JAVA PROGRAMMING
LAB MANUAL

Editors
Asadullah Shah
Zeeshan Bhatti

Department of
Kulliyyah of Information & Communication Technology
International Islamic University Malaysia

January 2011

IIUM Press
TABLE OF CONTENTS

Dedication v
Preface xi
Acknowledgement xiii

Chapter 1: Introduction to Java 15
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 2: Compiling and Executing Java Code 21
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 3: Introduction to Data Types in Java 27
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 4: Arithmetic and Relational Operators 33
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 5: Relational and Logical Operators 39
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 6: Bitwise Operators and Type Casting 45
  - Asadullah Shaikh
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 7: Conditional Statements 51
  - Asadullah Shaikh
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 8: If-Else Statements 55
  - Asadullah Shaikh
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 9: The If-Else-If Statements 61
  - Asadullah Shaikh
  - Zeeshan Bhatti
  - Asadullah Shah
Chapter 10: Switch Statement
- Asadullah Shaikh
- Zeeshan Bhatti
- Asadullah Shah

Chapter 11: Loops
- Agha Javed Phatan
- Zeeshan Bhatti
- Asadullah Shah

Chapter 12: Do-While Loops
- Agha Javed Phatan
- Zeeshan Bhatti
- Asadullah Shah

Chapter 13: For Loops
- Agha Javed Phatan
- Zeeshan Bhatti
- Asadullah Shah

Chapter 14: Loop Pitfalls and Exercise
- Agha Javed Phatan
- Zeeshan Bhatti
- Asadullah Shah

Chapter 15: Nested Loops
- Agha Javed Phatan
- Zeeshan Bhatti
- Asadullah Shah

Chapter 16: Arrays in Java
- Waseem Javaid Soomro
- Zeeshan Bhatti
- Asadullah Shah

Chapter 17: Two Dimensional Arrays
- Waseem Javaid Soomro
- Zeeshan Bhatti
- Asadullah Shah

Chapter 18: Strings
- Waseem Javaid Soomro
- Zeeshan Bhatti
- Asadullah Shah
Chapter 19: Methods of String Class  
- Waseem Javaid Soomro  
- Zeeshan Bhatti  
- Asadullah Shah  

Chapter 20: Object and Classes  
- Waseem Javaid Soomro  
- Zeeshan Bhatti  
- Asadullah Shah  

Chapter 21: Constructors  
- Dil Nawaz Hakro  
- Zeeshan Bhatti  
- Asadullah Shah  

Chapter 22: User defined Methods  
- Dil Nawaz Hakro  
- Zeeshan Bhatti  
- Asadullah Shah  

Chapter 23: Arguments Passing  
- Dil Nawaz Hakro  
- Zeeshan Bhatti  
- Asadullah Shah  

Chapter 24: Return Statement  
- Dil Nawaz Hakro  
- Zeeshan Bhatti  
- Asadullah Shah  

Chapter 25: Working with Classes  
- Dil Nawaz Hakro  
- Zeeshan Bhatti  
- Asadullah Shah  

Chapter 26: Multiple Classes  
- Muniba Shaikh  
- Zeeshan Bhatti  
- Asadullah Shah  

Chapter 27: Method Overloading and Overriding  
- Muniba Shaikh  
- Zeeshan Bhatti  
- Asadullah Shah
Chapter 28: Modifiers for class data 191
  - Muniba Shaikh
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 29: Inner Classes 199
  - Muniba Shaikh
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 30: Comparison between different class types 207
  - Muniba Shaikh
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 31: Exception Handling 215
  - Kamran Khowaja
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 32: Throw & Throws Keyword 223
  - Kamran Khowaja
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 33: Local Block and Static Block 231
  - Kamran Khowaja
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 34: Abstract Classes 237
  - Dini Oktarina Dwi
  - Zeeshan Bhatti
  - Asadullah Shah

Chapter 35: Interfaces 245
  - Dini Oktarina Dwi
  - Zeeshan Bhatti
  - Asadullah Shah
Chapter 27

Method Overloading and Overriding

Muniba Shaikh, Zeeshan Bhatti, Asadullah Shah

Abstract

Method Overloading and Overriding are two key principles of Object Oriented Language where multiple methods having same name can coexist in a class. The use and principles of these types of methods is discussed in this experiment with examples for easy understanding of students.

27.1 Method Overloading

Methods of the same class that have the same name but different numbers or types of parameters are called overloaded methods. Two methods in the same class may have the same name, too, but then they must differ in the number or types of parameters. Then the compiler Figures out which one to call. These are called “overloaded methods.”

- Methods in different classes may have the same name.
- Use overloaded methods when they perform similar tasks for example:
  
  ```java
  public void myCalcMethod (int x, int y)  { ... }
  public void myCalcMethod (double x, double y)  { ... }
  public void myCalcMethod (Point p)  { ... }
  public Fraction add (int n)  { ... }
  public Fraction add (Fraction other)  { ... }
  ```

- The compiler treats overloaded methods as completely different methods.
- The return type alone is not sufficient for distinguishing between overloaded methods.