4
2.1.2	Cohesion and Coupling.....	2 - 6
2.1.3	Local and Global Variables .....	2 - 7
2.1.4	Parameters.....	2 - 8
2.1.5	Return Values.....	2 - 11
2.1.6	Variable Names and the Data Dictionary.....	2 - 12

2.1.7	Four Logic Structures.....	2 - 13
2.2	Problem Solving with the Sequential Logic Structure.....	2 - 13
2.2.1	The Sequential Logic Structure .....	2 - 13
2.2.2	Solution Development .....	2 - 15
2.3	Problem Solving with Decisions .....	2 - 21
2.3.1	The Decision Logic Structure .....	2 - 21
2.3.2	Multiple IF/THEN/ELSE Instructions.....	2 - 23
2.3.3	Using Straight-Through Logic .....	2 - 24
2.3.4	Using Positive Logic.....	2 - 26
2.3.5	Using Negative Logic .....	2 - 26
2.3.6	Logic Conversion .....	2 - 27
2.3.7	Decision Tables.....	2 - 29
2.4	Problem Solving with Loops and Case Logic Structures.....	2 - 37
2.4.1	Problem Solving with Loops Logic Structures .....	2 - 37
2.4.2	Problem Solving with Case Logic Structure.....	2 - 49
	Review Questions.....	2 - 52

---

**Unit - III      Array and Text Processing      (3 - 1) to (3 - 66)**

3.1	Processing Array.....	3 - 2
3.1.1	One-Dimensional Arrays .....	3 - 2
3.1.2	Multidimensional Arrays.....	3 - 4
3.1.3	Table Look-up Technique .....	3 - 6
3.1.4	Pointer Technique .....	3 - 8
3.2	Array Techniques.....	3 - 8
3.2.1	Array Order Reversal .....	3 - 8
3.2.2	Array Counting or Histogramming .....	3 - 12
3.2.3	Finding the Maximum Number in a Set .....	3 - 14
3.2.4	Partitioning an Array .....	3 - 16
3.2.5	Finding the k <sup>th</sup> Smallest Element.....	3 - 22
3.2.6	Searching an Array for a Range.....	3 - 23
3.3	Text Processing Technique .....	3 - 27
3.3.1	Text Line Length Adjustment .....	3 - 27
3.3.2	Left and Right Justification of Text.....	3 - 32
3.3.3	Keyword Searching in Text.....	3 - 40
3.3.4	Text Line Editing .....	3 - 44
3.4	Pattern Search.....	3 - 50
3.4.1	Linear Pattern Search.....	3 - 50
3.4.2	Sub-linear Pattern Search .....	3 - 57
	Review Questions.....	3 - 64

<b>Unit - IV</b>	<b>Characteristics of OOP</b>	<b>(4 - 1) to (4 - 68)</b>
4.1	Procedure-Oriented Programming .....	4 - 2
4.2	Introduction to Modular Programming .....	4 - 3
4.3	Introduction to Object-Oriented Programming .....	4 - 3
4.4	Introduction to Generic Programming .....	4 - 4
4.5	Limitation of Procedure-Oriented Programming .....	4 - 5
4.6	Need of Object-Oriented Programming .....	4 - 5
4.7	Fundamentals of Object-Oriented Programming .....	4 - 6
4.7.1	Objects .....	4 - 6
4.7.2	Classes .....	4 - 7
4.7.3	Data Members and Methods .....	4 - 7
4.7.4	Data Encapsulation, Data Abstraction and Information Hiding .....	4 - 8
4.7.5	Inheritance .....	4 - 8
4.7.6	Polymorphism .....	4 - 9
4.7.7	Dynamic Binding .....	4 - 10
4.7.8	Messages .....	4 - 10
4.8	C++ : Extensions to C .....	4 - 11
4.8.1	Variable Declaration .....	4 - 13
4.8.2	Global Scope .....	4 - 15
4.8.3	'const' / Literals .....	4 - 16
4.8.3.1	Types of Literals .....	4 - 16
4.8.3.2	Defining Constants .....	4 - 19
4.8.4	Reference Variables .....	4 - 20
4.8.5	Comments .....	4 - 21
4.8.6	Function Prototypes .....	4 - 22
4.8.7	Default Parameters .....	4 - 25
4.8.8	Function Overloading .....	4 - 26
4.8.9	Inline Functions .....	4 - 27
4.8.10	Constant Arguments .....	4 - 29
4.8.11	cin and cout .....	4 - 30
4.8.12	Formatting and I/O Manipulators .....	4 - 32
4.8.13	Dynamic Memory Allocation .....	4 - 34
4.8.14	new and delete Operators .....	4 - 35
4.8.15	Defining a Class .....	4 - 37
4.8.16	Data Members and Methods .....	4 - 38
4.8.17	Object .....	4 - 38
4.8.18	Array of Objects .....	4 - 40
4.8.19	Public, Private and Protected Members .....	4 - 41

