

บทที่ 1 แนะนำ

1.1 ประวัติ

1.2 characteristic of Java Programming language

1.3 **overview of** software development process

1.4 The Java Platform

1.5 ขั้นตอนการเขียนโปรแกรม

1.6 การคอมไพล์และรันโปรแกรมเบื้องต้น

1.7 ออบเจค และ คลาส

1.8 เริ่มต้นเขียน application กับจาวา

1.1 ประวัติความเป็นมา

บ.Sun Microsystems ประสบความสำเร็จทางด้านเครือข่ายคอมพิวเตอร์ ต้องการขยายตลาดด้านเครื่องใช้ไฟฟ้าที่มีคอมพิวเตอร์ควบคุมสินค้าอิเล็กทรอนิกส์

ก่อนที่จะเป็น จาวา

ค.ศ. 1990 ก่อตั้ง Green group มี james glosling เป็นหัวหน้าทีม

พัฒนาโปรแกรมเพื่อควบคุมการทำงานของเครื่องใช้ไฟฟ้าให้สามารถเป็นระบบอัจฉริยะ (smart consumer electronics)

สร้างเครื่องต้นแบบชื่อ star7 ควบคุมด้วยรีโมท มีขนาดเท่ามือถือเพื่อควบคุมให้ The Duke เคลื่อนที่ผ่านวัตถุในมิติสมมุติ

ในเวลาต่อมาเดอะดุก เป็นแมสคอตของภาษาจาวา

พัฒนาโดยใช้ภาษา c++ มีปัญหา

- แก้ไขยาก
- ใช้งานแล้วมีปัญหา โปรแกรมมีขนาดใหญ่เครื่องใช้ไฟฟ้ามีขนาดเล็ก เบอร์หน่วยประมวลผลมีหลายเบอร์ หลายยี่ห้อ มีชุดคำสั่งไม่เหมือนกัน
- การจัดการ resource ผู้เขียนโปรแกรมต้องจัดการเอง

คิดภาษาใหม่

ภาษา oak คล้าย ภ. c++ แก้ข้อบกพร่องของ c++

1994 internet ได้รับความนิยม

บ.sun พัฒนา oak ให้สนับสนุนงานด้าน internet

ชื่อนี้ถูกจดเครื่องหมายการค้าไปแล้ว

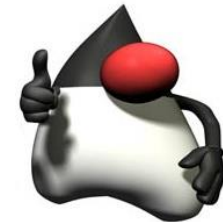
ทีมงานต้องการชื่อที่บ่งบอกถึงเทคโนโลยีที่มีการเคลื่อนไหวไม่หยุดนิ่ง

มีความเร็ว มีชีวิตชีวา

ชื่อ java เป็นพันธุ์กาแฟ กลิ่นหอมหวาน Programmer ชอบดื่ม

บริษัท Sun นำโปรแกรมชุดพัฒนาภาษา Java แจกจ่ายให้สามารถ download ได้ฟรี

ค.ศ.2010 บ.oracle เข้ามาซื้อกิจการ



ในการพัฒนาสมัยแรกๆ ได้สร้าง web browser ชื่อว่าเว็บรันเนอร์ (WebRunner) ต่อมาเว็บเบราว์เซอร์ดังกล่าวเปลี่ยนชื่อเป็นฮอตจาวา (Hotjava)

นอกจากนี้ภาษาจาวาได้รับความนิยมมีการตั้งชื่อคล้ายกับภาษาจาวา ที่เราต้องทราบคือ

Java / JavaScript

- ชื่อคล้ายเท่านั้น
- Java เป็น ภาษาโปรแกรม

JavaScript สร้างโดยบริษัท **Netscape Corporation**

เดิมชื่อ **LiveScript** แต่เนื่องจาก **Java** ได้รับความนิยม

ชื่อคล้ายง่าย เป็นเหตุผลทางธุรกิจ

JavaScript เป็น **script** คือกลุ่มของคำสั่ง/โปรแกรมสั้น ๆ ที่ใช้

กำหนดหรือควบคุมเว็บเพจเพื่อสร้าง **active html**

JavaScript is the programming language of HTML and the Web.

