



COMPUTER SCIENCE DEPARTMENT FACULTY OF
ENGINEERING AND TECHNOLOGY

ADVANCED PROGRAMMING COMP231

Instructor :Murad Njoun
Office : Masri322

Chapter 1 Introduction to Computers, Programs, and Java

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun



Programs

Computer *programs*, known as *software*, are instructions to the computer.

You tell a computer what to do through programs. Without programs, a computer is an empty machine. Computers do not understand human languages, so you need to use computer languages to communicate with them.

Programs are written using programming languages.

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun



Programming Languages

Machine Language Assembly Language High-Level Language

Machine language is a set of primitive instructions built into every computer. The instructions are in the form of binary code, so you have to enter binary codes for various instructions. Program with native machine language is a tedious process. Moreover the programs are highly difficult to read and modify. For example, to add two numbers, you might write an instruction in binary like this:

```
1101101010011010
```



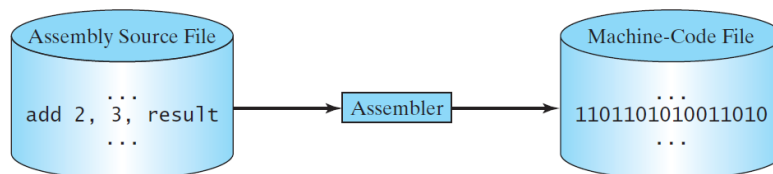
liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Programming Languages

Machine Language **Assembly Language** High-Level Language

Assembly languages were developed to make programming easy. Since the computer cannot understand assembly language, however, a program called assembler is used to convert assembly language programs into machine code. For example, to add two numbers, you might write an instruction in assembly code like this:

```
ADDF3 R1, R2, R3
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Programming Languages

Machine Language Assembly Language **High-Level Language**

The high-level languages are English-like and easy to learn and program. For example, the following is a high-level language statement that computes the area of a circle with radius 5:

```
area = 5 * 5 * 3.1415;
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Interpreting/Compiling Source Code

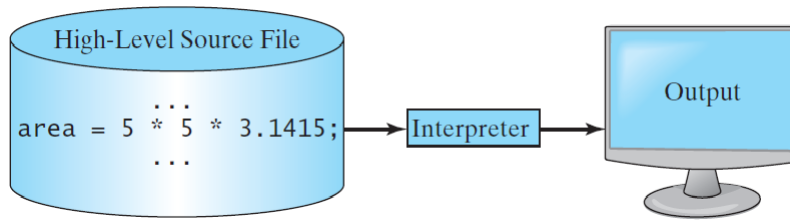
A program written in a high-level language is called a *source program* or **source code**. Because a computer cannot understand a source program, a source program must be translated into machine code for execution. The translation can be done using another programming tool called an **interpreter or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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

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.

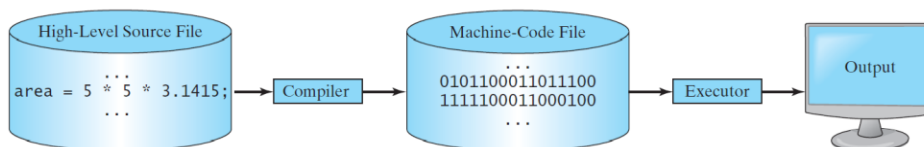


liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



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.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Why Java?

The answer is that 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 the Internet programming language.

- Java is a general purpose programming language.
- Java is the Internet programming language.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

9

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.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

10

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

www.cs.armstrong.edu/liang/JavaCharacteristics.pdf

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



11

Companion
Website

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.



12

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

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.**



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

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.



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

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).



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 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.



Companion
Website

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

17

Companion
Website

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

Write once, run anywhere

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

18

Companion
Website

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

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

19

Companion
Website

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.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

20

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

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

- 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 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.

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



23

Popular Java IDEs

- NetBeans (V.10)
- Eclipse (Luna, Mars, Oxygen)



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



24

A Simple Java Program

Listing 1.1

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

Welcome

Run

Note: Clicking the green button displays the source code with interactive animation. You can also run the code in a browser. Internet connection is needed for this button.

Note: Clicking the blue button runs the code from Windows. If you cannot run the buttons, see <http://www.example.pearsoncmg.com/Slide/javaslide/notes.doc>.

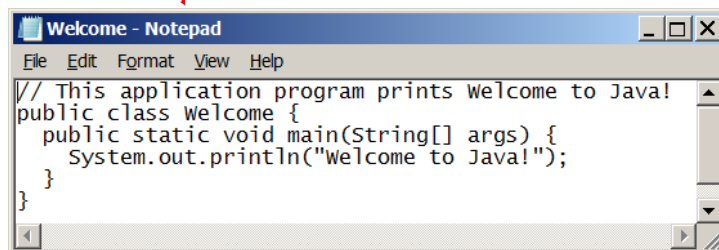


liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

25

Creating and Editing Using NotePad

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



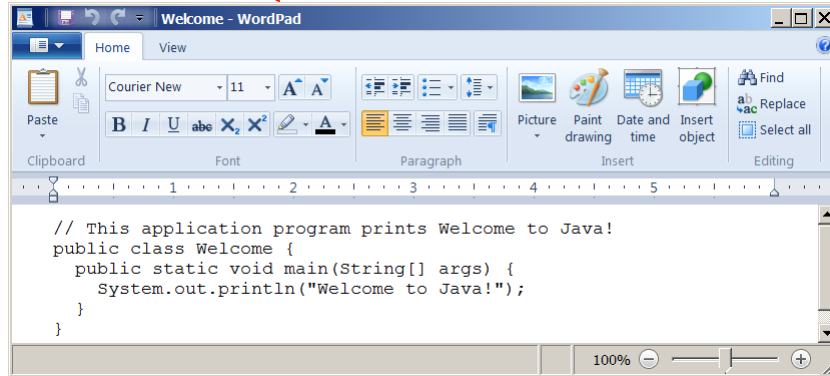
liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



