Lecture 1: The History and Evolution Of Java

1.1 Java's Lineage

- 1.1.1 The Birth of Modern Programming: C
- 1.1.2 C++: The Next Step
- 1.1.3 The Stage is Set for Java

1.2 How Java Changed the Internet

- 1.2.1 Java Applets
- 1.2.2 Security
- 1.2.3 Portability

1.3 Servlets: Java on the Server Side

1.4 The Java Buzzwords

- 1.4.1 Object-Oriented
- 1.4.2 Robust
- 1.4.3 Multithreaded
- 1.4.4 Interpreted and High Performance
- 1.4.5 Distributed
- 1.4.6 Dynamic

Lecture 2: An Overview of Java

2.1 Object-Oriented Programming

- 2.1.1 Two Paradigms
- 2.1.2 Abstraction
- 2.1.3 The Three OOP Principles

2.2 A First Simple Program

- 2.2.1 Entering the Program
- 2.2.2 Compiling the Program
- 2.2.3 A Closer Look at the First Sample Program
- 2.2.4 A Second Short Program

2.3 Lexical Issues

- 2.3.1 Whitespace

- 2.3.2 Identifiers
- 2.3.3 Literals
- 2.3.4 Comments
- 2.3.5 Separators
- 2.3.6 The Java Keywords

Lecture 3: Data Types, Variables, and Arrays

3.1 The Primitive Types

- 3.1.1 Integers: `byte`, `short`, `int`, `long`
- 3.1.2 Floating-Point Types: `float`, `double`
- 3.1.3 Characters
- 3.1.4 Booleans

3.2 A Closer Look at Literals

- 3.2.1 Integer Literals
- 3.2.2 Floating-Point Literals
- 3.2.3 Boolean Literals
- 3.2.4 Character Literals
- 3.2.5 String Literals

3.3 Variables

- 3.3.1 Declaring a Variable
- 3.3.2 Dynamic Initialization
- 3.3.3 The Scope and Lifetime of Variables

3.4 Type Conversion and Casting

- 3.4.1 Java’s Automatic Conversions
- 3.4.2 Casting Incompatible Types

3.5 Automatic Type Promotions in Expressions

- 3.5.1 The Type Promotion Rules

3.6 Arrays

- 3.6.1 One-Dimensional Arrays
- 3.6.2 Multi-Dimensional Arrays

Lecture 4: Operators

4.1 Arithmetic Operators

- 4.1.1 The Basic Arithmetic Operators
- 4.1.2 The Modulus Operator
- 4.1.3 Arithmetic Compound Assignment Operators
- 4.1.4 Increment and Decrement

4.2 The Bitwise Operators

- 4.2.1 The Bitwise Logical Operators
- 4.2.2 The Left Shift
- 4.2.3 The Right Shift
- 4.2.4 The Unsigned Right Shift
- 4.2.5 Bitwise Operator Compound Assignments

4.3 Relational Operators

4.4 Boolean Logical Operators

- 4.4.1 Short-Circuit Logical Operators

4.5 The Assignment Operators

4.6 Ternary Operators

4.7 Operator Precedence

- 4.7.1 Using Parentheses

Lecture 5: Control Statements

5.1 Java's Selection Statements

- 5.1.1 `if`, `else`, `else if`
- 5.1.2 `switch`

5.2 Iteration Statements

- 5.2.1 `while`
- 5.2.2 `do-while`
- 5.2.3 `for`

- 5.2.4 For-Each Version of the `for` Loop
- 5.2.5 Nested Loops
- 5.2.6 Infinite Loops

5.3 Jump Statements

- 5.3.1 Using `break`
- 5.3.2 Using `continue`
- 5.3.3 `return`

Lecture 6: Introducing Classes

6.1 Class Fundamentals

- 6.1.1 The General Form of a Class
- 6.1.2 A Simple Class

6.2 Declaring Objects

- 6.2.1 A Closer Look at `new`
- 6.2.2 Assigning Object Reference Variables

6.3 Introducing Methods

- 6.3.1 Adding a Method in the Box Class
- 6.3.2 Returning a Value
- 6.3.3 Adding a Method that Takes Parameters

6.4 Constructors

- 6.4.1 Parameterized Constructors
- 6.4.2 The `this` Keyword: Instance Variable Hiding

6.5 Garbage Collection

- 6.5.1 The `finalize()` Method

6.6 A Stack Class

Lecture 7: A Closer Look at Methods and Classes

7.1 Overloading Methods

- 7.1.1 Overloading Constructors

7.2 Using Objects as Parameters

7.3 A Closer Look at Argument Passing

7.4 Returning Objects

7.5 Recursion

7.6 Introducing Access Control

7.7 Understanding `static`

7.8 Introducing `final`

7.9 Nested and Inner Classes

7.10 Exploring the `String` Class

7.11 Using Command-Line Arguments

7.12 Varargs

- 7.12.1 Variable-Length Arguments
- 7.12.2 Overloading Varargs Methods
- 7.12.3 Varargs and Ambiguity

Lecture 8: Inheritance

8.1 Inheritance Basics

- 8.1.1 Member Access and Inheritance
- 8.1.2 A More Practical Example
- 8.1.3 A Superclass Variable Can Reference a Subclass Object

8.2 Using `super`

- 8.2.1 Using `super` to Call Superclass Constructors
- 8.2.2 A Second Use for `super`

8.3 Creating a Multi-Level Hierarchy

8.4 When Constructors Are Called

8.5 Method Overriding

- 8.5.1 Dynamic Method Dispatch
- 8.5.2 Why Overridden Methods?
- 8.5.3 Applying Method Overriding

8.6 Using Abstract Classes

8.7 Using `final` with Inheritance

- 8.7.1 Using `final` to Prevent Overriding
- 8.7.2 Using `final` to Prevent Inheritance

8.8 The Object Class

Lecture 9: Packages and Interfaces

9.1 Packages

- 9.1.1 Defining a Package
- 9.1.2 Finding Packages and Class Path
- 9.1.3 A Short Package Example

9.2 Access Protection

- 9.2.1 An Access Example

9.3 Importing Packages

9.4 Interfaces

- 9.4.1 Defining an Interface
- 9.4.2 Implementing Interfaces
- 9.4.3 Nested Interface
- 9.4.4 Applying Interfaces
- 9.4.5 Variables in Interfaces
- 9.4.6 Interfaces Can Be Extended

Lecture 10: Exception Handling

10.1 Exception Handling Fundamentals

10.2 Exception Types

10.3 Uncaught Exceptions

10.4 Using `try` and `catch`

10.5 Multiple Catch Clauses

10.6 Nested `try` Statements

10.7 `throw`

10.8 `throws`

10.9 `finally`

10.10 Java Built-in Exceptions

10.11 Creating Your Own Exception Subclasses

10.12 Chained Exceptions

10.13 Using Exceptions

Lecture 11: Multithreaded Programming

11.1 The Java Thread Model

11.2 Thread Priorities

11.3 Synchronization

11.4 Messaging

11.5 The `Thread` Class and the `Runnable` Interface

11.6 The Main Thread