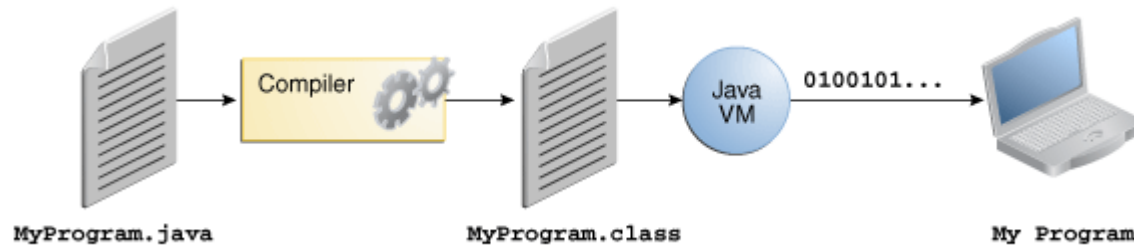
1.2 characteristic of Java Programming language

- Simple
- Object oriented
- Distributed network client-server
- Robust :reliable software : *automatic garbage collector*
- Secure applications written in the Java programming language are secure from intrusion by unauthorized code attempting to get behind the scenes and create viruses or invade file systems.
- Multi threaded many concurrent thread
- Dynamic run program
- Architecture neutral
- Portable
- High performance :automatic garbage collector

JVM(java virtual machine)



1.3 overview of software development process



โปรแกรมภาษา java ถูกคอมไพล์โดย java compiler

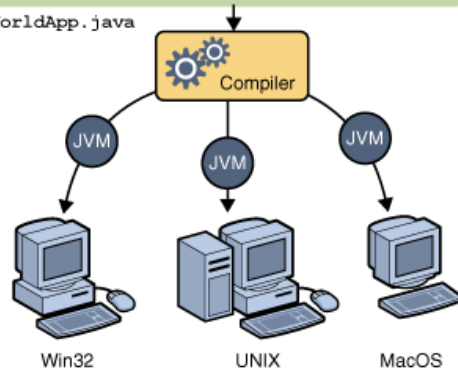
ได้ ไฟล์ MyProgram.class ซึ่งเป็น bytecode (the machine language of the **Java Virtual Machine :Java VM :JVM**)

จากนั้น JavaVM แปลง bytecode ให้เป็นภาษาเครื่องที่เหมาะสม กับระบบปฏิบัติการ เพื่อทำงานบนเครื่องใดๆ แต่เครื่องนั้นต้องมี java interpreter

Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java



Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris™ Operating System (Solaris OS), Linux, or Mac OS.

Platform independence
write once run anywhere

the Java VM, the same application is capable of running on multiple platforms.

1.4 Java Platform

ทำความเข้าใจกับคำว่า platform

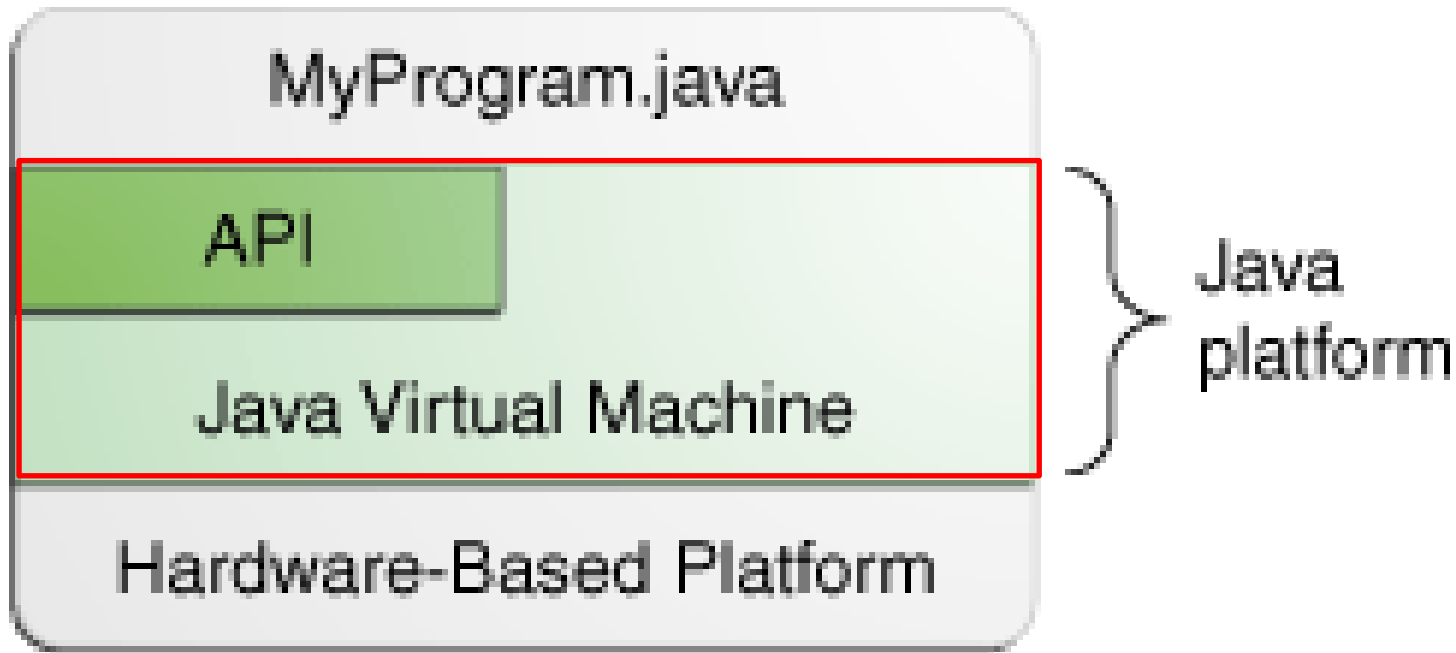
- **แพลตฟอร์ม (Platform)** หมายถึง สภาพแวดล้อมของ **hardware** หรือ **software** ที่โปรแกรมใช้ในการรัน (Execute / ทำงาน)
- ตัวอย่างของแพลตฟอร์ม เช่น
 - Microsoft Window, Linux, Solaris ที่ติดตั้งบน IBM compatible PC
 - Mac OS ที่ติดตั้งบนเครื่อง Mac เป็นต้น
- โดยแพลตฟอร์มส่วนใหญ่เป็นการรวมกันของระบบปฏิบัติการ (operating system) กับฮาร์ดแวร์ (hardware)
- แต่ Java Platform คือ **Software-only platform**