4.8.20 this Pointer .....	4 - 46
4.8.21 Constructors.....	4 - 48
4.8.21.1 Characteristics of Constructor Function.....	4 - 48
4.8.21.2 Types of Constructor Function.....	4 - 49
4.8.22 Destructors.....	4 - 53
4.8.23 Friend Function .....	4 - 54
4.8.24 Pointer and Classes .....	4 - 60
4.8.25 Class as ADTs and Code Reuse .....	4 - 61
4.8.26 Static Data Members.....	4 - 64
4.8.27 Static Member Functions .....	4 - 65
Review Questions.....	4 - 67

---

**Unit - V                      Overloading and Inheritance                      (5 - 1) to (5 - 58)**

5.1 Need of Operator Overloading.....	5 - 2
5.2 Overloading the Assignment Operator .....	5 - 2
5.3 Overloading the Binary Operators .....	5 - 4
5.4 Overloading the Unary Operators.....	5 - 6
5.5 Friend Function Operator Overloading.....	5 - 7
5.6 Rules for Overloading Operators .....	5 - 9
5.7 Type Conversion.....	5 - 10
5.8 Introduction to Inheritance.....	5 - 16
5.9 Base and Derived Classes .....	5 - 17
5.10 Types of Inheritance.....	5 - 19
5.10.1 Single Inheritance.....	5 - 19
5.10.2 Multilevel Inheritance .....	5 - 21
5.10.3 Multiple Inheritance.....	5 - 23
5.10.4 Hierarchical Inheritance .....	5 - 25
5.10.5 Hybrid Inheritance and Ambiguity .....	5 - 28
5.11 Virtual Base Class .....	5 - 30
5.12 Constructors Under Inheritance .....	5 - 32
5.13 Destructors Under Inheritance .....	5 - 34
5.14 Virtual Destructors .....	5 - 37
5.15 Member Access Control .....	5 - 38
5.15.1 Accessing Public Data.....	5 - 38
5.15.2 Accessing the Protected Data .....	5 - 41
5.15.3 Accessing the Private Data .....	5 - 44
5.16 Friend Class .....	5 - 46

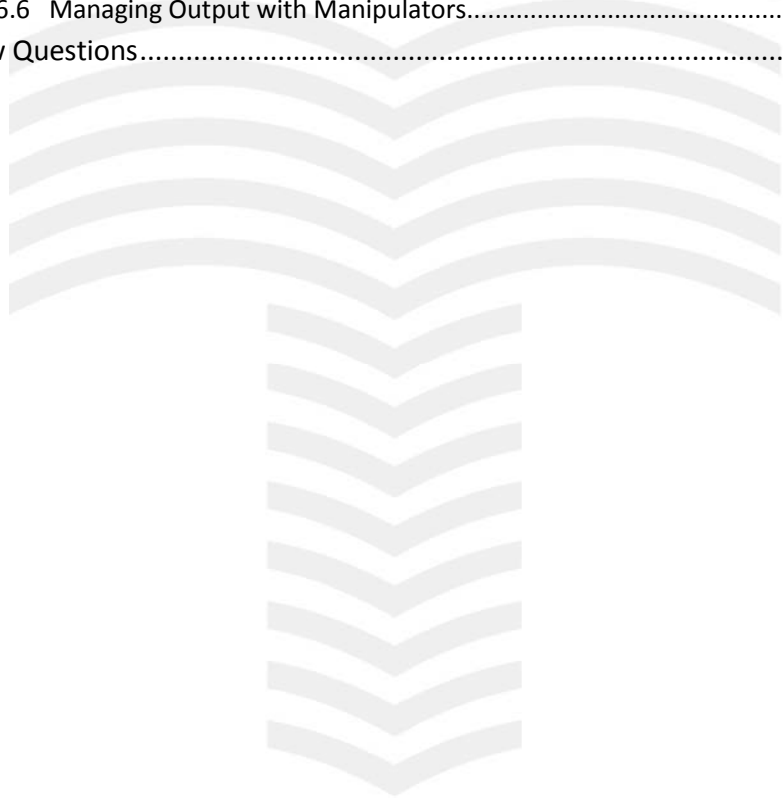
5.17	Introduction to Polymorphism .....	5 - 48
5.17.1	Early Binding.....	5 - 48
5.17.2	Late Binding.....	5 - 48
5.18	Virtual Function .....	5 - 52
5.19	Pure Virtual Functions and Abstract Class .....	5 - 57
5.20	Container Classes .....	5 - 57
	Review Questions.....	5 - 58

