Peeling Design Patterns

For Beginners and Interviews

By

Prof. Meda Sreenivasa Rao

Narasimha Karumanchi

☼ CONCEPTS ☼ PROBLEMS ☼ INTERVIEW QUESTIONS

Copyright ©2014 by *CareerMonk Publications*

All rights reserved.

Designed by *Narasimha Karumanchi*

Copyright ©2014 CareerMonk Publications. All rights reserved.

All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without written permission from the publisher or author.

Acknowledgements

First and foremost, we would like to thank our *families*, and dear *friends* whose support and encouragement throughout our lives have made it possible for us to build the skill set necessary to succeed.

We would like to express our gratitude to many people who saw us through this book, to all those who provided support, talked things over, read, wrote, offered comments, allowed us to quote their remarks and assisted in the editing, proofreading and design. In particular, we would like to thank the following individuals.

- *Kalyani Tummala*, IIT Kharagpur, Xilinx Pvt. Ltd.
- *Prof. Girish P. Saraph*, Founder, Vegayan Systems Pvt. Ltd.
- *Manoj Patra*, Senior Manager, Microsoft India Pvt. Ltd.
- *A. Vamshi Krishna*, IIT Kanpur, Mentor Graphics Pvt. Ltd.
- *Venkata Ramana Sanaka*, Nokia Pvt. Ltd.
- *Kishore Jinka*, IIT Bombay
- *Vikas Kedia*, IIT Bombay, Google India Pvt. Ltd.
- *Suman Somavarapu*, IIT Bombay, De-Shaw India Pvt. Ltd.
- *Chaganti Siva Rama Krishna Prasad*, Founder, StockMonks Pvt. Ltd.

-Prof. Sreenivasa Rao Meda
Ph. D., Director, School of IT,
JNTU, Hyderabad

-Narasimha Karumanchi
M. Tech, *IIT Bombay*
Founder of *CareerMonk.com*

Preface

Dear Reader,

Please Hold on! I know many people do not read the preface. But I would strongly recommend that you go through the preface of this book at least. The reason for this is that this preface has*something different* to offer.

This book assumes you have basic knowledge about computer science. Main objective of the book is *not* to provide you the *catalog* of design patterns and *design interview* questions. Before writing the book, we set the following *goals*:

- The book be written in *such a way* that readers without any background in software design should be able to understand it *easily* and *completely*.
- The book should present the concepts of design patterns in *simple* and straightforward manner with a *clear – cut* explanation.
- After reading the book, they should be in a position to come up with *better* designs than before and participate in design discussions which happen in their *daily* office work.
- The book should provide enough *realtime* examples so that readers get better understanding of the design patterns and also useful for the interviews. We mean, the book should cover *design interview* questions.
- After reading the book, the programmers should be in a position to try for software architect positions.

Design patterns were introduced to programming practices at the end of 1980*s* as a result of dissatisfaction with software's state of affairs. In those days, few means of abstraction (algorithms and data structures) suited well for procedural/functional programming. They were *not* well connected with object-oriented programming.

The introduction of design patterns marks a turning point in the history of software design. In 1995 we have seen the publication of a catalogue (*Gamma*, *Helm*, *Johnson* & *Vlissides*, 1995) with *twenty-three* design patterns written by four experienced object-oriented designers. The catalogue, which came to be known as the *Gang of Four* (*GoF*) catalogue, was a super success.

Design patterns *help* novices (and *beginners*) to avoid common *mistakes* and encourage experienced programmers to build better software.

This book is an enthusiastic celebration of design patterns, especially those which are given by *GoF*. In the recept past, most of the object-oriented architectures are built with *full* of design patterns.

In this book, *we* explained the design patterns with simple examples which helps *students* and *instructors* in understanding them easily. At the end of book, he gave common *design interview questions* which helps job seekers to perform better in their interviews.

A thorough understanding of the design patterns, processes, and methods in this book will likewise help you develop better software applications and infrastructure. If you want thorough coverage of the key patterns then read this book. We have learned much from it and I am confident that you will too.

As a job seeker if you read complete book with good understanding, we are sure you will challenge the interviewers and that is the objective of this book.

This book is very much useful for the students of engineering degree and masters during their academic preparations. All the chapters of this book contain theory and their related problems. If you read as a student preparing for competition exams, content of this book covers all the required topics in full details.

It is *recommended* that, at least *one complete* reading of this book is required to get full understanding of all the topics. In the *subsequent* readings, you can directly go to any chapter and refer. Even though, enough readings were given for correcting the errors, due to human tendency there could be some minor typos in the book. If any such typos found, they will be updated at www.*CareerMonk*.com. We request you to constantly monitor this site for any corrections, new problems and solutions. Also, please provide your valuable suggestions at: *Info@CareerMonk.com*.