platform is the hardware or software **environment** in which a program runs
popular platforms like Microsoft Windows, Linux, Solaris OS, and Mac OS

จาวาเป็นแพลตฟอร์ม

- เนื่องจากจาวาได้ถูกพัฒนาเพื่อให้สามารถรันบนระบบใดก็ได้
ดังนั้นจาวาจึงได้สร้างแพลตฟอร์ม
เพื่อให้การแปลความของภาษาจาวาเป็นหนึ่งเดียว
virtual machine
- จาวาได้นำแนวคิดการจำลองเครื่องจักรสมมติขึ้นมา
เรียกว่า **java virtual machine (JVM)** เพื่อมา
ติดต่อกับระบบที่มีอยู่

Java Platform



The Java platform has two components:

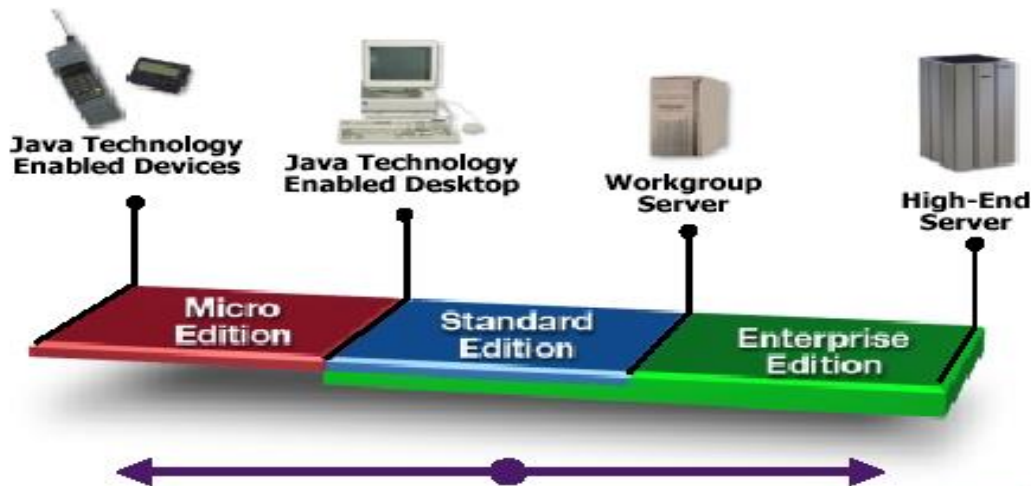
The *Java Virtual Machine*

The *Java Application Programming Interface (API)*

- **Java Virtual Machine**; it's the base for the Java platform and is ported onto various hardware-based platforms.
- The **API** is a large collection of ready-made **software components** that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as **packages**.

Java Platform

- จาว่าถูกพัฒนามาเพื่อให้รันได้โดยไม่ขึ้นกับระบบปฏิบัติการและฮาร์ดแวร์
- แต่เนื่องจากระบบต่างๆ เหล่านี้มีจุดประสงค์ในการใช้งานแตกต่างกัน จึงได้ถูกออกแบบมาไม่เหมือนกัน
- ทางบริษัทซันไมโครซิสเต็มจึงแบ่งจาว่าแพลตฟอร์มออกเป็น 3 รุ่นเพื่อประโยชน์ในการใช้งานโปรแกรมจาว่าในแต่ละอุปกรณ์ให้มีประสิทธิภาพที่สุด นั่นคือ
 - **Java 2 Platform Standard Edition (J2SE)**
 - **Java 2 Platform Enterprise Edition (J2EE)**
 - **Java 2 Platform Micro Edition (J2ME)**



