جامِعة بيرزيت

BIRZEIT UNIVERSITY

Java Programming

Chapter 1 Introduction to Computers, Programs, and Java

Course Description

– Demonstrate understanding of classes, constructors, objects, and instantiation.

– Access variables and modifier keywords.

– Develop methods using parameters and return values.

– Build control structures in an object-oriented environment.

– Convert data types using API methods and objects.

– Design object-oriented programs using scope, inheritance, and other design techniques.

– Create an object-oriented application using Java packages, APIs. and interfaces, in conjunction with classes and objects.

Uploaded By: Jibreel Bornat

Logistics

- Instructor: Bassem Sayrafi (Masri 316)
- Facebook Group: www.facebook.com/groups/bzucomp231
- Textbook:
  - Introduction To JAVA Programming, **11**th edition.
  - Author: Y. Daniel Liang.
  - Publisher: Prentice Hall.
- Lab Manual:
  - **Title:** LABORATORY WORK BOOK
  - Eclipse

Special Regulations

- **Assignments**:
  - All assignments are individual efforts any duplicated copies will be treated as a cheating attempt which lead to ZERO mark.
  - Using code from the internet will be treated as cheating as well.
  - The assignments should be submitted through Ritaj within the specified deadline.
  - No late submissions are accepted.

Honor Code (ميثاق شرف)

- All work/code must be your own work. You are not allowed to submit code or parts of code that are not your own. Furthermore, you are not allowed to share your code with other students– even after you finish this class. Also you are not to ask others to show their work to you or write parts of the code for you.

- كل العمل يجب أن يكون عملك الخاص. لا يُسمح لك بتقديم برمجيات (كود) أو أجزاء من برمجيات لم تقم انت بإنتاجها. علاوة على ذلك ، لا يُسمح لك بمشاركة الكود الخاص بك مع طلاب آخرين حتى بعد الانتهاء من هذا الفصل. كما أنك لن تطلب من الآخرين إظهار أعمالهم لك أو كتابة أجزاء من الكود لك.

Interpreting/Compiling Source Code

A program written in a high-level language is called a *source program* or *source code*. Because a computer cannot understand a source program, a source program must be translated into machine code for execution. The translation can be done using another programming tool called an *interpreter* or a *compiler*.

Uploaded By: Jibreel Bornat

6

Interpreting Source Code

An interpreter reads one statement from the source code, translates it to the machine code or virtual machine code, and then executes it right away, as shown in the following figure. Note that a statement from the source code may be translated into several machine instructions.

