COM2044
OBJECT ORIENTED PROGRAMMING

Lecturer: Begüm MUTLU BİLGE, PhD
begummutlubilge+com2044@gmail.com

Objectives

Understanding Basic Concepts: Gain a solid understanding of fundamental OOP concepts such as classes, objects, inheritance, encapsulation, and polymorphism.

Practical Implementation Skills: Develop practical skills in writing Java code to define classes, create objects, and implement OOP principles in solving programming problems.

Java Syntax and Language Features: Learn the syntax and key language features of Java, including data types, control structures, methods, and exception handling.

Objectives

Understanding Objects: Learn what objects are and how they represent real-world entities in code. Objects have attributes (data) and behaviors (functions/methods) that interact with each other.

Creating Classes: Discover how to create blueprints for objects called classes. Classes define the attributes and behaviors that objects will have.

Objectives

Encapsulation: Understand encapsulation, which involves bundling data and methods within a class to protect data from unauthorized access and modification. Think of it as packaging data and functions together, like a gift box.

Inheritance: Explore how classes can inherit attributes and behaviors from other classes. This promotes code reuse and helps in organizing related classes into hierarchies.

Polymorphism: Learn about polymorphism, where objects of different classes can be treated as objects of a common superclass. This allows for flexibility in designing and using classes.

