

2. Getting Started with Java

13 Aug 2015

Objectives

- Identify the basic components of Java programs.
- Write simple Java programs.
- Describe the difference between object declaration and object creation.
- Describe the process of creating and running Java programs.

Objectives

- Use the Date, SimpleDateFormat, String, and Scanner classes from the standard Java classes.

First Java Program

```
/*  
    Chapter 2 Sample Program: Displaying a Window  
    File: Ch2Sample1.java  
*/  
import javax.swing.*;  
class Ch2Sample1 {  
    public static void main(String[] args) {  
        JFrame    myWindow;  
        myWindow = new JFrame();  
        myWindow.setSize(300, 200);  
        myWindow.setTitle("My First Java Program");  
        myWindow.setVisible(true);  
    }  
}
```

First Java Program

- Output:



Figure 2.1 Result of running the **Ch2Sample1** program. The window size is 300 by 200 pixels and has the title **My First Java Program**.

First Java Program

- Type of Classes in Java:
 1. Standard Java Classes: คลาสมาตรฐานที่มากับ JDK สามารถเรียกใช้งานได้เลย
 - Ex. JFrame, String, Integer, Date
 2. Programmer-defined Classes (Custom Classes): คลาสที่โปรแกรมเมอร์สร้างขึ้นเอง
 - Ex. Student, Customer, Account

Program Diagram

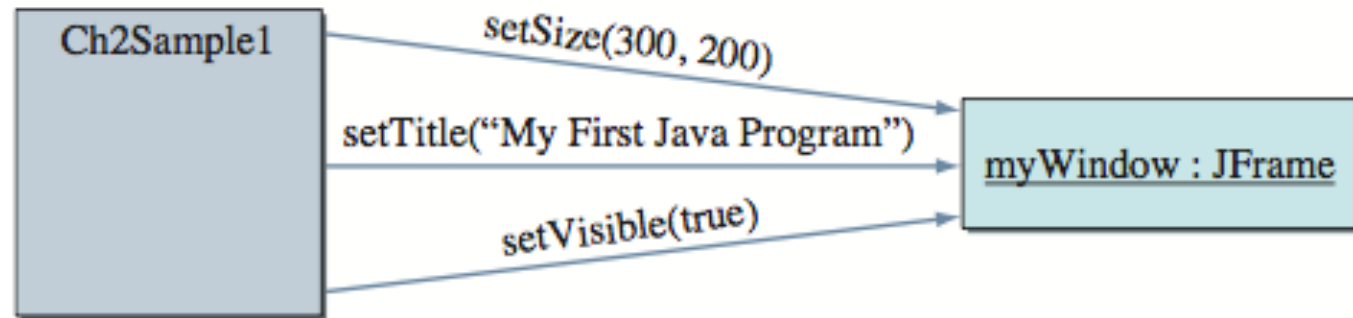


Figure 2.2 The program diagram for the **Ch2Sample1** program.

Dependency Relationship

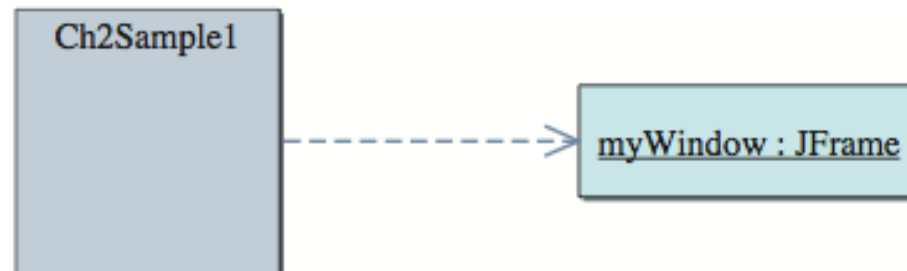


Figure 2.3 The program diagram for the **Ch2Sample1** program that shows the dependency relationship.

Main Method

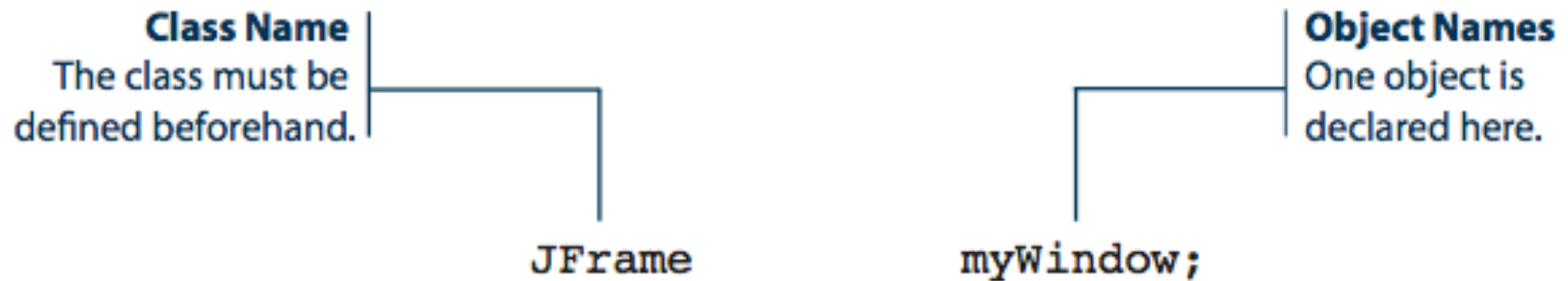
```
JFrame    myWindow;  
  
myWindow = new JFrame();  
  
myWindow.setSize(300, 200);  
myWindow.setTitle("My First Java Program");  
myWindow.setVisible(true);
```

Object Declaration

- คือการประกาศชื่อของ object และระบุว่า object นั้นสร้างมาจาก class ไດ มีรูปแบบดังนี้

`<CLASS_NAME> <OBJECT_NAME>;`

Object Declaration



Here are more examples:

```
Account    checking;  
Customer   john, jack, jill;
```

Object Declaration

- Java Identifier: เป็นการตั้งชื่อให้ class, object หรือ method โดยมีเงื่อนไขคือ
 - สร้างขึ้นจาก
 - ตัวอักษร
 - ตัวเลข
 - Underscores (_)
 - Dollar Sign (\$)
 - จะขึ้นต้นด้วยตัวเลขไม่ได้

Object Declaration

- Valid Identifier:

Student

MyStudent

student1

_std1

\$customer

CUST_1

นักศึกษา

นศ1

_นศ๑

- Invalid Identifier:

Student#1

My-Student

1stCustomer

นศ.

Customer 1

Object Declaration

- Java Standard Naming Convention: เป็นหลัก
การตั้งชื่อ Identifier ให้เป็นมาตรฐาน สามารถ
อ่าน โค้ดและทำความเข้าใจได้ง่าย
 - การตั้งชื่อ class
 - ควรเป็นภาษาอังกฤษ
 - ควรขึ้นต้นด้วยตัวใหญ่
 - ถ้าชื่อเกิดจากหลายคำ ให้ขึ้นต้นแต่ละคำด้วยตัวใหญ่
 - Ex. MyMainWindow, SavingAccount, VIPCustomer

Object Declaration

- การตั้งชื่อ object
 - ควรเป็นภาษาอังกฤษ
 - ควรขึ้นต้นด้วยตัวเล็ก
 - ถ้าชื่อเกิดจากหลายคำ ให้ขึ้นต้นแต่ละคำด้วยตัวใหญ่
- Ex. myMainWindow, savingAccount, vipCustomer