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

7
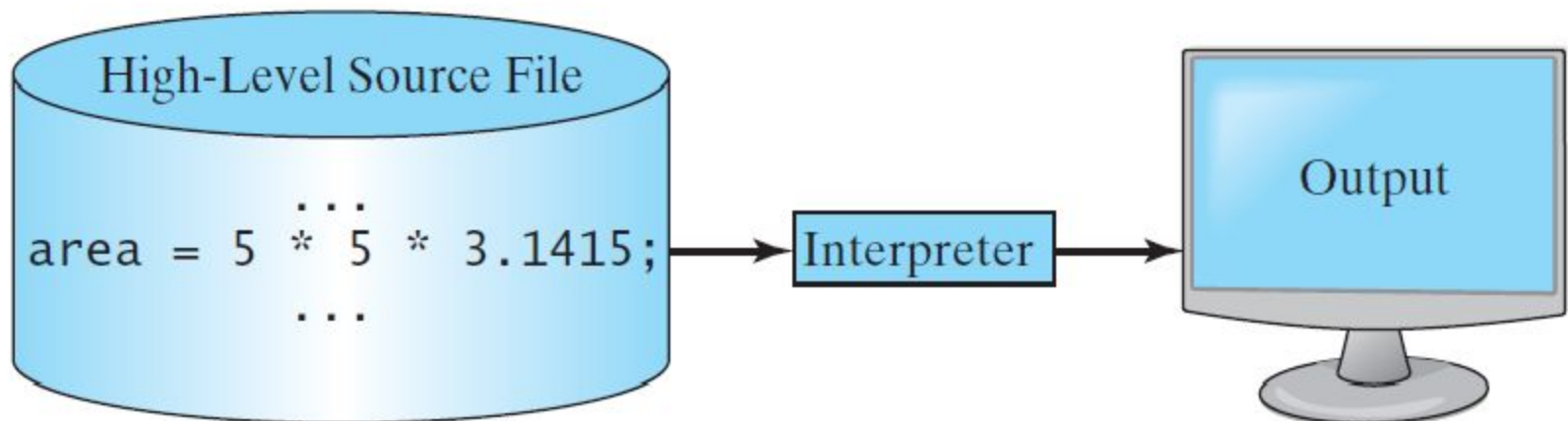
Compiling Source Code

A compiler translates the entire source code into a machine-code file, and the machine-code file is then executed, as shown in the following figure.

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

8
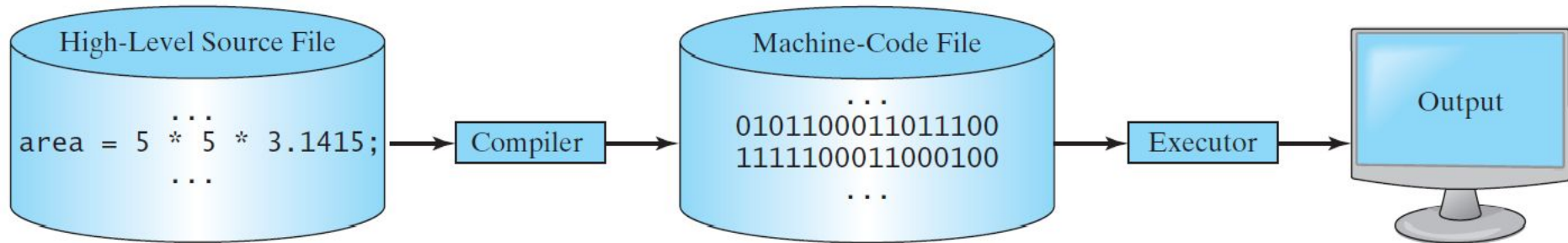
Java's History

- James Gosling and Sun Microsystems

- Oak

- Java, May 20, 1995, Sun World

- HotJava
  - The first Java-enabled Web browser

- Early History Website:

  http://www.java.com/en/javahistory/index.jsp

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

9

Why Java?

Java enables users to develop and deploy applications on the Internet for servers, desktop computers, and small hand-held devices. The future of computing is being profoundly influenced by the Internet, and Java promises to remain a big part of that future.

- Java is a general purpose programming language.

- Java is the Internet programming language.

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

Java, Web, and Beyond

- Java can be used to develop standalone applications.

- Java can be used to develop applications running from a browser.

- Java can also be used to develop applications for hand-held devices.

- Java can be used to develop applications for Web servers.

11

Uploaded By: Jibreel Bornat

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is partially modeled on C++, but greatly simplified and improved. Some people refer to Java as "C++--" because it is like C++ but with more functionality and fewer negative aspects.

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

12

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is inherently object-oriented. Although many object-oriented languages began strictly as procedural languages, Java was designed from the start to be object-oriented. Object-oriented programming (OOP) is a popular programming approach that is replacing traditional procedural programming techniques.

One of the central issues in software development is how to reuse code. Object-oriented programming provides great flexibility, modularity, clarity, and reusability through encapsulation, inheritance, and polymorphism.

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

Characteristics of Java

BIRZEIT UNIVERSITY

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Distributed computing involves several computers working together on a network. Java is designed to make distributed computing easy. Since networking capability is inherently integrated into Java, writing network programs is like sending and receiving data to and from a file.

Uploaded By: Jibreel Bornat
14

Characteristics of Java

BIRZEIT UNIVERSITY

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

You need an interpreter to run Java programs. The programs are compiled into the Java Virtual Machine code called bytecode. The bytecode is machine-independent and can run on any machine that has a Java interpreter, which is part of the Java Virtual Machine (JVM).