26

Creating and Editing Using WordPad

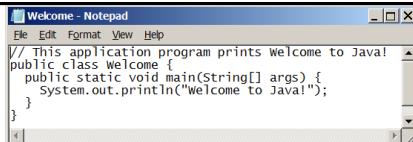
To use WordPad, type
write Welcome.java
from the DOS prompt.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Creating, Compiling, and Running Programs



Source code (developed by the programmer)

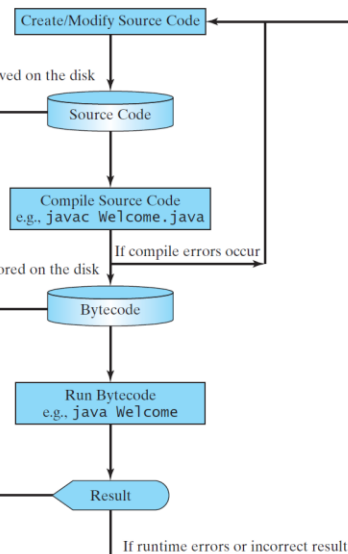
```
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

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!
```

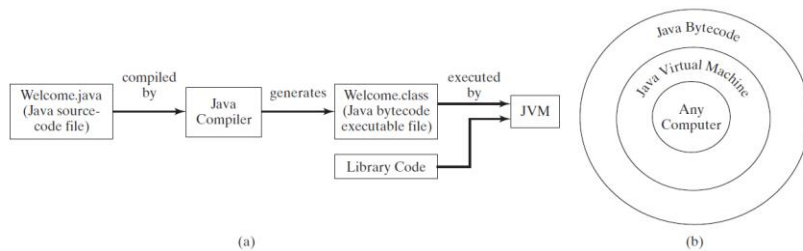


liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



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.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

29

animation

Trace a Program Execution

Enter main method

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

30

animation

Trace a Program Execution

Execute statement

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

animation

Trace a Program Execution

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



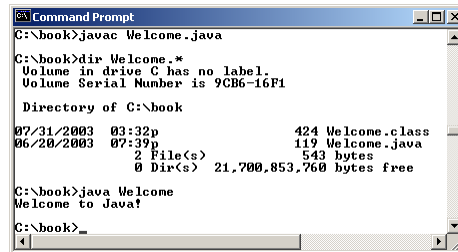
print a message to the console



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

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



```

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>_
  
```

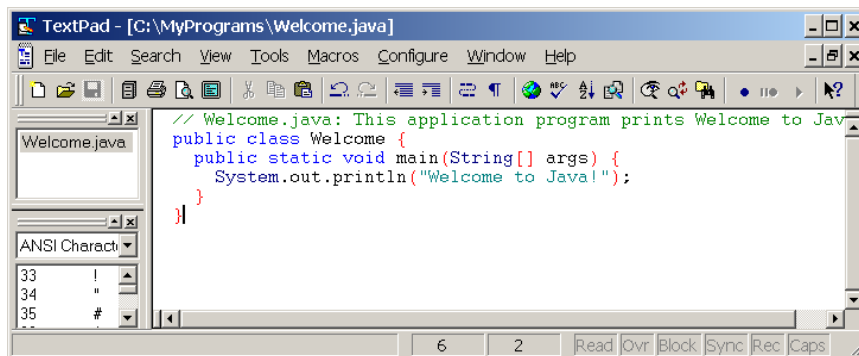


liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

33

Compiling and Running Java from TextPad

- See Supplement II.A on the Website for details



```

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

34

Anatomy of a Java Program

- Class name
- Main method
- Statements
- Statement terminator
- Reserved words
- Comments
- Blocks



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

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.

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

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.

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

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



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!".

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

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



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!");
    }
}
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

39

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.

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

40

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) {
        System.out.println("Welcome to Java!");
    }
}
```

Diagram illustrating blocks in the code above:

- The outermost braces `{` and `}` are labeled "Class block".
- The inner braces `{` and `}` around the `main` method are labeled "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.



```
{ ... }
```

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

43

```
( ... )
```

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

44

;

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

// ...

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

" ... "

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

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun



Programming Style and Documentation

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

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun



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.



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`.



Proper Indentation and Spacing

- Indentation
 - Indent two spaces.
- Spacing
 - Use blank line to separate segments of the code.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

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");
    }
}
```

*End-of-line
style*

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Programming Errors

- Syntax Errors
 - Detected by the compiler
- Runtime Errors
 - Causes the program to abort
- Logic Errors
 - Produces incorrect result



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Syntax Errors

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

ShowSyntaxErrors

Run



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Runtime Errors

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

ShowRuntimeErrors

Run



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Logic Errors

```
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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum