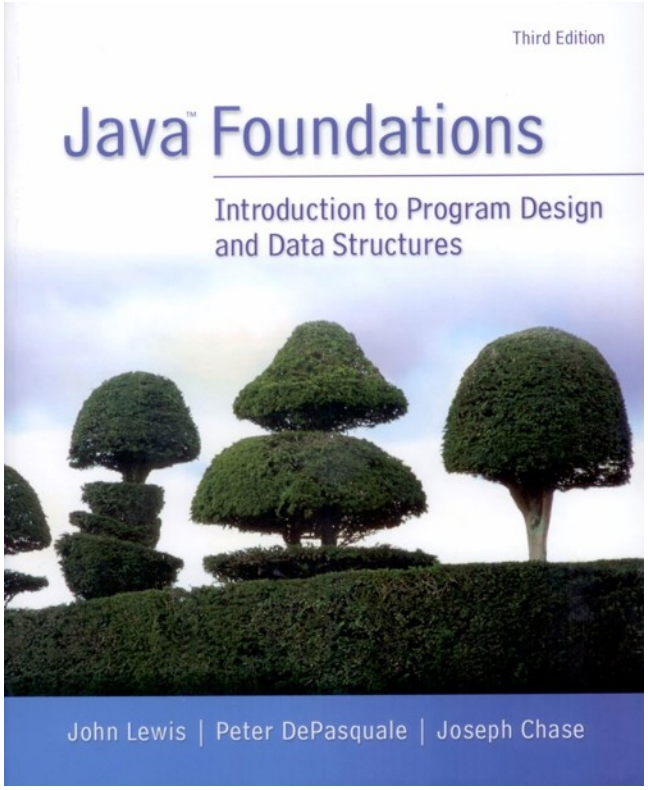


Chapter 1

Introduction

Objectives

- Learn fundamental computer programming concepts and vocabulary
- Introduction to the Java programming language
- Familiarity with programming development activities
- Basic understanding of object oriented programming concepts

Basic Programming Concepts and Definitions

The Java Programming Language

A computer is made up of hardware and software

hardware – the physical, tangible pieces that support the computing effort

program – a series of instructions that the hardware executes one after another

Programs are sometimes called ***applications***

software – consists of programs and the data those programs use

The Java Programming Language

A ***programming language*** specifies the words and symbols that we can use to write a program

A programming language employs a set of rules that dictate how the words and symbols can be put together to form valid ***program statements***

The **Java** programming language was created by Sun Microsystems, Inc.

It was introduced in 1995 and its popularity grew quickly

A Java Program

In the Java programming language

- a program is made up of one or more ***classes***
- a class contains one or more ***methods***
- a method contains program ***statements***

These terms will be explored in detail throughout the course

A Java application always contains a method called `main`

Lincoln.java

```
//********************************************************************
//  Lincoln.java       Java Foundations
//
//  Demonstrates the basic structure of a Java application.
//********************************************************************

public class Lincoln
{
   //-----------------------------------------------------------------
   //  Prints a presidential quote.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      System.out.println ("A quote by Abraham Lincoln:");

      System.out.println ("Whatever you are, be a good one.");
   }
}
```

Java Program Structure

```
//  comments about the class
public class MyProgram
{


}
```

class header

class body

Comments can be placed almost anywhere

Java Program Structure

```
//  comments about the class
public class MyProgram
{
   //  comments about the method
   public static void main (String[] args)
   {

   }

}
```

method body

method header

Comments

Comments in a program are called ***inline documentation***

They should be included to explain the purpose of the program and describe processing steps

They do not affect how a program works

Java comments can take three forms:

```
// this comment runs to the end of the line

/*  this comment runs to the terminating
    symbol, even across line breaks        */

/** this is a javadoc comment   */
```

Identifiers

Identifiers are the words a programmer uses in a program

- can be made up of letters, digits, the underscore character (_), and the dollar sign
- cannot begin with a digit

Java is ***case sensitive*** - `Total`, `total`, and `TOTAL` are different identifiers

By convention, programmers use different case styles for different types of identifiers, such as

- ***title case*** for class names - `Lincoln`
- ***upper case*** for constants - `MAXIMUM`