---

**Unit - VI          Templates and Exception Handling          (6 - 1) to (6 - 48)**

6.1	Introduction to Templates .....	6 - 2
6.1.1	Class Template.....	6 - 2
6.1.2	Function Template.....	6 - 4
6.1.3	Overloading of Function Templates .....	6 - 5
6.1.4	Member Function Templates .....	6 - 6
6.1.5	Template Arguments.....	6 - 8
6.2	Introduction to Generic Programming.....	6 - 8
6.2.1	Introduction to Standard Template Library .....	6 - 8
6.2.2	Containers .....	6 - 9
6.2.3	Algorithms .....	6 - 13
6.2.4	Iterators.....	6 - 14
6.2.5	Study of Container Classes for Vectors and Stacks .....	6 - 15
6.3	Namespaces .....	6 - 20
6.3.1	Introduction to Namespaces .....	6 - 20
6.3.2	Rules of Namespace .....	6 - 23
6.4	Exception Handling.....	6 - 24
6.4.1	Introduction to Exception Handling .....	6 - 24
6.4.1.1	Exceptions.....	6 - 24
6.4.1.2	Exception Propagation.....	6 - 24
6.4.2	Syntax for Exception Handling Code .....	6 - 24
6.4.2.1	Exception Handling Mechanism.....	6 - 24
6.4.2.2	Functions that Generates Exception .....	6 - 26
6.4.2.3	Throwing Mechanism .....	6 - 28
6.4.2.4	Catching Mechanism.....	6 - 28
6.4.3	Multiple Exceptions.....	6 - 28
6.4.3.1	Multiple Catch Statements .....	6 - 29
6.4.3.2	Catch All Exceptions .....	6 - 30
6.4.3.3	Rethrowing an Exception .....	6 - 31
6.4.4	Exceptions with Arguments.....	6 - 32

6.5	Introduction to RTTI .....	6 - 34
6.6	Managing Console I/O Operations .....	6 - 36
6.6.1	Introduction .....	6 - 36
6.6.2	C++ Streams .....	6 - 36
6.6.3	C++ Stream Classes .....	6 - 37
6.6.4	Unformatted input/output Operations.....	6 - 38
6.6.4.1	Overloaded Operators >> and<< .....	6 - 38
6.6.4.2	put() and get() Functions .....	6 - 39
6.6.4.3	getline() and write() Functions.....	6 - 40
6.6.5	Formatted Console I/O Operations.....	6 - 42
6.6.6	Managing Output with Manipulators.....	6 - 46
	Review Questions.....	6 - 48





# Unit I

## Problem Solving Concepts

### ***Syllabus***

*General problem solving concepts - Types of problems, Problems solving with computers, Difficulties with problem solving, Problem solving aspects, Problem solving concepts for computer- Constants and variables, Data types, Functions, Operators, Expressions and equations, Programming concepts - Communicating with computers, Organizing the problem, Using the tools, Testing the solution, Coding the program, Top down design*

### ***Contents***

- 1.1 General Problem Solving Concepts . . . . . **Dec.-09,10, May-10,11**, . . . . Marks 8
- 1.2 Problem Solving Concepts for Computer . . . . . **May-10,11, Dec.-09**, . . . . Marks 8
- 1.3 Programming Concepts . . . . . **May-10,11, Dec.-09**, . . . . Marks 8
- 1.4 Top-down Design

## 1.1 General Problem Solving Concepts

**Dec.-09,10, May-10,11**

People make decisions everyday to solve problems that affect their lives. If a bad decision is made, time and resources are wasted, so it is important that people should know how to make right decision.

There are six steps to follow to ensure the best decision. These are given below :

**Dec.-09, Marks 6; May-10, Marks 8; May-11, Marks 8**

### 1) Identify the problem

You identify the problem before you start solving it. If you don't know what the problem is, you cannot solve it.

### 2) Understand the problem

You must understand what is involved in the problem before you can continue toward the solution. You cannot solve a problem if you do not know the subject. For example, to solve a problem involving accounting you must know accounting. Thus before start of solution, you understand knowledgebase of the person or machine for whom you are solving the problem.

### 3) Identify alternative ways to solve the problem

Find out different alternatives for solving problem. Alternative solutions must be acceptable ones.

### 4) Select the best way to solve the problem from the list of alternative solutions

In this step, you identify and evaluate the pros and cons of each possible solution before selecting the best one. For doing this, you should have selection criteria for the evaluation, which will serve as the guideline for evaluating each solution.

### 5) List instructions that enable you to solve the problem using the selected solution

List step-by-step instructions from knowledgebase for solving the problem using the selected solution. No instruction can be used unless the individual or the machine can understand it.

### 6) Evaluate the solution

Evaluation of solution means to check its result to see if it is correct, and to see if it satisfies the needs of the person with the problem. If the result is unsatisfactory, restart the process. If any of these six steps are not completed well, the results may be less than desired.

#### ➤ Example

Take the problem of 'what to do this evening'.

# Problem Solving And Object Oriented Programming (A Conceptual Approach)



Publisher : [Technical Publications](#)

ISBN : 9789350993651

Author : [Prof. Seema V. Kedar](#)

Type the URL : <https://www.kopykitab.com/product/32167>



Get this eBook