Object Declaration

- Java Standard Naming Convention

Category	Convention	Example
Class	Use an uppercase letter for the first letter of the class names. If the name consists of multiple words, the first letter of every word is capitalized.	Customer MainWindow MyInputHandler
Instance	Use a lowercase letter for the first letter of the object names. If the name consists of multiple words, the first letter of every word (except the first word) is capitalized.	customer inputHandler myFirstApplication
Constant	(Note: Sample use of a constant will appear in Chap. 4. We include it here for completeness and easy reference later.) Use all uppercase letters. If the constant consists of multiple words, the underscore characters are used to separate the words.	DEFAULT_RATE DEG_TO_RAD CANCEL
Package	Use all lowercase letters.	java game finance

Main Method

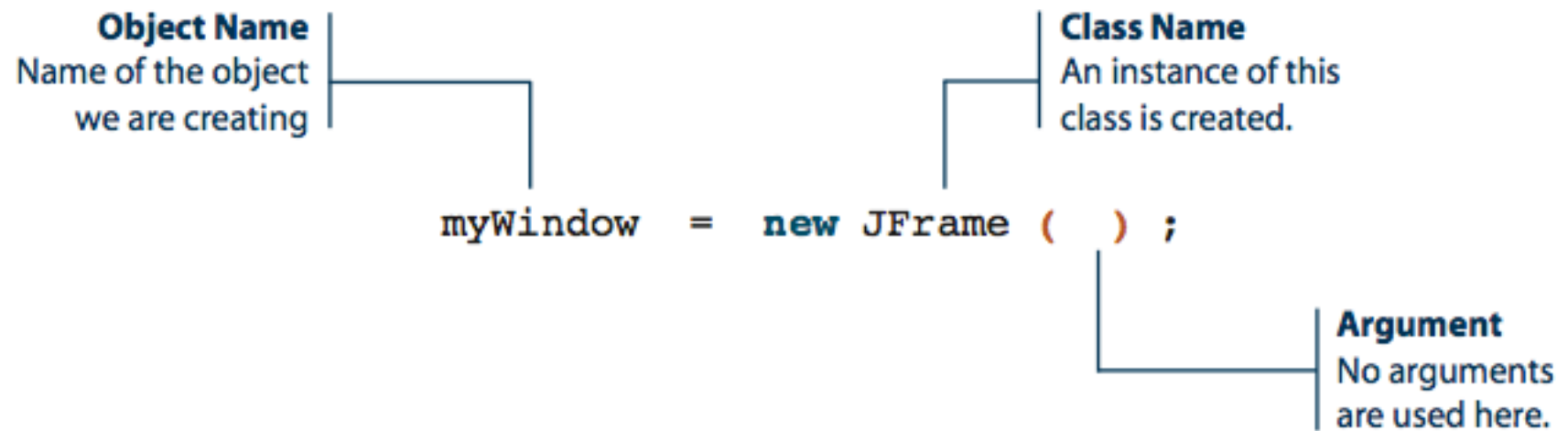
```
JFrame    myWindow;  
  
myWindow = new JFrame();  
  
myWindow.setSize(300, 200);  
myWindow.setTitle("My First Java Program");  
myWindow.setVisible(true);
```

Object Creation

- คือการสร้าง object ขึ้นจริง โดยการใช้ 'new' operator และระบบให้เข้าถึง object ผ่านทางชื่อตัวแปรที่ประกาศไว้ก่อนแล้ว มีรูปแบบดังนี้