Identifiers

Sometimes we choose identifiers ourselves when writing a program (such as `Lincoln`)

Sometimes we are using another programmer's code, so we use the identifiers that he or she chose (such as `println`)

Often we use special identifiers called ***reserved words*** that already have a predefined meaning in the language

A reserved word cannot be used in any other way

Reserved Words

The Java reserved words

`abstract`	`else`	`interface`	`switch`
`assert`	`enum`	`long`	`synchronized`
`boolean`	`extends`	`native`	`this`
`break`	`false`	`new`	`throw`
`byte`	`final`	`null`	`throws`
`case`	`finally`	`package`	`transient`
`catch`	`float`	`private`	`true`
`char`	`for`	`protected`	`try`
`class`	`goto`	`public`	`void`
`const`	`if`	`return`	`volatile`
`continue`	`implements`	`short`	`while`
`default`	`import`	`static`	
`do`	`instanceof`	`strictfp`	
`double`	`int`	`super`	

White Space

Spaces, blank lines, and tabs are called ***white space***

- White space is used to separate words and symbols in a program
- Extra white space is ignored

Programs should be formatted to enhance readability, using consistent indentation

Lincoln2.java

```
//********************************************************************
//  Lincoln2.java       Java Foundations
//
//  Demonstrates a poorly formatted, though valid, program.
//********************************************************************

public class Lincoln2{public static void main(String[]args){
System.out.println("A quote by Abraham Lincoln:");
System.out.println("Whatever you are, be a good one.");}}
```

Lincoln3.java

```
//********************************************************************
//  Lincoln3.java       Java Foundations
//
//  Demonstrates another valid program that is poorly formatted.
//********************************************************************

          public       class
     Lincoln3
   {
                  public
   static
        void
  main
        (
String
            []
    args                        )
  {
  System.out.println        (
"A quote by Abraham Lincoln:"          )
  ;        System.out.println
            (
       "Whatever you are, be a good one."
      )
  ;
}
          }
```

Software Development Process and Tools

Program Development

The mechanics of developing a program include several activities

- writing the program in a specific programming language (such as Java)
- translating the program into a form that the computer can execute
- investigating and fixing various types of errors that can occur

Software tools can be used to help with all parts of this process

Language Levels

There are four programming language levels

- machine language
- assembly language
- high-level language
- fourth-generation language

Each type of CPU has its own specific ***machine language***

The other levels were created to make it easier for a human being to read and write programs

Programming Languages

Each type of CPU executes only a particular ***machine language***

A program must be translated into machine language before it can be executed

A ***compiler*** is a software tool which translates ***source code*** into a specific target language

Often, that target language is the machine language for a particular CPU type

The Java approach is somewhat different

Java Translation

The Java compiler translates Java source code into a special representation called ***bytecode***