1.5 ขั้นตอนการเขียนโปรแกรม

1) ในการพัฒนางานด้วยจาวาจะต้องมี**เครื่องมือ**ช่วยซึ่งพัฒนาโดย **Sun Microsystems** เรียกว่า **Java Platform , Standard Edition** ที่สามารถ **download** ได้ฟรีที่

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Java Platform , Standard Edition เป็นชุดโปรแกรมที่ใช้สำหรับสร้างและพัฒนา โปรแกรมภาษาจาวา

ภายในชุดประกอบด้วยโปรแกรม **JDK (Java Development Kit)**
JRE (Java Runtime Environment)

หรือ อาจจะใช้ **third-party Java IDE(Integrated Development Environment)** ที่มีขาย/opensource เช่น **JBuilder, Visual Cafe** หรือ **Visual Age for Java , NetBeans , Eclipse IDE for Java Developers**

ซึ่ง **IDE** เหล่านี้ จะรวมเครื่องมือในการเขียนและพัฒนาโปรแกรมไว้ในรูปของ **GUI** ทำให้ไม่จำเป็นต้องใช้ **Text Editor**

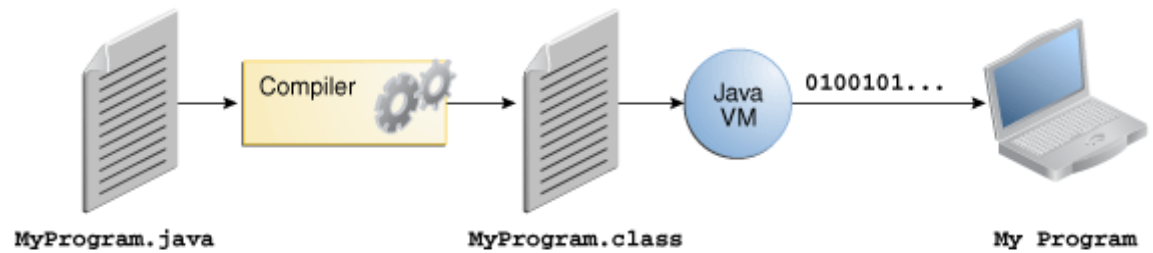
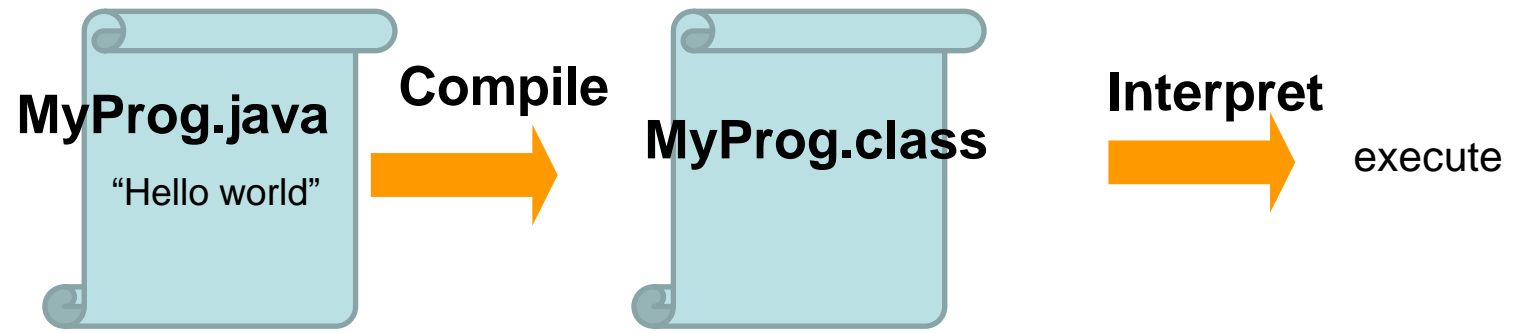
หากไม่ใช้ third-party Java IDE(Integrated Development Environment)

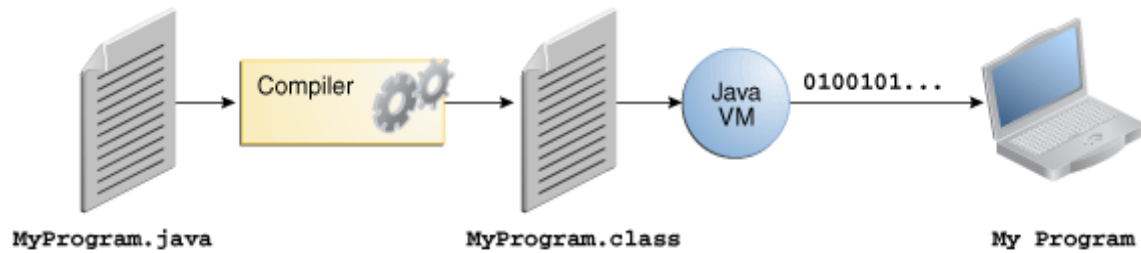
2) เขียนโปรแกรมด้วย Text Editor

เช่น Notepad , WinEdit, Edit Plus

เมื่อใช้ text editor เขียนโปรแกรมต้องกำหนดนามสกุลของ source code เป็น .java แล้ว compile จะได้ไฟล์ bytecode ที่มีนามสกุลเป็น .class