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

15

Characteristics of Java

BIRZEIT UNIVERSITY

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java compilers can detect many problems that would first show up at execution time in other languages.

Java has eliminated certain types of error-prone programming constructs found in other languages.

Java has a runtime exception-handling feature to provide programming support for robustness.

Characteristics of Java

BIRZEIT UNIVERSITY

- Java Is Simple

- Java Is Object-Oriented

- Java Is Distributed

- Java Is Interpreted

- Java Is Robust

- Java Is Secure

- Java Is Architecture-Neutral

- Java Is Portable

- Java's Performance

- Java Is Multithreaded

- Java Is Dynamic

Java implements several security mechanisms to protect your system against harm caused by stray programs.

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

17

Characteristics of Java

BIRZEIT UNIVERSITY

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Write once, run anywhere

With a Java Virtual Machine (JVM), you can write one program that will run on any platform.

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

18

Characteristics of Java

BIRZEIT UNIVERSITY

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Because Java is architecture neutral, Java programs are portable. They can be run on any platform without being recompiled.

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java's performance Because Java is architecture neutral, Java programs are portable. They can be run on any platform without being recompiled.

Uploaded By: Jibreel Bornat

20

Characteristics of Java

BIRZEIT UNIVERSITY

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Multithread programming is smoothly integrated in Java, whereas in other languages you have to call procedures specific to the operating system to enable multithreading.

Characteristics of Java

BIRZEIT UNIVERSITY

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java was designed to adapt to an evolving environment. New code can be loaded on the fly without recompilation. There is no need for developers to create, and for users to install, major new software versions. New features can be incorporated transparently as needed.

JDK Versions

- JDK 1.02 (1995)
- …
- JDK 1.4 (2002)
- JDK 1.5 (2004) a. k. a. JDK 5 or Java 5
- JDK 1.6 (2006) a. k. a. JDK 6 or Java 6
- JDK 1.7 (2011) a. k. a. JDK 7 or Java 7
- JDK 1.8 (2014) a. k. a. JDK 8 or Java 8
- …
- JDK 17 (2021 soon!)

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

JDK Editions

- Java Standard Edition (J2SE)
  - J2SE can be used to develop client-side standalone applications or applets.

- Java Enterprise Edition (J2EE)
  - J2EE can be used to develop server-side applications such as Java servlets, Java ServerPages, and Java ServerFaces.

- Java Micro Edition (J2ME).
  - J2ME can be used to develop applications for mobile devices such as cell phones.

This book uses J2SE to introduce Java programming.