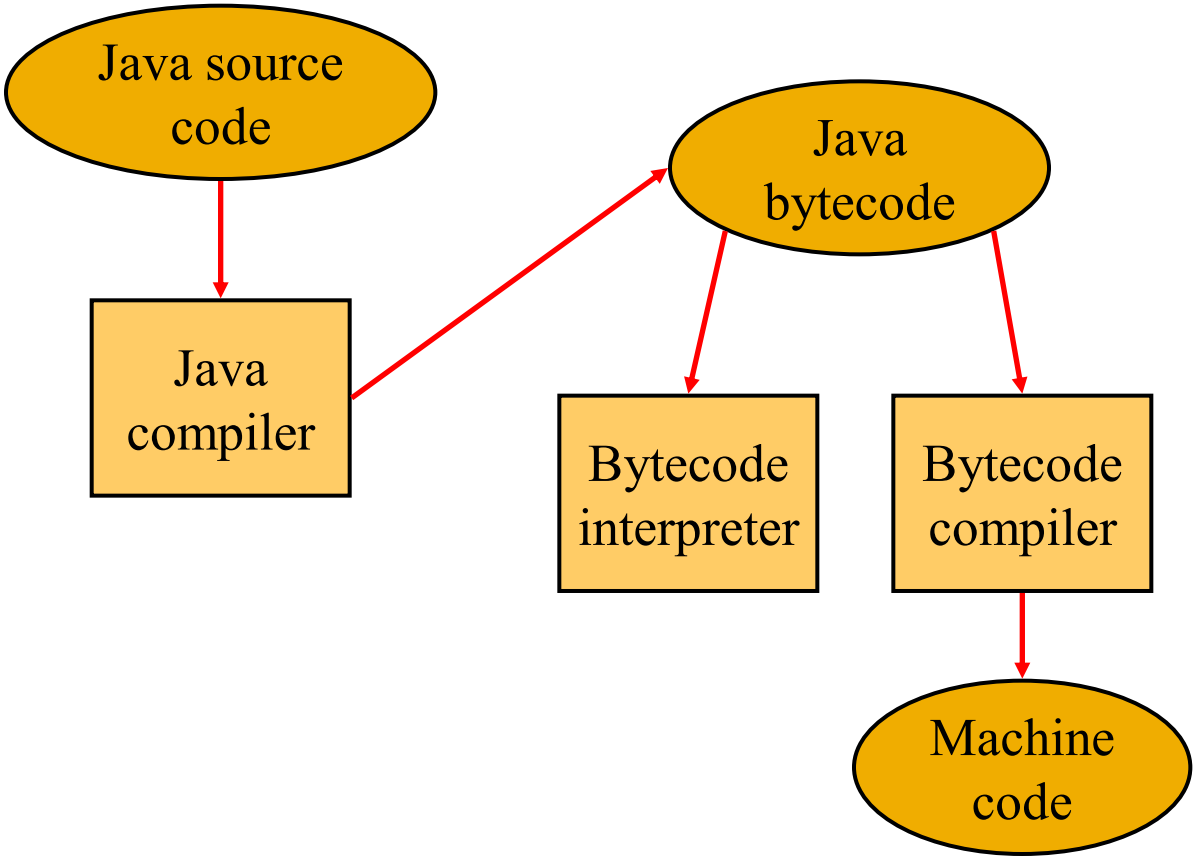
Java bytecode is not the machine language for any traditional CPU

Another software tool, called an ***interpreter***, translates bytecode into machine language and executes it

Therefore the Java compiler is not tied to any particular machine

Java is considered to be ***architecture-neutral***

Java Translation



Development Environments

There are many programs that support the development of Java software, including

- Sun Java Development Kit (JDK)
- **Eclipse**
- NetBeans
- BlueJ
- jGRASP

Though the details of these environments differ, the basic compilation and execution process is essentially the same

Syntax and Semantics

The ***syntax rules*** of a language define how we can put together symbols, reserved words, and identifiers to make a valid program

The ***semantics*** of a program statement define what that statement means (its purpose or role in a program)

A program that is syntactically correct is not necessarily logically (semantically) correct

A program will always do what we tell it to do, not what we meant to tell it to do