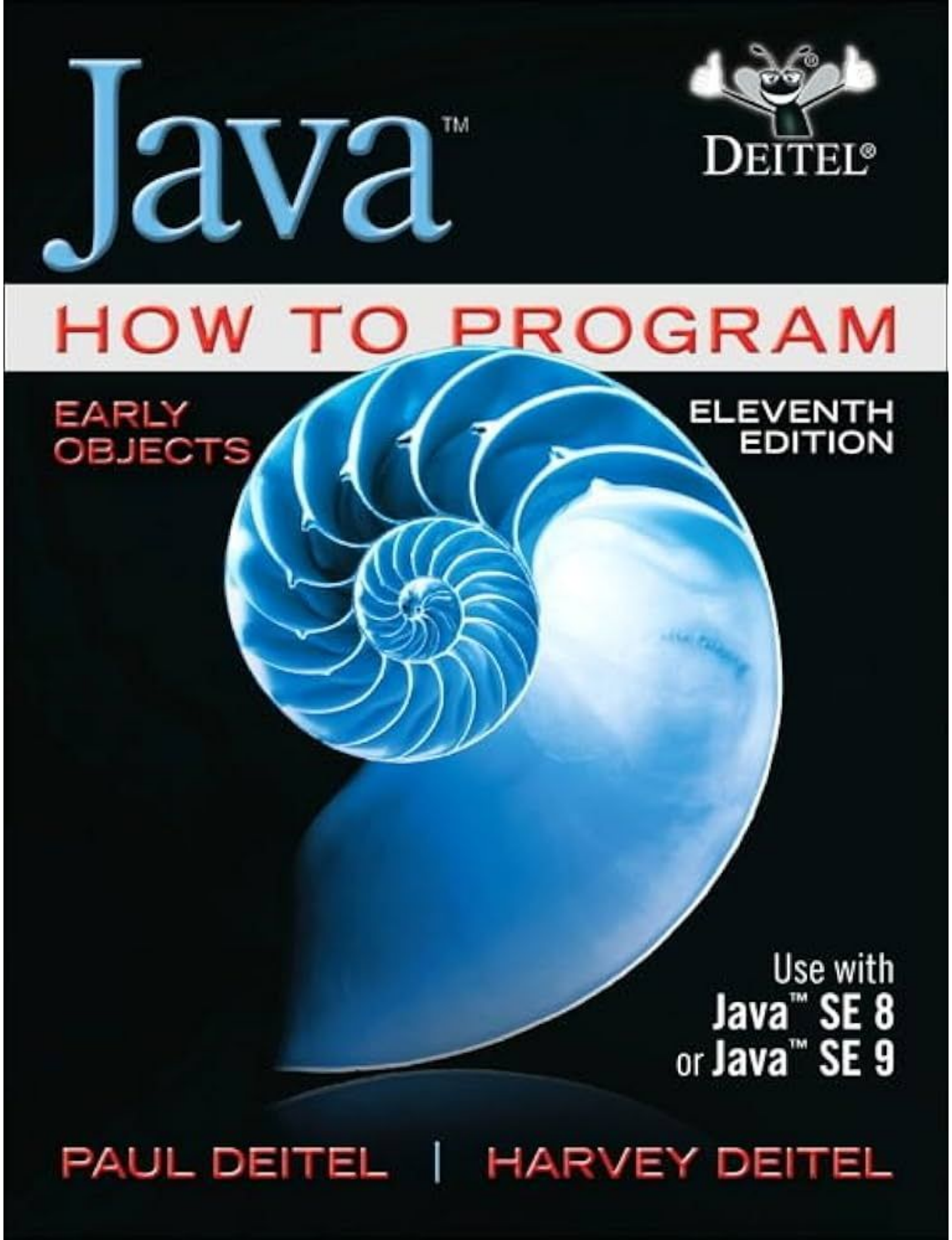
Objectives

Basic Problem-Solving: Practice breaking down problems into smaller, more manageable parts, and solving them using objects and classes. Start with simple exercises to apply OOP concepts in practical scenarios.

Designing Simple Systems: Begin designing small systems using OOP principles. This involves identifying objects, their attributes, and behaviors, and how they interact with each other to achieve a goal.

Understanding the Role of OOP in Software Development: Gain insight into how OOP fits into the broader context of software development. Learn about its importance in creating maintainable, scalable, and reusable code.

Textbook



Java™

DEITEL®

HOW TO PROGRAM

EARLY OBJECTS

ELEVENTH EDITION

Use with Java™ SE 8 or Java™ SE 9

PAUL DEITEL | HARVEY DEITEL

Overview

1. Introduction to Computers, the Internet and Java
2. Introduction to Java Applications; Input/Output and Operators
3. Introduction to Classes, Objects, Methods and Strings
4. Control Statements: Part 1; Assignment, ++ and -- Operators
5. Control Statements: Part 2; Logical Operators
6. Methods: A Deeper Look
7. Arrays and ArrayLists
8. Classes and Objects: A Deeper Look
9. Object-Oriented Programming: Inheritance
10. Object-Oriented Programming: Polymorphism and Interfaces

Overview

1. Introduction to Computers, the Internet and Java
2. Introduction to Java Applications; Input/Output and Operators
3. Introduction to Classes, Objects, Methods and Strings
4. Control Statements: Part 1; Assignment, ++ and -- Operators
5. Control Statements: Part 2; Logical Operators
6. Methods: A Deeper Look
7. Arrays and ArrayLists
8. Classes and Objects: A Deeper Look
9. Object-Oriented Programming: Inheritance
10. Object-Oriented Programming: Polymorphism and Interfaces
11. Exception Handling: A Deeper Look
12. JavaFX Graphical User Interfaces: Part 1 [Laboratory Lecture]
13. JavaFX GUI: Part 2 [Laboratory Lecture]

Overview

14. Strings, Characters and Regular Expressions
15. Files, Input/Output Streams, NIO and XML Serialization
16. Generic Collections
17. Lambdas and Streams
18. Recursion
19. Searching, Sorting and Big O
20. Generic Classes and Methods: A Deeper Look
21. Custom Generic Data Structures
22. JavaFX Graphics and Multimedia
23. Concurrency
24. Accessing Databases with JDBC
25. Introduction to JShell: Java 9’s REPL

Assessment Plan (Estimated)

	Count	**Weighting (%)**
Midterm Exam	1	15
Laboratory Assignment	10	13
Project (multidisciplinary teamwork)	1	5
Popup quiz / Assignment	1-2	2
Attendance (lectures & labs)	14	5
Final Exam	1	60
Percent of in-term studies (%)		**40**
Percentage of final exam & project to total score (%)		**60**

Introduction to Computers, the Internet and Java

1

Man is still the most extraordinary computer of all.
—John F. Kennedy

Good design is good business.
—Thomas J. Watson, Founder of IBM

Objectives

In this chapter you'll:

- Learn about exciting recent developments in the computer field.
- Learn computer hardware, software and networking basics.
- Understand the data hierarchy.
- Understand the different types of programming languages.
- Understand the importance of Java and other leading programming languages.
- Understand object-oriented programming basics.
- Learn the importance of the Internet and the web.
- Learn a typical Java program-development environment.
- Test-drive a Java application.
- Learn some key recent software technologies.
- See how to keep up-to-date with information technologies.

