# The Object-Oriented Thought Process

 **Fifth Edition**

**Copyright © 2019 Pearson Education, Inc.**

ISBN-10: 0135181968
ISBN-13: 9780135181966

**Warning and Disclaimer**

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD or programs accompanying it.

When reviewing corrections, always check the print number of your book. Corrections are made to printed books with each subsequent printing.

*First Printing: April 2019*

**Corrections for July 6, 2021**

|  |  |  |
| --- | --- | --- |
| **Pg** | **Error** | **Correction** |
| 22-23 | public class IntSquare {//private attributeprivate int squareValue;//public interfacepublic intgetSquare (int value) { SquareValue = calculateSquare(value);return squareValue;}// private implementationprivate intcalculateSquare (int value) {return value\*value;}} | public class IntSquare { //private attribute private int squareValue; //public interface public in getSquare(int value) { squareValue = calculateSquate(value); return squareValue; } // private implementation private in calculateSquare (int value) { return value\*value; }} |
| 68 | //Set the Name of the Cabbiepublic void setName(String iName) {Name = iName;}//Get the Name of the Companypublic static string getName() { return Name;}//Get the Name of the Cabbiepublic static String getCompanyName() {return companyName;} | //Set the Name of the Cabbiepublic void setName(string iName) { Name = iName;}//Get the Name of the Cabbiepublic string getName() { return Name;}//Get the Name of the Companypublic static String getCompanyName() { return companyName;} |

**Corrections for February 10, 2020**

|  |  |  |
| --- | --- | --- |
| **Pg** | **Error** | **Correction** |
| 63 | Delete “OO languages such as Java and .NET do not allow operator overloading.” | Modify next paragraph to “Although some languages may not allow programmers to overload operators directly, many languages do implement operator overloading as a feature (a common example is the plus sign for concatenation). When utilizing operator overloading, take care to document your logic so that people maintaining the code (perhaps even a future you) understand what is happening.” |

This errata sheet is intended to provide updated technical information. Spelling and grammar misprints are updated during the reprint process, but are not listed on this errata sheet.