Uploaded By: Jibreel Bornat

Popular Java IDEs

- NetBeans

- Eclipse

Uploaded By: Jibreel Bornat
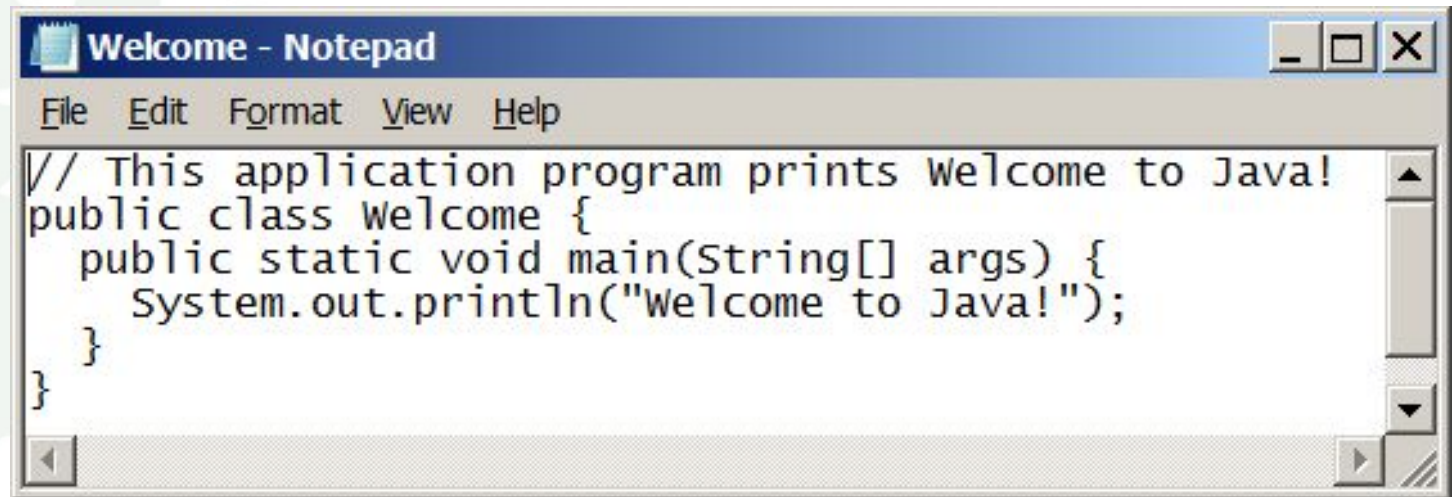
25

A Simple Java Program

Listing 1.1

```java
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Welcome

Run

Uploaded By: Jibreel Bornat

Creating and Editing Using NotePad

To use NotePad, type
   notepad Welcome.java
from the DOS prompt.

Command Prompt

```
C:\book>notepad Welcome.java_
```


Welcome - Notepad

File   Edit   Format   View   Help

```
// This application program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```
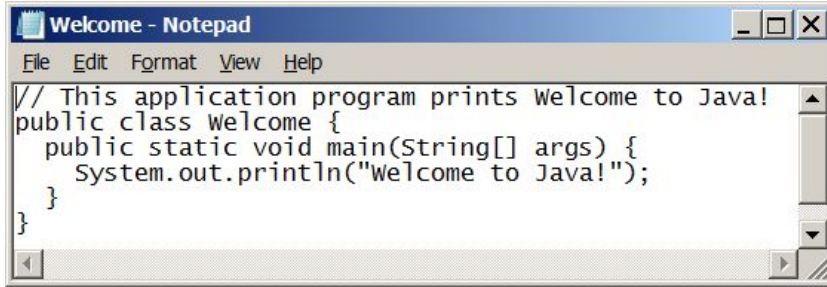
Creating, Compiling, and Running Programs

```
Welcome - Notepad
File  Edit  Format  View  Help
// This application program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Source code (developed by the programmer)