2 Introduction to Java Applications; Input/Output and Operators

What's in a name?
That which we call a rose
By any other name would
smell as sweet.
—William Shakespeare

The chief merit of language
is clearness.
—Galen

One person can make a
difference and every person
should try.
—John F. Kennedy

Objectives

In this chapter you'll:

- Write simple Java applications.
- Use input and output statements.
- Learn about Java's primitive types.
- Understand basic memory concepts.
- Use arithmetic operators.
- Learn the precedence of arithmetic operators.
- Write decision-making statements.
- Use relational and equality operators.

Introduction to Classes, Objects, Methods and Strings

3

Your public servants serve you right.
—Adlai E. Stevenson

Nothing can have value without being an object of utility.
—Karl Marx

Objectives

In this chapter you'll learn:

- How to declare a class and use it to create an object.
- How to implement a class's behaviors as methods.
- How to implement a class's attributes as instance variables.
- How to call an object's methods to make them perform their tasks.
- What local variables of a method are and how they differ from instance variables.
- What primitive types and reference types are.
- How to use a constructor to initialize an object's data.
- How to represent and use numbers containing decima points.

Control Statements: Part 1; Assignment, ++ and -- Operators

4

Let's all move one place on.
—Lewis Carroll

How many apples fell on Newton's head before he took the hint!
—Robert Frost

Objectives

In this chapter you'll:

- Learn basic problem-solving techniques.
- Develop algorithms through the process of top-down, stepwise refinement.
- Use the `if` and `if`...`else` selection statements to choose between alternative actions.
- Use the `while` repetition statement to execute statements in a program repeatedly.
- Use counter-controlled repetition and sentinel-controlled repetition.
- Use the compound assignment operator, and the increment and decrement operators.
- Learn about the portability of primitive data types.

5

Control Statements: Part 2; Logical Operators

The wheel is come full circle.
—William Shakespeare

All the evolution we know of proceeds from the vague to the definite.
—Charles Sanders Peirce

Objectives

In this chapter you'll:

- Learn the essentials of counter-controlled repetition.
- Use the `for` and `do...while` repetition statements to execute statements in a program repeatedly.
- Understand multiple selection using the `switch` selection statement.
- Use the `break` and `continue` program control statements to alter the flow of control.
- Use the logical operators to form complex conditional expressions in control statements.

6 Methods: A Deeper Look

Form ever follows function.
—Louis Henri Sullivan

E pluribus unum.
(One composed of many.)
—Virgil

O! call back yesterday, bid time return.
—William Shakespeare

Answer me in one word.
—William Shakespeare

There is a point at which methods devour themselves.
—Frantz Fanon

Objectives

In this chapter you'll learn:

- How `static` methods and fields are associated with classes rather than objects.
- How the method-call/return mechanism is supported by the method-call stack.
- About argument promotion and casting.
- How packages group related classes.
- How to use secure random-number generation to implement game-playing applications.
- How the visibility of declarations is limited to specific regions of programs.
- What method overloading is and how to create overloaded methods.

Arrays and ArrayLists

7

Begin at the beginning, … and go on till you come to the end: then stop.
—Lewis Carroll

To go beyond is as wrong as to fall short.
—Confucius

Objectives

In this chapter you'll:

- Learn what arrays are.
- Use arrays to store data in and retrieve data from lists and tables of values.
- Declare arrays, initialize arrays and refer to individual elements of arrays.
- Iterate through arrays with the enhanced `for` statement.
- Pass arrays to methods.
- Declare and manipulate multidimensional arrays.
- Use variable-length argument lists.
- Read command-line arguments into a program.
- Build an object-oriented instructor gradebook class.
- Perform common array manipulations with the methods of class `Arrays`.
- Use class `ArrayList` to manipulate a dynamically resizable arraylike data structure.