`<OBJECT_NAME> = new <CLASS_NAME>(<args>);`

Object Creation




Object Declaration and Creation

Ⓐ `Account account;`
`account = new Account();`

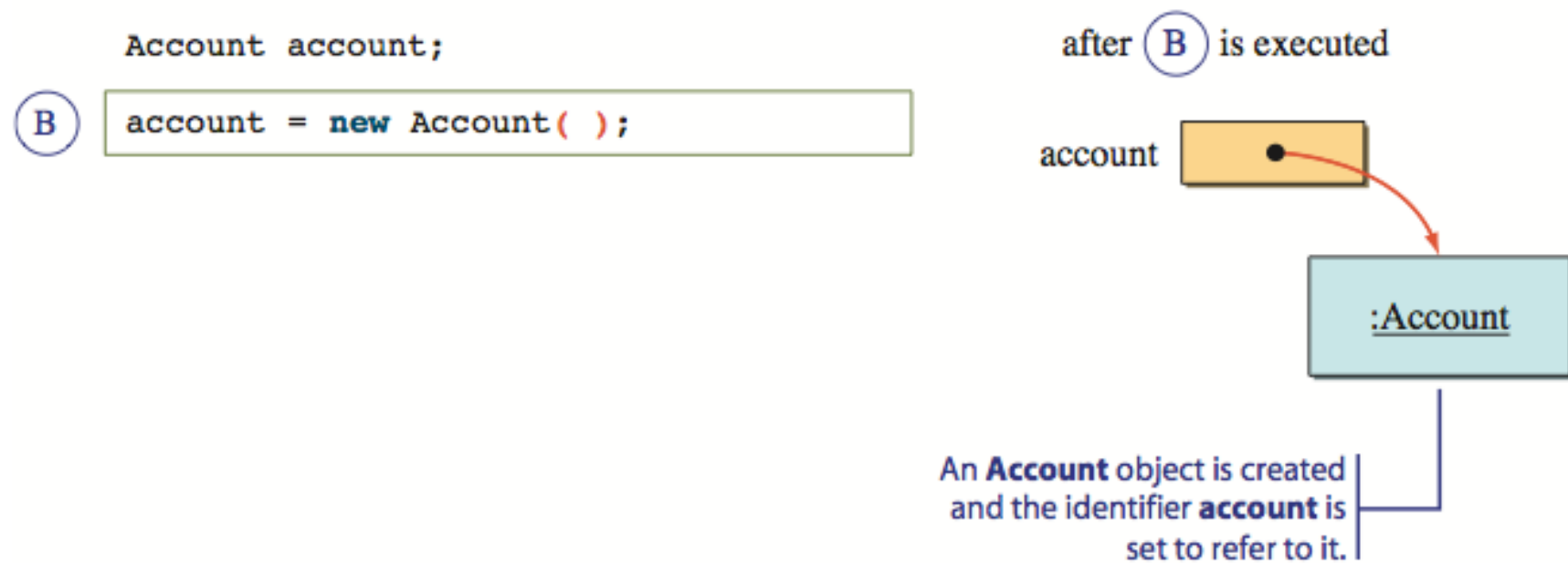
State of Memory

after Ⓐ is executed

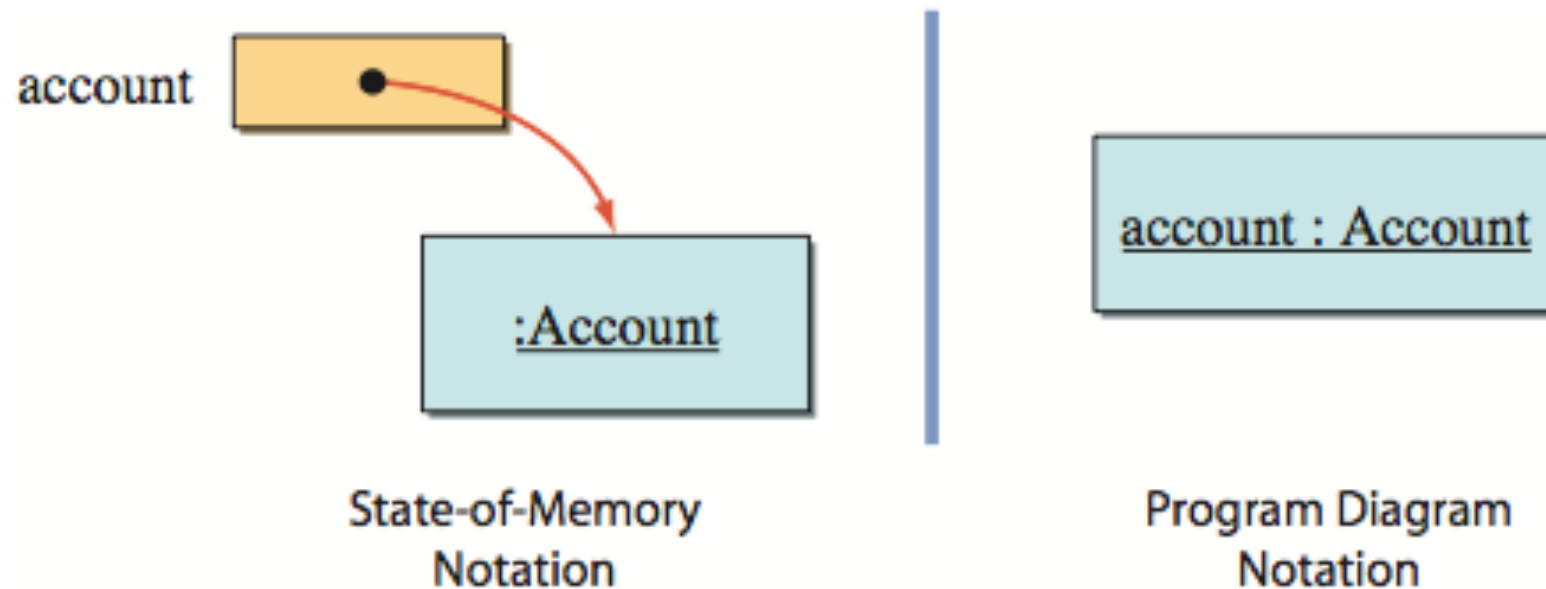
account 

The identifier **account** is
declared and space is
allocated in memory.

Object Declaration and Creation



Object Declaration and Creation



The state-of-memory diagram uses the same UML notation, but it also includes symbols and notations not found in UML.

Object Declaration and Creation

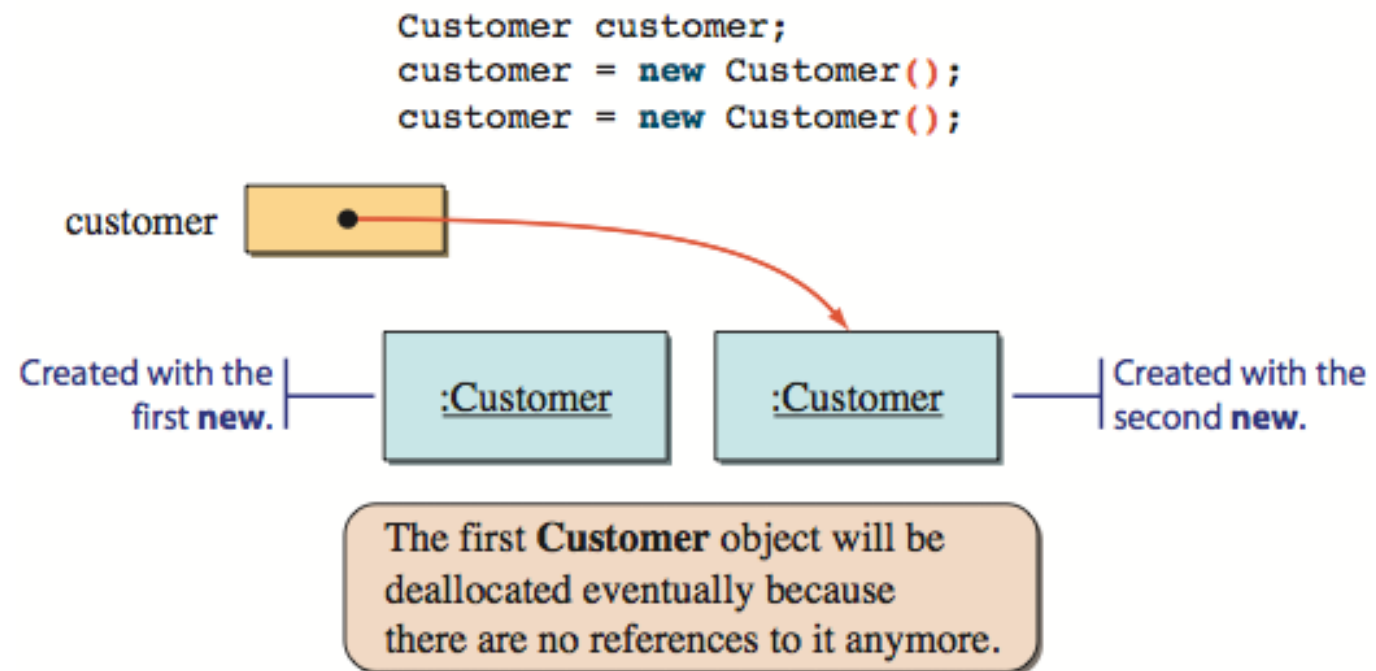


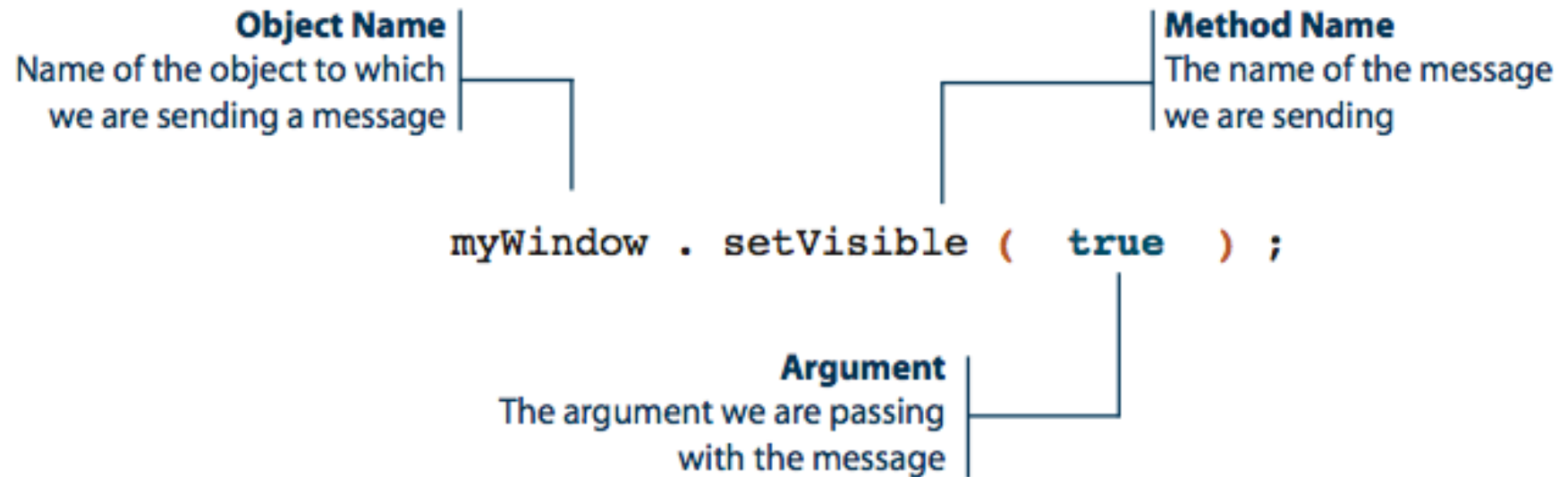
Figure 2.6 The state after two `new` commands are executed.