Errors

A program can have three types of errors

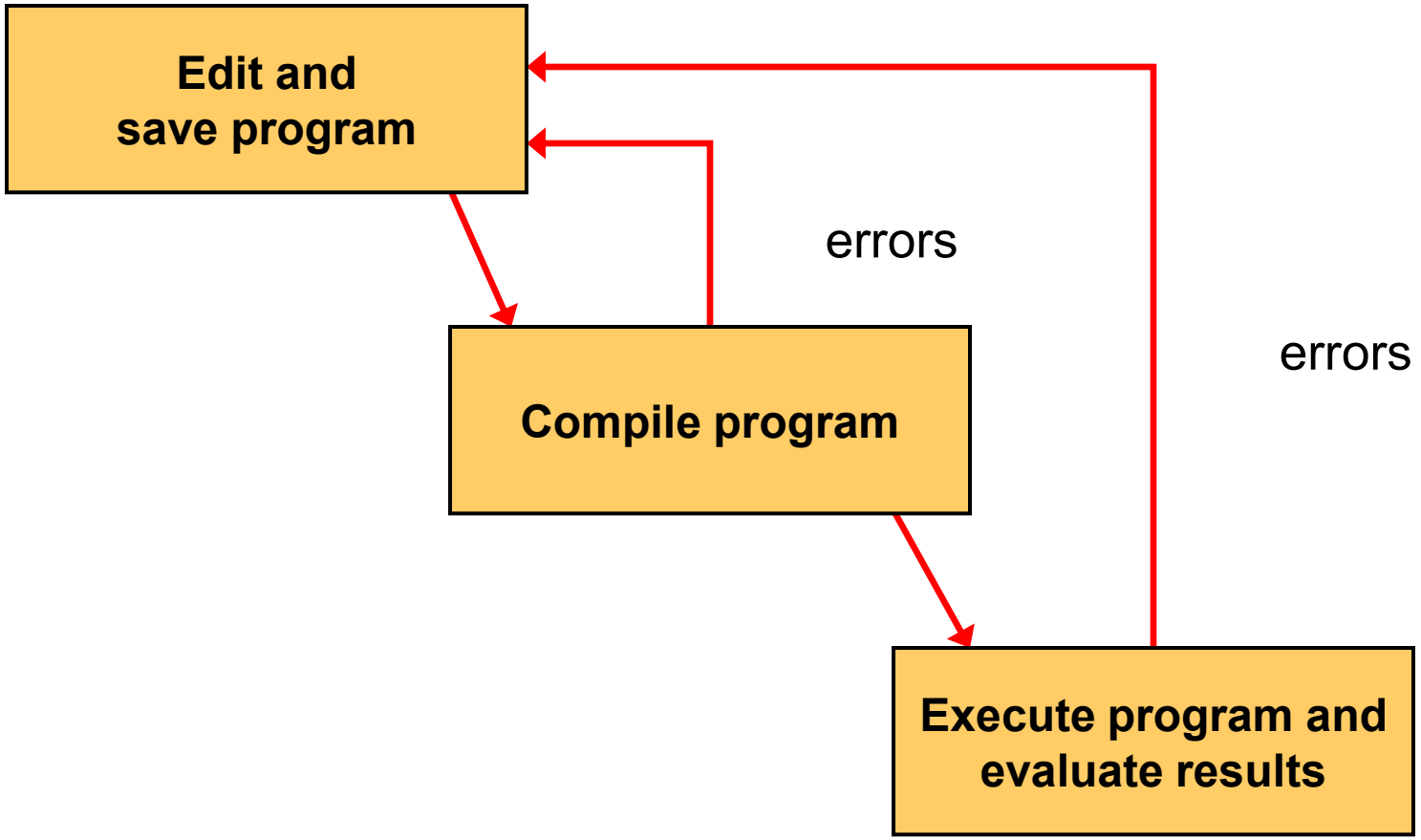
- The compiler will find syntax errors and other basic problems (***compile-time errors***)

  If compile-time errors exist, an executable version of the program is not created

- A problem can occur during program execution, such as trying to divide by zero, which causes a program to terminate abnormally (***run-time errors***)

- A program may run, but produce incorrect results, perhaps using an incorrect formula (***logical errors***)

Basic Program Development



Software Development Activities

Any proper software development effort consists of four basic ***development activities***

- establishing the requirements
- creating a design
- implementing the design
- testing

These steps also are never purely linear and often overlap and interact

Software Development Activities

Software requirements specify *what* a program must accomplish

Requirements are expressed in a document called a ***functional specification***

A ***software design*** indicates how a program will accomplish its requirements

Implementation is the process of writing the source code that will solve the problem

Testing is the act of ensuring that a program will solve the intended problem given all of the constraints under which it must perform

Object-Oriented Concepts

Object-Oriented Programming

Java is an ***object-oriented*** programming language

- An object is a fundamental entity
- Objects can be used effectively to represent real-world entities
- i.e., an object might represent an employee in a company
- Each employee object handles the processing and data management related to that employee

Objects

An object has

- ***state*** - descriptive characteristics
- ***behaviors*** - what it can do (or what can be done to it)

The state of a bank account includes its account number and its current balance

The behaviors associated with a bank account include the ability to make deposits and withdrawals

Note that the behavior of an object might change its state

Classes

An object is defined by a ***class***

A class is the blueprint of an object

- Uses methods to define the behaviors of the object
- The class that contains the main method of a Java program represents the entire program
- A class represents a concept, and an object represents the embodiment of that concept
- Multiple objects can be created from the same class