สั่ง run โปรแกรมจากไฟล์ bytecode โดยอาศัย JVM ในส่วนของ execution engine

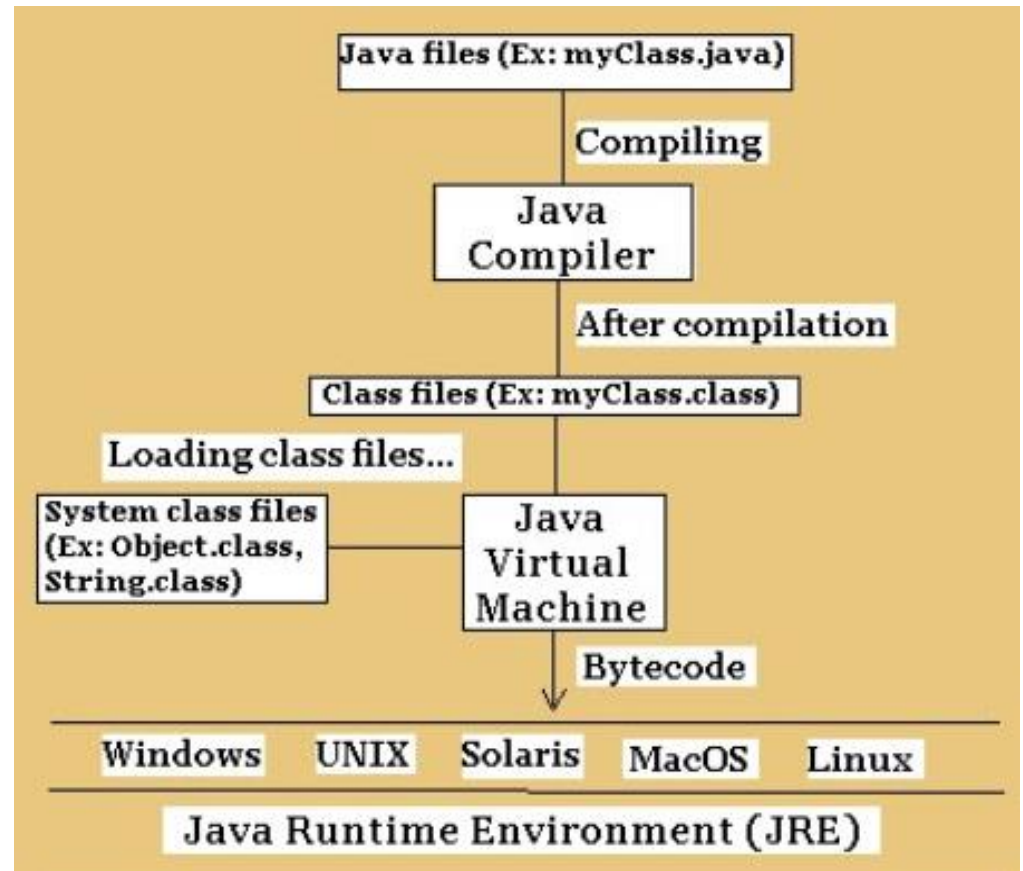
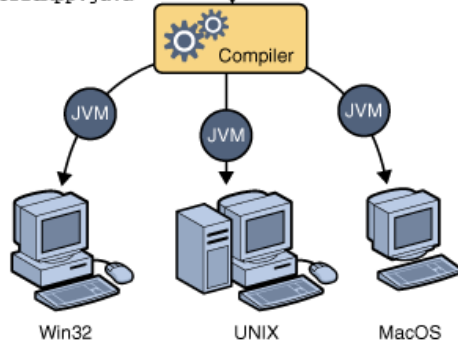