Message Sending

- คือการสั่งให้ object เรียกใช้งาน method ใดๆ มีรูปแบบดังนี้

`<OBJECT_NAME>.<METHOD_NAME>(<args>);`

Message Sending



Message Sending

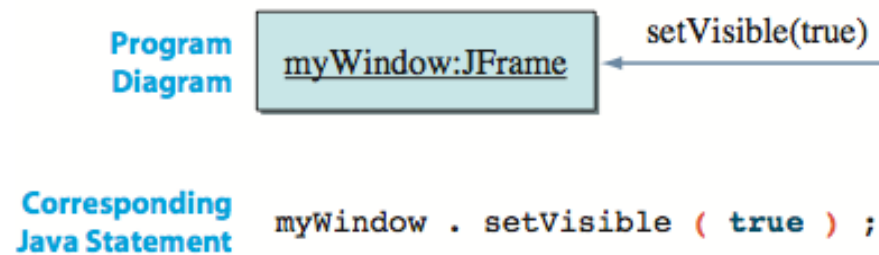


Figure 2.7 Correspondence between message sending as represented in the program diagram and in the actual Java statement.

Program Components

- Sample Program: Ch2Sample1

```
/*
    Chapter 2 Sample Program: Displaying a Window
    File: Ch2Sample1.java
*/
import javax.swing.*;
class Ch2Sample1 {
    public static void main(String[] args) {
        JFrame    myWindow;
        myWindow = new JFrame();
        myWindow.setSize(300, 200);
        myWindow.setTitle("My First Java Program");
        myWindow.setVisible(true);
    }
}
```

Program Components

- Ch2Sample1 is composed of 3 parts
 1. Comment
 2. Import Statement
 3. Class Declaration

Program Components

1. Comment: คือส่วนที่ไม่มีผลต่อการทำงานของโปรแกรม นิยมใช้ในการอธิบายการทำงานของโค้ด

```
/*  
    Chapter 2 Sample Program: Displaying a Window  
    File: Ch2Sample1.java  
*/
```

```
import javax.swing.*;
```

```
class Ch2Sample1 {
```

```
    public static void main(String[] args) {
```

```
        JFrame    myWindow;
```

```
        myWindow = new JFrame();
```

```
        myWindow.setSize(300, 200);
```

```
        myWindow.setTitle("My First Java Program");
```

```
        myWindow.setVisible(true);
```

```
    }
```

```
}
```

Comment

Program Components

- Comment มี 3 แบบ
 1. Multi-line Comments: ใช้ `/*...*/`
 2. Single-line Comments: ใช้ `//`
 3. Java Doc Comments

```
/*  
    This is a comment with  
    three lines of  
    text.  
*/  
  
// This is a comment  
// This is another comment  
// This is a third comment
```

Program Components

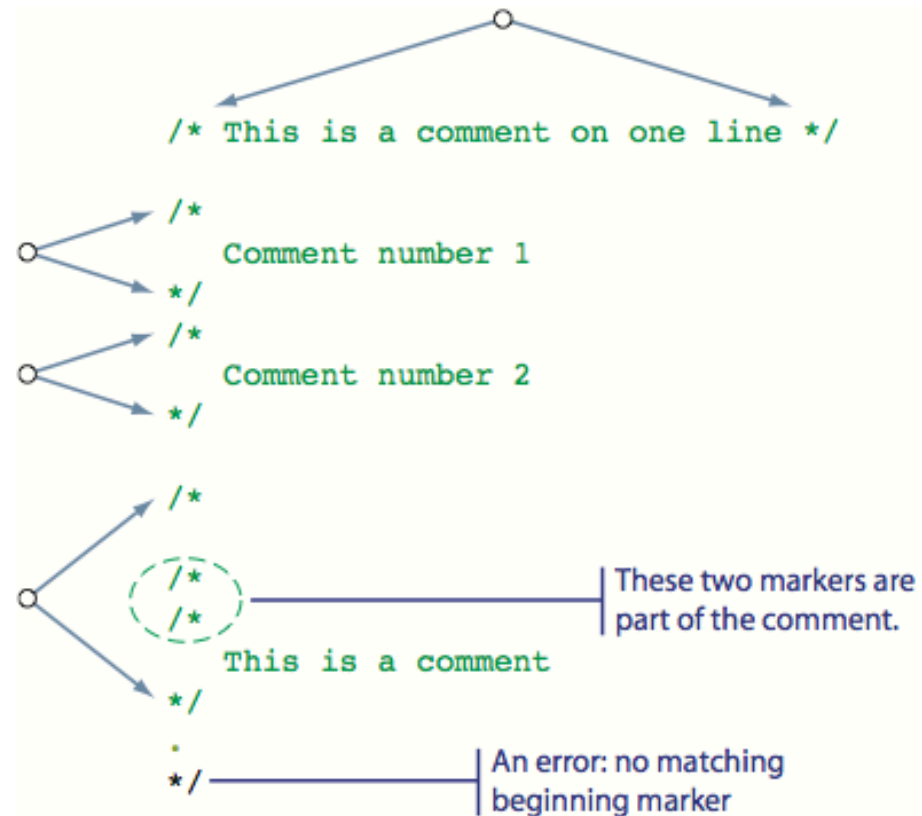


Figure 2.8 How the beginning and ending comment markers are matched.

Program Components

2. Import Statement: ใช้ระบุชื่อ package ของ class ที่เราต้องการนำมาใช้ใน โปรแกรม มีรูปแบบดังนี้
- `import <PACKAGE_NAME>.<CLASS_NAME>;` เพื่อเจาะจง class ที่ต้องการ
 - `import <PACKAGE_NAME>.*;` เพื่อ import class ทั้งหมดใน package
 - Ex. `import javax.swing.*;`

Program Components

- กรณีที่ไม่ใช้ Import Statement: เราต้องใช้ Fully Qualified Name ในการเรียกชื่อ class

```
/*  
    Chapter 2 Sample Program: Displaying a Window  
    File: Ch2Sample1.java  
*/  
  
class Ch2Sample1 {  
    public static void main(String[] args) {  
        javax.swing.JFrame myWindow;  
        myWindow = new javax.swing.JFrame ();  
        myWindow.setSize(300, 200);  
        myWindow.setTitle("My First Java Program");  
        myWindow.setVisible(true);  
    }  
}
```

No import statement

Fully qualified names

Program Components

- กรณีที่ใช้ Import Statement: เราไม่ต้องใช้ Fully Qualified Name ใช้แค่ชื่อ class ในการอ้างอิงถึง

```
/*
```

```
Chapter 2 Sample Program: Displaying a Window
```

```
File: Ch2Sample1.java
```

```
*/
```

```
import javax.swing.*;
```

```
class Ch2Sample1 {
```

```
    public static void main(String[] args) {
```

```
        JFrame    myWindow;
```

```
        myWindow = new JFrame();
```

```
        myWindow.setSize(300, 200);
```

```
        myWindow.setTitle("My First Java Program");
```

```
        myWindow.setVisible(true);
```

```
    }
```

```
}
```

Import Statement

The **import** statement allows the program to refer to classes defined in the designated package without using the fully qualified class name.

Program Components

3. Class Declaration: คือส่วนที่เราเขียนโครงสร้างของ class ว่าประกอบด้วย data (property) และ method อะไรบ้าง มีรูปแบบดังนี้

```
class <CLASS_NAME>{  
    <Data Member Declaration>  
    <Method Declaration>  
}
```

Program Components

- Class Declaration

```
class Ch2Sample1 {  
    public static void main(String[] args) {  
        JFrame    myWindow;  
        myWindow = new JFrame();  
        myWindow.setSize(300, 200);  
        myWindow.setTitle("My First Java Program");  
        myWindow.setVisible(true);  
    }  
}
```

Class Declaration

Every program
must include at
least one class.