```
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

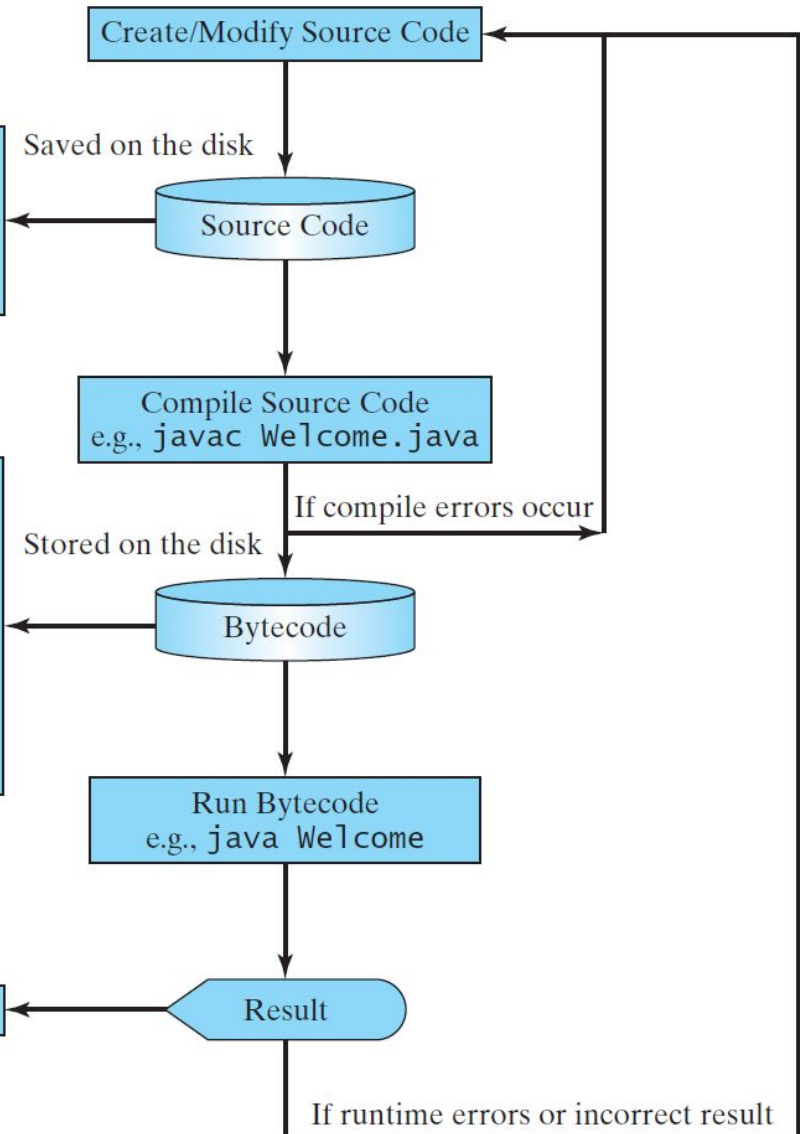**Bytecode (generated by the compiler for JVM to read and interpret)**

```
…
Method Welcome()
  0 aload_0
  …

Method void main(java.lang.String[])
  0 getstatic #2 …
  3 ldc #3 <String "Welcome to Java!">
  5 invokevirtual #4 …
  8 return
```

"Welcome to Java " is displayed on the console

```
Welcome to Java!
```

Create/Modify Source Code

Saved on the disk

Source Code

Compile Source Code
e.g., `javac Welcome.java`

If compile errors occur

Stored on the disk

Bytecode

Run Bytecode
e.g., `java Welcome`

Result

If runtime errors or incorrect result

Compiling Java Source Code

You can port a source program to any machine with appropriate compilers. The source program must be recompiled, however, because the object program can only run on a specific machine. Nowadays computers are networked to work together. Java was designed to run object programs on any platform. With Java, you write the program once, and compile the source program into a special type of object code, known as *bytecode*. The bytecode can then run on any computer with a Java Virtual Machine, as shown below. Java Virtual Machine is a software that interprets Java bytecode.

(a)

(b)

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat
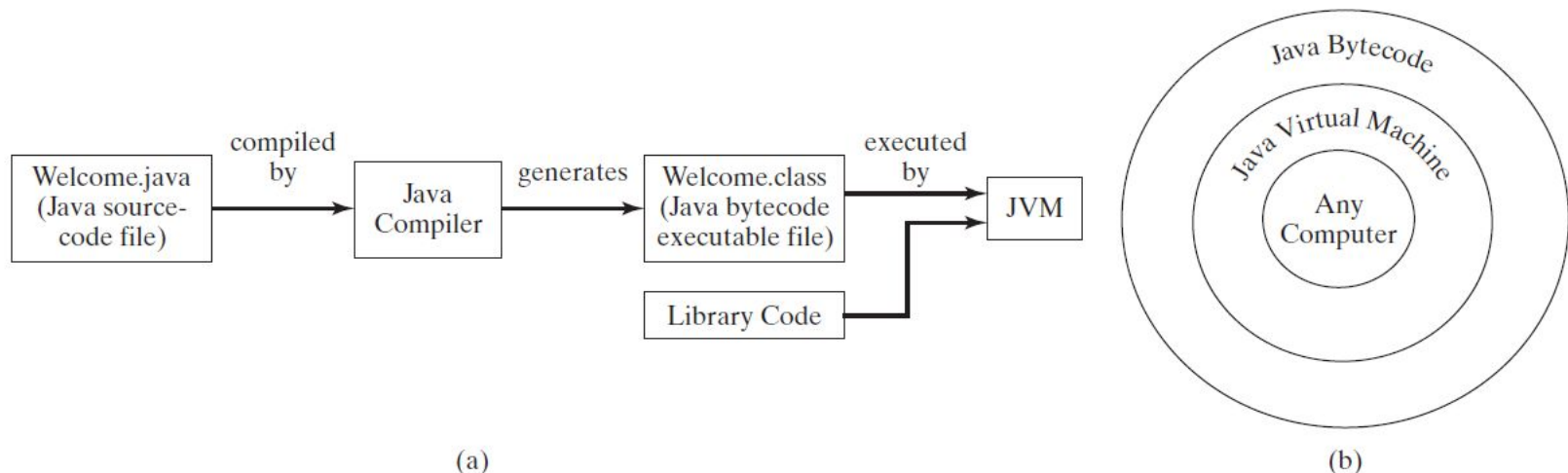
Trace a Program Execution

```java
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Command Prompt