Java Program

```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

HelloWorldApp.java



1.6 การคอมไพล์และรันโปรแกรมภาษาจาวา

รูปแบบของโปรแกรมทำได้ 2 ลักษณะคือ

1) Java applications เป็นโปรแกรมทั่วไปที่เขียนด้วยภาษาจาวา สามารถทำงานได้กับทุก platform โดยไม่ต้องอาศัย Browser

2) Java applets เป็นโปรแกรมที่เขียนขึ้นมาเพื่อฝังตัว(embedded) ไว้ใน Web page เพื่อใช้สำหรับอินเทอร์เน็ต

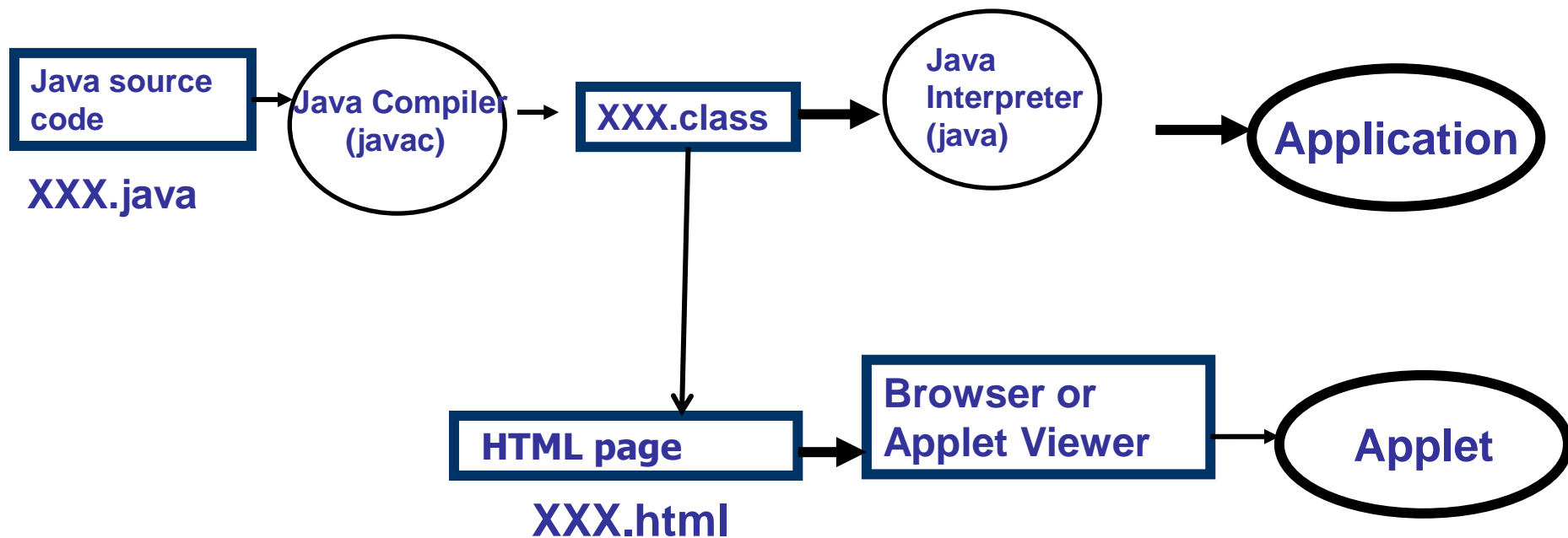
การคอมไพล์และรันของโปรแกรม

กรณี applications

- 1) สร้าง **source code** กำหนดนามสกุลเป็น **.java**
- 2) คอมไพล์โปรแกรม
ใช้คำสั่ง **javac** ชื่อไฟล์.java เมื่อผ่านการคอมไพล์จะได้
ไฟล์ นามสกุล **.class** ซึ่งเป็น **byte code**
- 3) run โปรแกรม
ใช้คำสั่ง **java** ชื่อไฟล์

กรณี applets

- 1) สร้าง **source code** กำหนดนามสกุลเป็น **.java**
- 2) คอมไพล์โปรแกรม
ใช้คำสั่ง **javac** ชื่อไฟล์.java เมื่อผ่านการคอมไพล์จะได้
ไฟล์นามสกุล **.class** ซึ่งเป็น **byte code**
- 3) สร้างเอกสาร **HTML** ขึ้นมา
- 4) นำไฟล์ในข้อ 2 คือ ชื่อไฟล์.class พิมพ์ชื่อไฟล์.classลงในเอกสาร **HTML**
- 5) เรียกดูผลลัพธ์ผ่านทาง **Browser** หรือ **AppletViewer**



รูปแสดงการคอมไพล์และรันของ Applications และ Applets

สิ่งที่ควรคำนึงเมื่อคอมไพล์โปรแกรม

การพิมพ์ชื่อของโปรแกรมต้องพิมพ์ตัวอักษรตัวใหญ่ตัวเล็กให้ถูกต้องเสมอ

ชื่อคลาส ตัวอักษรตัวแรกควรใช้ตัวอักษรพิมพ์ใหญ่

ชื่อตัวแปร ชื่อmethod ชื่อobject ตัวอักษรตัวแรกควรใช้ตัวอักษรพิมพ์เล็ก

1.7 Object & class

object คือ วัตถุ ในชีวิตประจำวันของเรา

วัตถุมี่ทั้งจับต้องได้ เป็น**รูปธรรม** เช่น โต๊ะ เก้าอี้ รถยนต์

เป็น**นามธรรม** ไม่สามารถจับต้องได้ เช่น ความรู้ วินัย คุณธรรม ความดี เป็นต้น

ดังนั้น **object** เป็นอะไรก็ได้เป็นได้ทั้งรูปธรรมและนามธรรม

Object

An abstraction of something in a problem domain, reflecting the capabilities of the system to keep information about it, interact with it, or both