Wish you all the best. We are sure that you will find this book useful.

-Prof. Sreenivasa Rao Meda
Ph. D., Director, School of IT,
JNTU, Hyderabad

-Narasimha Karumanchi
M. Tech, *IIT Bombay*
Founder of *CareerMonk.com*

Table of Contents

1. **Introduction** ---------- **11**

1.1 What Is This Book About? ---------- 11

1.2 Brief History Of Design Patterns ---------- 12

1.3 Should I Take This Book? ---------- 12

1.4 Is It Useful For Interviews? ---------- 12

1.5 How To Use this book? ---------- 13

1.6 Organization Of Chapters ---------- 13

1.7 Source Code Disclaimer ---------- 14

1.8 Tools Used For Book ---------- 14

2. **UML Basics** ---------- **15**

2.1 What Is UML? ---------- 15

2.2 Why UML? ---------- 15

2.3 UML Notations ---------- 16

2.4 Object Oriented Concepts ---------- 16

2.5 OO Analysis and Design (OOAD) ---------- 18

2.6 UML Building Blocks and Notations ---------- 19

2.7 Things ---------- 19

2.8 Relationship ---------- 26

2.9 UML Diagrams ---------- 30

3. **Design Patterns Introduction** ---------- **40**

3.1 What Are Design Patterns? ---------- 40

3.2 Brief History Of Design Patterns ---------- 40

3.3 Why Design Patterns? ---------- 40

3.4 Categories Of Design Patterns ---------- 41

3.5 What To Observe For A Design Pattern? ---------- 42

3.6 Using Patterns To Gain Experience ---------- 42

3.7 Can We Use Design Patterns Always? ---- 43
3.8 Design Patterns vs. Frameworks ---- 43
4. **Creational Patterns ---- 45**
4.1 Creational Design Patterns ---- 45
4.2 Categories Of Creational Design Patterns ---- 45
4.3 Factory Method Design Pattern ---- 46
4.4 Abstract Factory Design Pattern ---- 50
4.5 Builder Design Pattern ---- 54
4.6 Singleton Design Pattern ---- 63
4.7 Prototype Design Pattern ---- 68
5. **Structural Patterns ---- 74**
5.1 Structural Design Patterns ---- 74
5.2 Categories Of Structural Design Patterns ---- 74
5.3 Adapter Design Pattern ---- 75
5.4 Bridge Design Pattern ---- 81
5.5 Composite Design Pattern ---- 85
5.6 Decorator Design Pattern ---- 91
5.7 Facade Design Pattern ---- 98
5.8 Proxy Design Pattern ---- 104
5.9 Flyweight pattern ---- 108
6. **Behavioral Patterns ---- 115**
6.1 Behavioral Design Patterns ---- 115
6.2 Categories Of Behavioral Design Patterns ---- 115
6.3 Chain of Responsibility Design Pattern ---- 116
6.4 Command Design Pattern ---- 122
6.5 Interpreter Design Pattern ---- 128
6.6 Iterator Design Pattern ---- 132
6.7 Mediator Design Pattern ---- 139
6.8 Memento Design Pattern ---- 146

6.9 Observer Design Pattern ---------- 150
6.10 State Design Pattern ---------- 155
6.11 Strategy Design Pattern ---------- 158
6.12 Template Method Design Pattern ---------- 162
6.13 Visitor Design Pattern ---------- 165
7. **Glossary and Tips ---------- 170**
7.1 What Are Design Patterns? ---------- 170
7.2 How To Use Design Patterns?---------- 170
7.3 Why Design Patterns?---------- 171
7.4 What To Observe For A Design Pattern?---------- 171
7.5 Using Patterns To Gain Experience---------- 171
7.6 How To Choose A Design Pattern? ---------- 172
7.7 Can We Use Design Patterns Always?---------- 172
7.8 Categories Of Design Patterns ---------- 173
7.9 Creational Design Patterns---------- 174
7.10 Structural Design Patterns ---------- 174
7.11 Behavioral Design Patterns ---------- 176
7.12 What Are Antipatterns? ---------- 177
7.13 Refactoring ---------- 177
7.14 Design Patterns vs. Frameworks---------- 177
7.15 Tips ---------- 178
8. **Design Interview Questions ---------- 180**
8.1 Design Interview Questions---------- 180
8.2 Sample Design Questions For Practice---------- 262
9. **Miscellaneous Concepts---------- 264**
9.1 Java Interview Questions---------- 264

PEELING DESIGN PATTERNS

(FOR BEGINNERS AND INTERVIEWS)

Other Titles by *Narasimha Karumanchi*

- Data Structures and Algorithms Made Easy (C/C++)
- Data Structures and Algorithms Made Easy in Java
- Data Structures and Algorithms for GATE
- Coding Interview Questions
- IT Interview Questions
- Elements of Computer Networking