Program Components

- ใน Class Declaration จะมีส่วนที่เป็น Method Declaration ที่เราเขียน method ของ class ขึ้นมา

```
import javax.swing.*;
```

```
class Ch2Sample1 {
```

```
    public static void main(String[] args) {
```

```
        JFrame myWindow;
```

```
        myWindow = new JFrame();
```

```
        myWindow.setSize(300, 200);
```

```
        myWindow.setTitle("My First Java Program");
```

```
        myWindow.setVisible(true);
```

```
    }
```

```
}
```

Method Declaration

This declaration declares the **main** method.

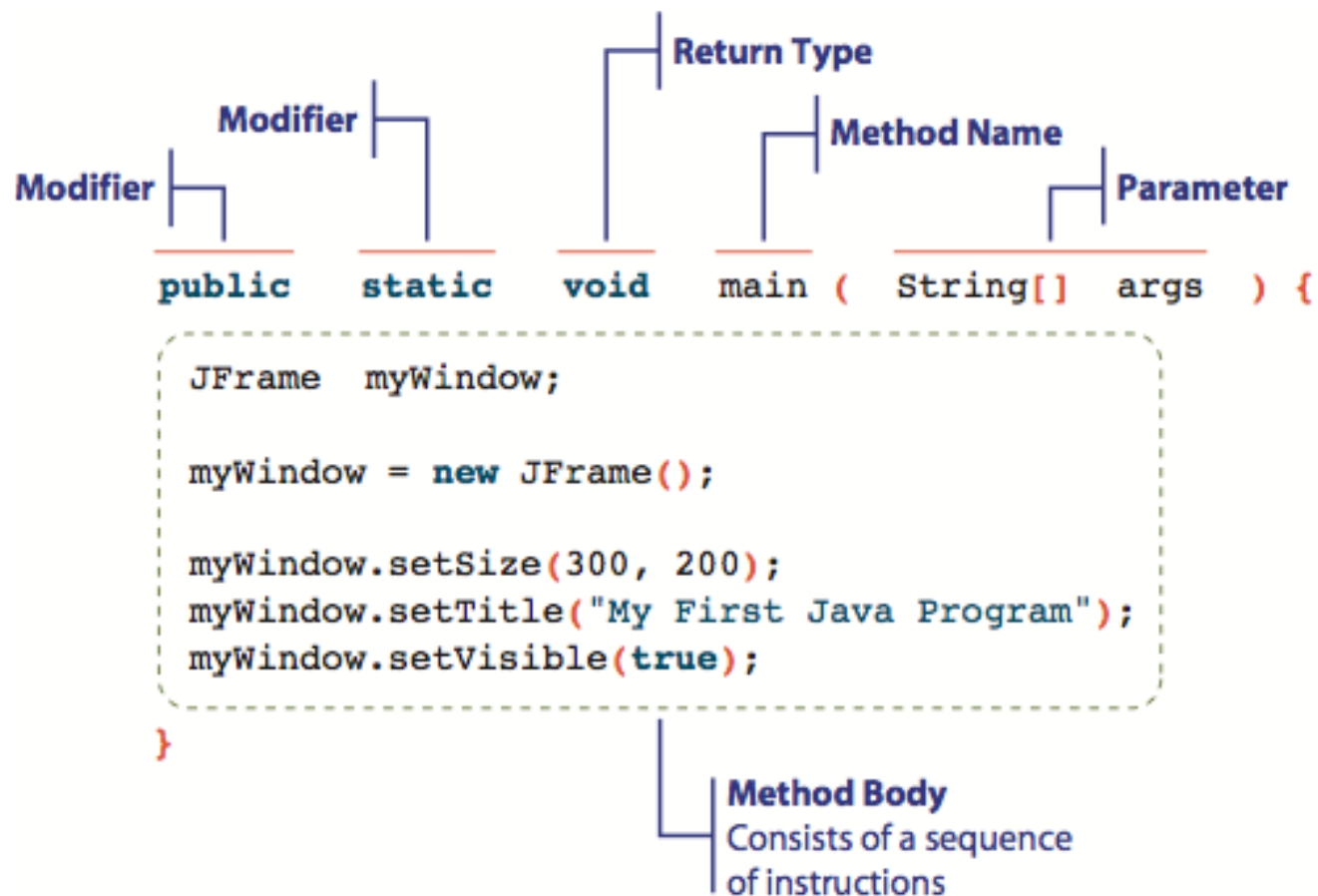
Program Components

- Method Declaration มีรูปแบบดังนี้

```
<Modifiers> <Return Type> <METHOD_NAME>(<Parameters>){  
    <Method Body>  
}
```