Objects and Classes

A class
(the concept)

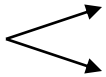
An object
(the realization)

John's Bank Account
Balance: $5,257

Jason's Bank Account
Balance: $1,245,069

Mary's Bank Account
Balance: $16,833

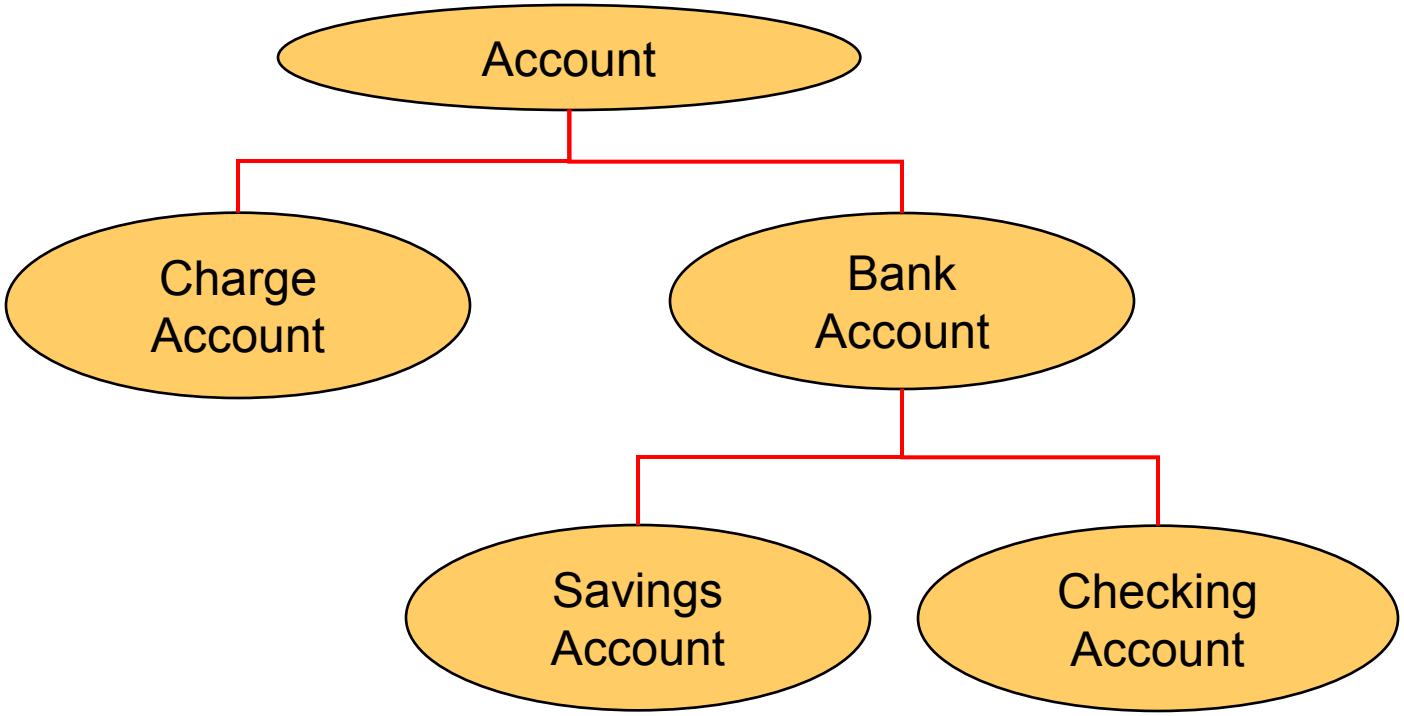
Multiple objects
from the same class



Inheritance

One class can be used to derive another via ***inheritance***

Classes can be organized into hierarchies



Summary

- Programming involves writing source code that is translated into machine readable instructions
- It is important to use meaningful identifier names, make use of whitespace, and provide comments
- Java is an object-oriented programming language
    - Java programs consist of classes, methods, and statements. The main method is the entry point.
    - A Java program is written as source code which is then translated into bytecode, which is then interpreted to machine code
- Software development activities aim to satisfy program requirements and eliminate errors