```
C:\book>java Welcome
Welcome to Java!

C:\book>
```

print a message to the console

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

30

Two More Simple Examples

| WelcomeWithThreeMessages | Run |
| ComputeExpression | Run |

Uploaded By: Jibreel Bornat
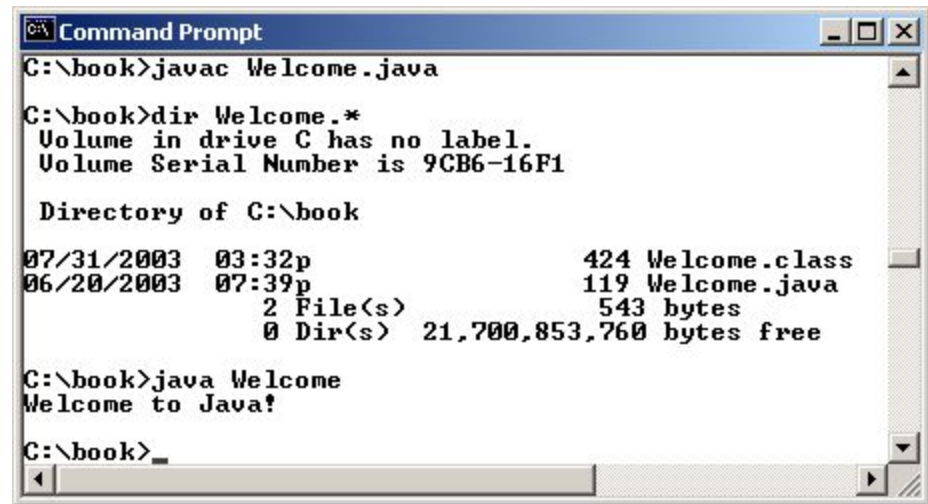
31

Compiling and Running Java from the Command Window

- Set path to JDK bin directory
  - set path=c:\Program Files\java\jdk1.8.0\bin
- Set classpath to include the current directory
  - set classpath=.
- Compile
  - javac Welcome.java
- Run
  - java Welcome

```
Command Prompt                                    _ □ ×
C:\book>javac Welcome.java

C:\book>dir Welcome.*
 Volume in drive C has no label.
 Volume Serial Number is 9CB6-16F1

 Directory of C:\book

07/31/2003  03:32p                424 Welcome.class
06/20/2003  07:39p                119 Welcome.java
             2 File(s)             543 bytes
             0 Dir(s)  21,700,853,760 bytes free

C:\book>java Welcome
Welcome to Java!

C:\book>_
```

Anatomy of a Java Program

- Class name

- Main method

- Statements

- Statement terminator

- Reserved words

- Comments

- Blocks

33

Class Name

Every Java program must have at least one class. Each class has a name. By convention, class names start with an uppercase letter. In this example, the class name is Welcome.