object จะต้องประกอบด้วย

คุณสมบัติ (attribute) และพฤติกรรม (behavior)

+identity คือระบุได้ว่าคนนั้น ตัวนั้น สิ่งนั้น

คุณสมบัติ อะไรที่เราอยากรู้ อยากจดจำ (thing that the **class is**)

พฤติกรรม สิ่งที่เรามองเห็นเป็นพฤติกรรมจริงๆ (things that the **class does**)

ครูอาจใช้คำว่า คุณลักษณะ หรือ คุณสมบัติ หรือ attribute / property

ใช้คำว่า พฤติกรรม(behavior) หรือ เมธอด (method) หรือ service

ตัวอย่าง attribute&behavior object

cow

attribute

Name Weight Breed Age

Behavior

Sleep Eat Move Give milk

Employee

attribute

Name Date of birth Employee number

Salary Date started Date finish

Behavior

Start work Finish work Take leave Quit

คน

คุณสมบัติ

หมายเลขบัตรประชาชน เพศ วันเดือนปีเกิด

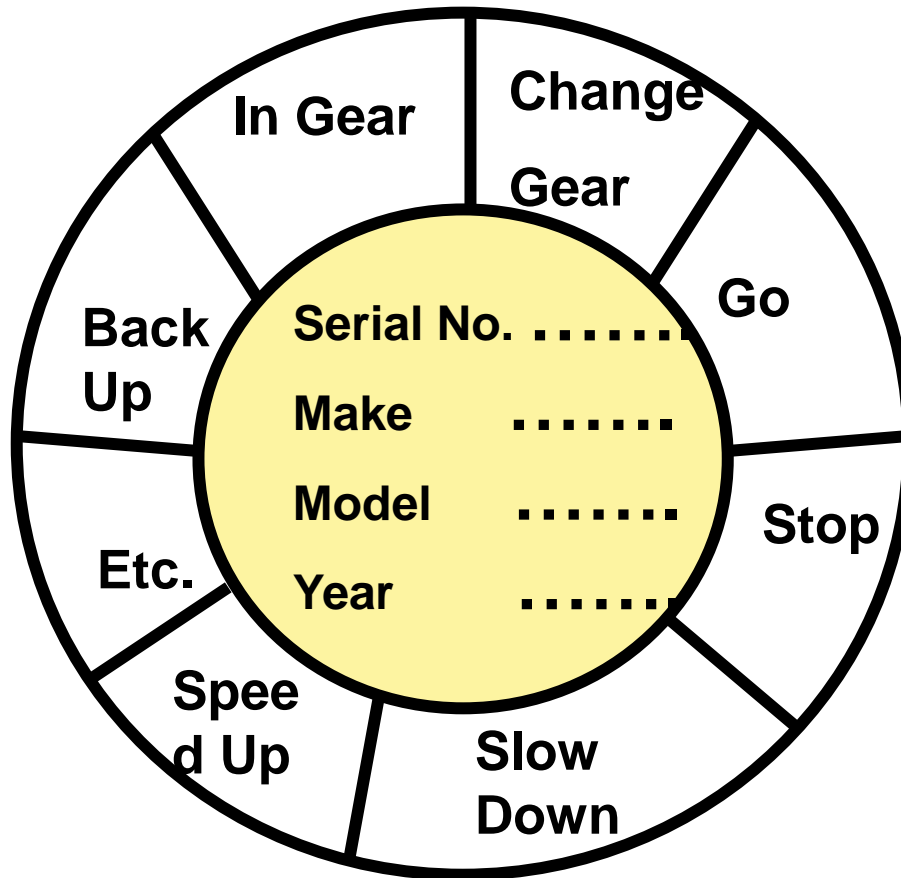
พฤติกรรม

กิน เดิน ยืน นั่ง หายใจ

ตัวอย่าง

class Car

object-oriented concepts
[abstraction](#) [encapsulation](#) [message](#)
[classification](#) [inheritance](#) [reuse](#)



Attribute and behavior

ลักษณะประจำของ class Customer (attribute)

CustomerID

First name

Last name

Home hone

Work phone

Object :

5661234

David

Jangtan

0881235511

05312345



Object :

5661255

Anna

Cangan

0716477789

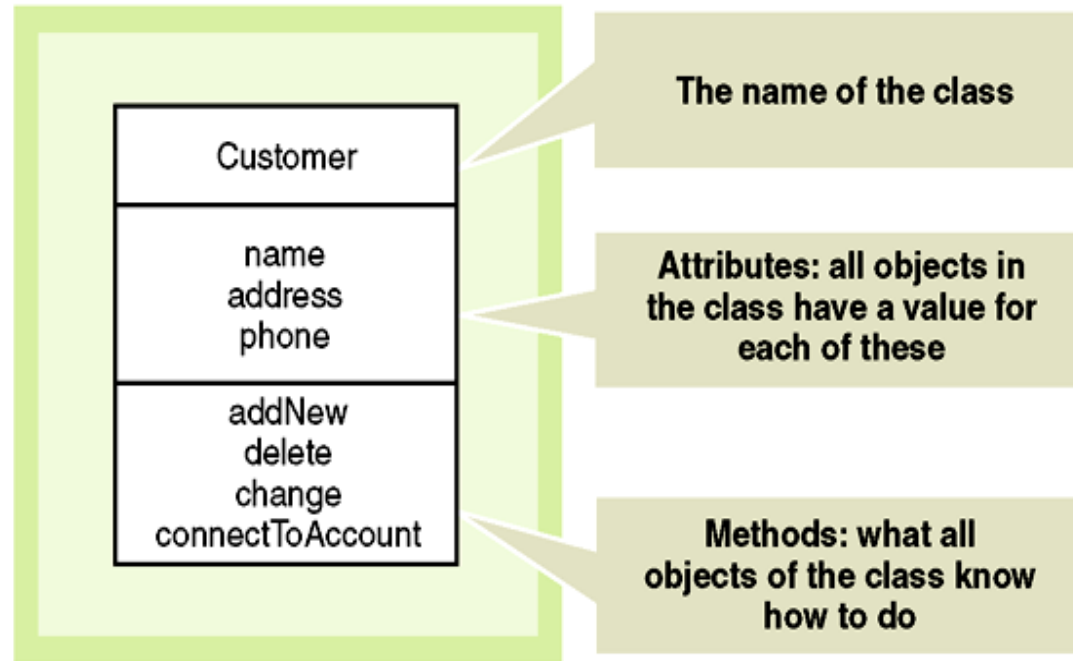
053787811



ตัวอย่าง class Customer

FIGURE

The class symbol with three sections for name, attributes, and methods.



จงยกตัวอย่าง Class
หา Attribute
หา Behavior

-real world นักศึกษา
-- attribute มีรหัส ชื่อ
-- behavior นั่งเรียน กลับ อ่านหนังสือ

-program s/w ??

1.8 เริ่มต้นเขียน application กับจาวา

- 1) รูปแบบการเขียน application
- 2) ดูเขาเขียน
- 3) ลองเขียนเอง แบบง่าย ๆ ก่อน

รูปแบบการเขียน application

application คือ โปรแกรมที่สามารถทำงานได้
โดยไม่ต้องการโปรแกรมอื่นใดมาทำการสั่งงาน