Program Components

- Method Declaration



Program Components

- Simple Java Program Template

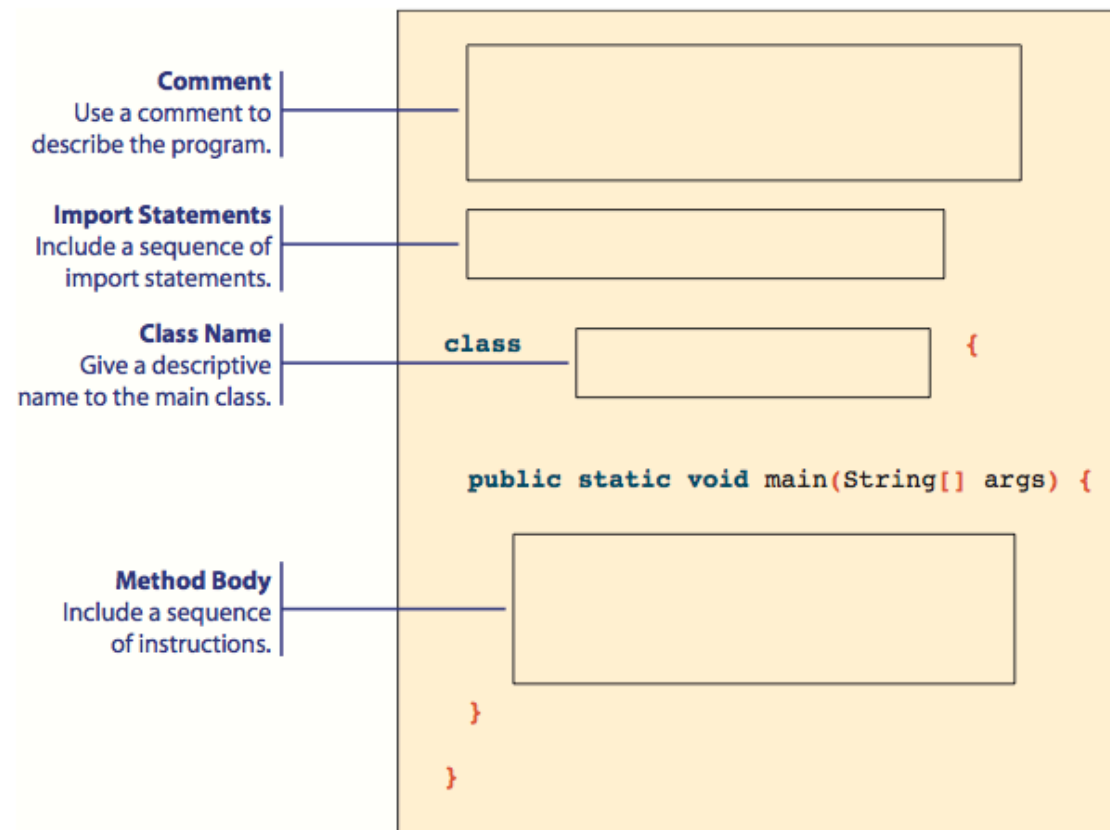
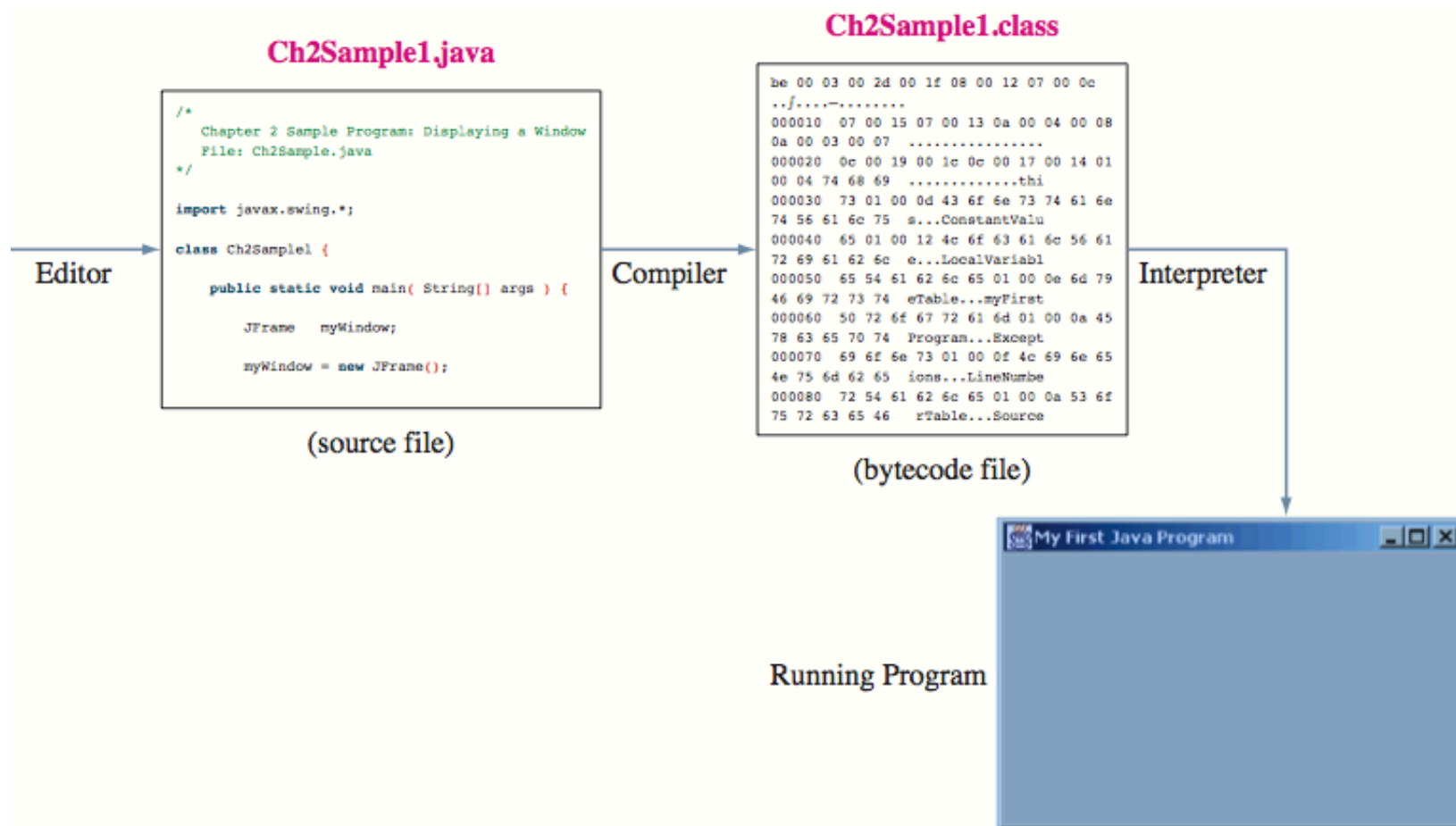


Figure 2.9 A program template for simple Java programs.

Edit-Compile-Run Cycle

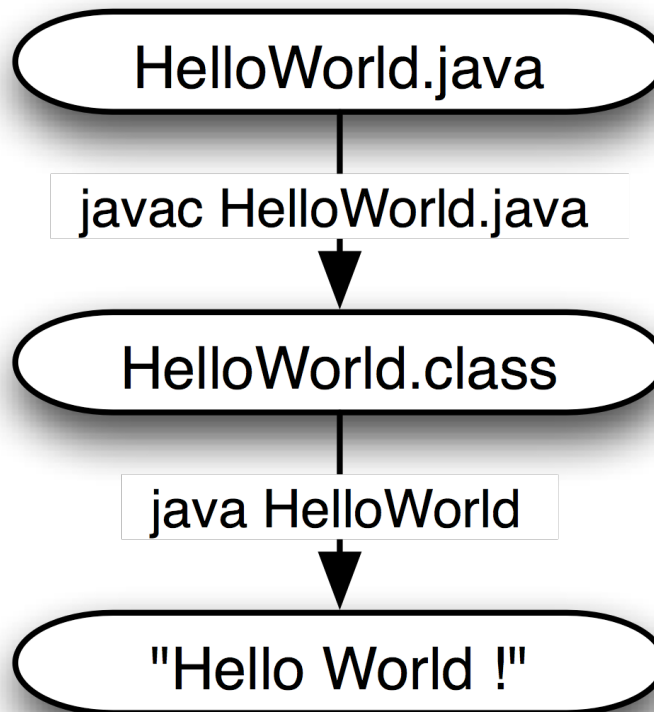
- ขั้นตอนในการสร้าง Java Program
 1. สร้าง source file (.java file)
 2. Compile source file ได้ bytecode file (.class file)
 3. Execute bytecode file

Edit-Compile-Run Cycle



Edit-Compile-Run Cycle

- HelloWorld (Lab1)



Sample Java Standard Classes

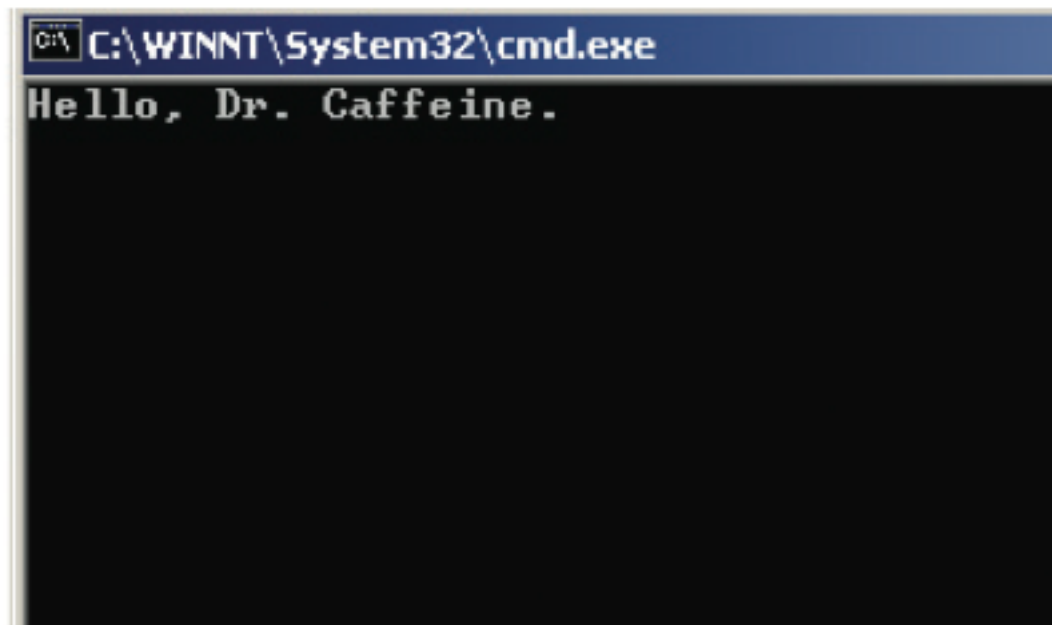
- บทนี้เราจะพูดถึง Java Standard Classes ดังนี้
 1. Standard Output สำหรับใช้แสดงผลบน console
 2. String สำหรับจัดการข้อมูลที่เป็นตัวหนังสือ
 3. Date สำหรับจัดการข้อมูลที่เป็นวัน-เวลา และ SimpleDateFormat สำหรับกำหนดรูปแบบการแสดงผลของ Date
 4. Standard Input สำหรับรับข้อมูลจาก console เข้าสู่โปรแกรม