```java
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Uploaded By: Jibreel Bornat 34

Main Method

Line 2 defines the main method. In order to run a class, the class must contain a method named main. The program is executed from the main method.

```java
// This program prints Welcome to Java!
public class Welcome {
   public static void main(String[] args) {
      System.out.println("Welcome to Java!");
   }
}
```

35

Uploaded By: Jibreel Bornat

Statement

A statement represents an action or a sequence of actions. The statement System.out.println("Welcome to Java!") in the program in Listing 1.1 is a statement to display the greeting "Welcome to Java!".

```java
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Uploaded By: Jibreel Bornat

Statement Terminator

Every statement in Java ends with a semicolon (;).

```
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

STUDENTS-HUB.com   Uploaded By: Jibreel Bornat

Reserved words

Reserved words or keywords are words that have a specific meaning to the compiler and cannot be used for other purposes in the program. For example, when the compiler sees the word class, it understands that the word after class is the name for the class.

```java
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Blocks

A pair of braces in a program forms a block that groups components of a program.

```
public class Test {                              ←────────────┐
   public static void main(String[] args) {   ←──────┐       │  Class block
      System.out.println("Welcome to Java!");  Method block  │
   }  ←──────────────────────────────────────────────┘       │
}  ←──────────────────────────────────────────────────────────┘
```

Special Symbols

Character	Name	Description
{}	Opening and closing braces	Denotes a block to enclose statements.
()	Opening and closing parentheses	Used with methods.
[]	Opening and closing brackets	Denotes an array.
//	Double slashes	Precedes a comment line.
" "	Opening and closing quotation marks	Enclosing a string (i.e., sequence of characters).
;	Semicolon	Marks the end of a statement.

Uploaded By: Jibreel Bornat

{ ... }

```
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Uploaded By: Jibreel Bornat

(...)

```
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Uploaded By: Jibreel Bornat

42

.
;

```java
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!") ;
  }
}
```

// ...

```
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Uploaded By: Jibreel Bornat

44

" ... "

```
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Programming Style and Documentation

- Appropriate Comments
- Naming Conventions
- Proper Indentation and Spacing Lines
- Block Styles

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

46

Appropriate Comments

Include a summary at the beginning of the program to explain what the program does, its key features, its supporting data structures, and any unique techniques it uses.

Include your name, class section, instructor, date, and a brief description at the beginning of the program.

Uploaded By: Jibreel Bornat

47

Naming Conventions

- Choose meaningful and descriptive names.

- Class names:
  - Capitalize the first letter of each word in the name. For example, the class name `ComputeExpression`.

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

Proper Indentation and Spacing

- Indentation
  - Indent two spaces.

- Spacing
  - Use blank line to separate segments of the code.

Uploaded By: Jibreel Bornat

Block Styles

Use end-of-line style for braces.

Next-line style →

```
public class Test
{
  public static void main(String[] args)
  {
    System.out.println("Block Styles");
  }
}
```

```
public class Test {
  public static void main(String[] args) {
    System.out.println("Block Styles");
  }
}
```

← *End-of-line style*

Liang, Introduction to Java Programming, Eleventh Edition, (c) 2017 Pearson Education, Inc. All rights reserved.

Programming Errors

- ## Syntax Errors
  - Detected by the compiler

- ## Runtime Errors
  - Causes the program to abort

- ## Logic Errors
  - Produces incorrect result

51

Uploaded By: Jibreel Bornat

Syntax Errors

```
public class ShowSyntaxErrors {
  public static main(String[] args) {
    System.out.println("Welcome to Java);
  }
}
```

ShowSyntaxErrors   Run

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

Runtime Errors

```java
public class ShowRuntimeErrors {
  public static void main(String[] args) {
    System.out.println(1 / 0);
  }
}
```

ShowRuntimeErrors   Run

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat

Logic Errors

```java
public class ShowLogicErrors {
  public static void main(String[] args) {
    System.out.println("Celsius 35 is Fahrenheit degree ");
    System.out.println((9 / 5) * 35 + 32);
  }
}
```

ShowLogicErrors   Run

STUDENTS-HUB.com

Uploaded By: Jibreel Bornat