Classes and Objects: A Deeper Look

8

Is it a world to hide virtues in?
—William Shakespeare

But what, to serve
our private ends,
Forbids the cheating
of our friends?
—Charles Churchill

This above all: to thine own self
be true.
—William Shakespeare.

Objectives

In this chapter you'll:

- Use the `throw` statement to indicate that a problem has occurred.
- Use keyword `this` in a constructor to call another constructor in the same class.
- Use `static` variables and methods.
- Import `static` members of a class.
- Use the `enum` type to create sets of constants with unique identifiers.
- Declare `enum` constants with parameters.
- Use `BigDecimal` for precise monetary calculations.

9 Object-Oriented Programming: Inheritance

Say not you know another entirely,
till you have divided an inheritance with him.
—Johann Kasper Lavater

This method is to define as the number of a class the class of all classes similar to the given class.
—Bertrand Russell

Objectives

In this chapter you'll:

- Understand inheritance and how to use it to develop new classes based on existing classes.
- Learn the notions of superclasses and subclasses and the relationship between them.
- Use keyword `extends` to create a class that inherits attributes and behaviors from another class.
- Use access modifier `protected` in a superclass to give subclass methods access to these superclass members.
- Access superclass members with `super` from a subclass.
- Learn how constructors are used in inheritance hierarchies.
- Learn about the methods of class `Object`, the direct or indirect superclass of all classes.

Object-Oriented Programming: Polymorphism and Interfaces

10

General propositions do not decide concrete cases.
—Oliver Wendell Holmes

A philosopher of imposing stature doesn't think in a vacuum. Even his most abstract ideas are, to some extent, conditioned by what is or is not known in the time when he lives.
—Alfred North Whitehead

Objectives

In this chapter you'll:

- Learn the concept of polymorphism.
- Use overridden methods to effect polymorphism.
- Distinguish between abstract and concrete classes.
- Declare abstract methods to create abstract classes.
- Learn how polymorphism makes systems extensible and maintainable.
- Determine an object's type at execution time.
- Declare and implement interfaces, and become familiar with the Java SE 8 interface enhancements.

Lectures

Lectures

- Monday at 13:30-16:30
- Lecturer: Dr. Begüm MUTLU BİLGE

Laboratory

- Monday at 10:30-11:30 (Section I)
- Monday at 11:30-12:30 (Section II)

Course Assistant

Arş. Gör. Talha Burak ÖZKÖK
talhabozkok+com2044@gmail.com

GOOGLE CLASSROOM

Students should enroll the COM2044 classroom by using the following class code:

ldw64fz

Assignment: Chapter Summary Report

Topic: Introduction to Computers, the Internet, and Java

Instructions:

Read the chapter *Introduction to Computers, the Internet, and Java* from your textbook. Based on your reading, write a **comprehensive report** summarizing the key concepts covered in the chapter.

Your report must:

- **Structure**: Include first-order subsections that logically organize the content.
- **Length**: Be **8 to 10 pages** long.
- **Formatting**: Use the **IEEE Manuscript Template** for document formatting.
- **Tools**: You may use **MS Word or LaTeX** for writing. If using LaTeX, **Overleaf is recommended** for ease of collaboration and formatting.
- **Presentation**: Ensure clarity, logical flow, and proper articulation of concepts.
- **Content**: Demonstrate a clear understanding of the fundamental topics introduced in the chapter.

Submission Guidelines:

- Upload your completed report to **Google Classroom** by the specified deadline.
- Ensure that your report adheres to IEEE formatting standards before submission.

Failure to meet the required structure and formatting may result in a deduction of marks.

If you have any questions, please reach out before the deadline.

COM2044
OBJECT ORIENTED PROGRAMMING

Lecturer: Begüm MUTLU BİLGE, PhD
begummutlubilge+com2044@gmail.com