Sample Java Standard Classes

1. Standard Output: ใช้ในการแสดงผลบน console โดยการเรียกใช้ผ่าน System.out
 - System คือ class ที่รวม class และ method สำหรับใช้งาน และอ่านค่าจาก Operating System
 - out คือ object ของ class PrintStream โดย out จะเป็น data member ของ class System

Sample Java Standard Classes

- Ex. `System.out.print("Hello, Dr. Caffeine.");`



A screenshot of a Windows command prompt window. The title bar at the top reads "C:\WINNT\System32\cmd.exe". The main window area is black with white text. The text "Hello, Dr. Caffeine." is displayed on the first line.

Note

Depending on the tool you use, you may see additional text such as

Press any key to continue...

or something similar to it. We will ignore any text that may be displayed automatically by the system.

Sample Java Standard Classes

- Method ที่สำคัญ
 - print() method: ใช้แสดงผลข้อมูล โดยไม่ขึ้นบรรทัดใหม่

Code

```
System.out.print("How do you do? ");  
System.out.print("My name is ");  
System.out.print("Seattle Slew.");
```

Output

```
How do you do? My name is Seattle Slew.
```

Sample Java Standard Classes

- println() method: ใช้แสดงผลข้อมูลแล้วขึ้นบรรทัดใหม่

Code

```
System.out.println("How do you do? ");  
System.out.println("My name is ");  
System.out.println("Seattle Slew.");
```

Output

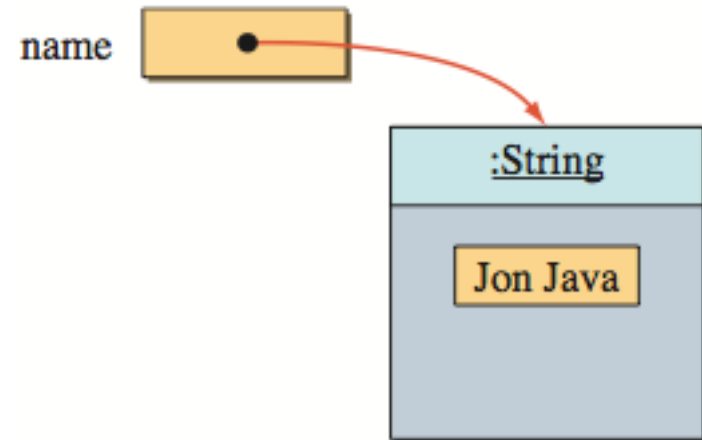
```
How do you do?  
My name is  
Seattle Slew.
```


Sample Java Standard Classes

2. String: เป็น class ที่ใช้เก็บและจัดการข้อมูลที่เป็นกลุ่มของตัวอักษร เป็น class พิเศษ สามารถสร้าง object ของ String ได้ 2 แบบคือ

1. แบบ class ปกติ

```
String name;  
name = new String("Jon Java");
```



Sample Java Standard Classes

2. แบบเฉพาะตัวของ String เท่านั้น

```
String name;  
name = "Decafe Latte";
```

Sample Java Standard Classes

- Method ที่สำคัญ
 - substring() method: ใช้ในการตัด String ย่อยออกมาจาก String หลัก มี parameter 2 ตัว
 - Param 1: the beginning index, inclusive.
 - Param 2: the ending index, exclusive.

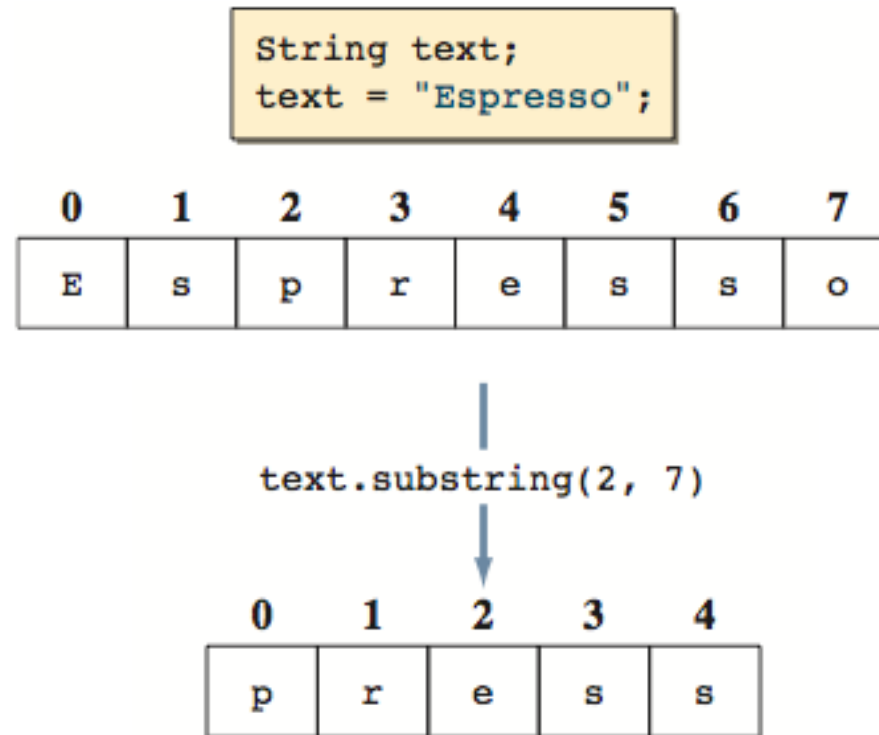
Code

```
String text;  
text = "Espresso";  
System.out.print(text.substring(2,7));
```

Output

press

Sample Java Standard Classes



Sample Java Standard Classes

```
String text;  
text = "Espresso";
```

0	1	2	3	4	5	6	7
E	s	p	r	e	s	s	o

`text.substring(6, 8)` → "so"

`text.substring(0, 8)` → "Espresso"

`text.substring(1, 5)` → "spre"

Sample Java Standard Classes

substring

```
public String substring(int beginIndex,  
                        int endIndex)
```

Returns a string that is a substring of this string. The substring begins at the specified `beginIndex` and extends to the character at index `endIndex` - 1. Thus the length of the substring is `endIndex - beginIndex`.

Examples:

```
"hamburger".substring(4, 8) returns "urge"  
"smiles".substring(1, 5) returns "mile"
```

Parameters:

`beginIndex` - the beginning index, inclusive.

`endIndex` - the ending index, exclusive.

Returns:

the specified substring.

Throws:

`IndexOutOfBoundsException` - if the `beginIndex` is negative, or `endIndex` is larger than the length of this `String` object, or `beginIndex` is larger than `endIndex`.

Sample Java Standard Classes

- length() method: ใช้ในการหาค่าความยาวของ String

```
text1 = "";           //empty string
text2 = "Hello";
text3 = "Java";

text1.length( )       → 0
text2.length( )       → 5
text3.length( )       → 4
```

Sample Java Standard Classes

- indexOf() method: ใช้ในการหาตำแหน่งของ String ย่อยใน String หลัก
 - ถ้ามี String ย่อยใน String หลักจะ return ค่า index ที่พบ
 - ถ้าไม่มี String ย่อยใน String หลักจะ return ค่า -1

The diagram illustrates the results of the `indexOf()` method for the string `text = "I Love Java and Java loves me.";`. Above the string, vertical lines mark the starting indices of the substrings: 3 for "Love", 7 for "Java", and 21 for "loves". Below, four calls to `text.indexOf()` are shown with arrows pointing to their respective return values.

<code>text = "I Love Java and Java loves me.";</code>		
<code>text.indexOf("J")</code>	→	7
<code>text.indexOf("love")</code>	→	21
<code>text.indexOf("ove")</code>	→	3
<code>text.indexOf("ME")</code>	→	-1

Sample Java Standard Classes

- concat() method: ใช้ในเชื่อม String 2 ตัว

```
String s1 = "Hello";  
String s2 = "world";  
String s3 = s1.concat(s2);  
  
System.out.println(s3);
```

```
run:  
HelloWorld
```

Sample Java Standard Classes

ในเชื่อม String 2 ตัวนอกจากจะใช้ concat() method แล้ว เรายังสามารถใช้เครื่องหมาย + แทนได้ด้วย

```
String s1 = "Hello";  
String s2 = "world";  
String s3 = s1 + " " + s2;  
  
System.out.println(s3);
```

```
run:  
Hello World
```

Sample Java Standard Classes

```
text1 = "Jon";  
text2 = "Java";
```

```
text1 + text2           → "JonJava"
```

```
text1 + " " + text2     → "Jon Java"
```

```
"How are you, " + text1 + "?"  
                    → "How are you, Jon?"
```

Sample Java Standard Classes

3. Date และ SimpleDateFormat:

- Date: เป็น class ใน package java.util ใช้เก็บค่า
เวลาขณะที่สร้าง object จาก class Date หรือเวลา
อื่นๆ ที่กำหนด เป็น millisecond นับจาก January 1,
1970, 00:00:00 GMT

```
Date today;  
  
today = new Date( );  
System.out.println(today.toString());
```

will display the current time in this format:

```
Thu Dec 18 18:16:56 PST 2008
```

Sample Java Standard Classes

- SimpleDateFormat: เป็น class ที่ใช้ในการกำหนดรูปแบบการแสดงผลให้กับ object ของ class Date มี Method ที่สำคัญคือ
 - format() method: ใช้สั่งให้ object ของ class Date แสดงผลตามรูปแบบที่กำหนด

Sample Java Standard Classes

```
Date          today;  
SimpleDateFormat sdf;  
  
today = new Date( );  
sdf    = new SimpleDateFormat("MM/dd/yy");  
  
System.out.println(sdf.format(today));
```

If today is June 28, 2008, the code will display the date as

06/28/08

Sample Java Standard Classes

```
sdf = new SimpleDateFormat("MMM dd, yyyy");
```

the dialog will display

```
June 28, 2008
```

Sample Java Standard Classes

```
Date          today;  
SimpleDateFormat sdf;  
  
today = new Date( );  
sdf    = new SimpleDateFormat("EEEE");  
  
System.out.println("Today is " + sdf.format(today));
```

```
run:  
Tuesday
```


Sample Java Standard Classes

- สัญลักษณ์รูปแบบการแสดงผลของ SimpleDateFormat

Symbol	Meaning	Value	Sample
y	Year	Number	yyyy → 2008
M	Month in year	Text or number	MM → 10 MMM → Oct MMMM → October
d	Day in month	Number	dd → 20
D	Day in year	Number	DDD → 289
h	Hour in AM/PM	Number	hh → 09
H	Hour in day (0-23)	Number	HH → 17
a	AM/PM marker	Text	a → AM
m	Minutes in hour	Number	mm → 35
s	Seconds in minute	Number	ss → 54
S	Millisecond	Number	mmm → 897
E	Day in week	Text	E → Sat EEEE → Saturday

Sample Java Standard Classes

4. Standard Input: ใช้ในการรับค่าจาก console โดยการเรียกใช้ `System.in` ผ่านทาง object ของ class `Scanner`
- `System` คือ class ที่รวม class และ method สำหรับใช้งาน และอ่านค่าจาก Operating System
 - `in` คือ object ของ class `InputStream` โดย `in` จะเป็น data member ของ class `System`
 - `Scanner` คือ class ที่ช่วยในการอ่านค่าจาก `System.in`

Sample Java Standard Classes

- Method ที่สำคัญของ class Scanner
 - next() method: ใช้ในการอ่านค่าที่ input มา 1 คำ

```
Scanner scanner = new Scanner(System.in);  
String firstName;  
  
//prompt the user for input  
System.out.print("Enter your first name: ");  
firstName = scanner.next( );  
System.out.println("Nice to meet you, " + firstName + ".");
```

Sample Java Standard Classes

```
System.out.print("Enter your first name: ");  
firstName = scanner.next( );  
System.out.println("Nice to meet you, " + firstName + ".");
```

Enter your first name: George ENTER
Nice to meet you, George.

This is entered by the user and echo printed by the system. To distinguish the input and output in the diagram, the input is displayed in a different color.

This icon shows the pressing of the Enter (Return) key.

Sample Java Standard Classes

```
Scanner scanner = new Scanner(System.in);

String firstName, lastName;

System.out.print("Enter your first name: ");
firstName = scanner.next( );

System.out.print("Enter your last name: ");
lastName = scanner.next( );

System.out.println("Your name is " + firstName +
                  " " + lastName + ".");
```

```
Enter your first name: George 
Enter your last name: Washington 
Your name is George Washington.
```

Sample Java Standard Classes

- โดยปกติแล้ว Scanner จะใช้ space เป็น delimiter (ตัวตัดคำ) ทุกครั้งที่เรียกใช้ next() จะส่งค่าแต่ละคำที่ตัดโดยใช้ space กลับไป

```
Scanner scanner = new Scanner(System.in);  
  
String first, last;  
  
System.out.print("Enter your first and last name: ");  
first = scanner.next( );  
last  = scanner.next( );  
  
System.out.println("Your name is " + first + " "  
                  + last + ".");
```

```
Enter your first and last name: George Washington ENTER  
Your name is George Washington.
```

Sample Java Standard Classes

- หากต้องการอ่านค่าประโยคยาวๆ ที่มี space อยู่ในประโยคมาแสดงผล เราต้องเปลี่ยนค่า delimiter (ตัวตัดคำ) ของ Scanner

```
String lineSeparator = System.getProperty("line.separator");
```

```
Scanner scanner = new Scanner(System.in);  
scanner.useDelimiter(lineSeparator);
```

```
System.out.print("Enter your favorite quote: ");
```

```
String quote = scanner.next( );
```

```
System.out.println("You entered: " + quote);
```

run:

```
Enter your favorite quote: This is just a test  
You entered: This is just a test
```

Summary

- The three basic components of a Java program are comments, import statements, and class declarations.
- A Java program must have one class designated as the main class. The designated main class must have the main method.
- An object must be declared and created before we can use it.

Summary

- To command an object or a class to perform a task, we send a message to it. We use the expression calling a method synonymously with sending a message.
- We follow the edit-compile-run cycle to execute programs.
- A source file is compiled into a bytecode file by a Java compiler.

Summary

- A Java interpreter (also called a Java Virtual Machine) executes the bytecode.
- The standard classes introduced (mentioned) in this chapter are
 - Standard Output (System.out : PrintStream)
 - String
 - Date and SimpleDateFormat
 - Standard Input (Scanner and System.in : InputStream)

Reference

- C. Thomas Wu, An Introduction to Object-Oriented Programming with Java, 5th Edition
 - Chapter 2: Getting Started with Java

Question?