```
class   ชื่อคลาส
{
    การประกาศคุณสมบัติ
    การประกาศเมธอด
}
```



```
//
```

comment คำอธิบายโปรแกรม

```
import ...;
```

คำสั่ง import

```
class ชื่อคลาส  
{  
    การประกาศคุณสมบัติ  
    การประกาศเมธอด  
  
    {  
    }  
  
}
```

→ คลาส

ชื่อคลาส ตัวอักษรตัวแรกควรใช้ตัวอักษรพิมพ์ใหญ่

ชื่อตัวแปร ชื่อmethod ชื่อobject ตัวอักษรตัวแรกควรใช้ตัวอักษรพิมพ์เล็ก

คำอธิบายโปรแกรม

คือ **comment** เป็นข้อความที่เราเขียนเพื่อใช้อธิบายการทำงานของโปรแกรม หรือใช้อธิบายชื่อผู้เขียนโปรแกรม วันเดือนปีที่เขียน หรือที่แก้ไขล่าสุด

comment มี 3 รูปแบบ

1. `/* */` ใช้สำหรับเขียนคำอธิบายโปรแกรมมากกว่าหนึ่งบรรทัด
2. `//` ใช้สำหรับเขียนคำอธิบายโปรแกรมหนึ่งบรรทัด
3. `/** */` หรือ `**/`

comment จะถูกโปรแกรม `javadoc.exe` นำไปสร้างไฟล์ `html` เป็น `doc` ของโปรแกรม

คำสั่ง import เราใช้เพื่อเรียกใช้งานคลาสต่างๆ ซึ่งภาษาจาวา จัดเก็บ
คลาส รวมไว้ที่ แพคเกจ (package)

java มี packages ต่างๆ อยู่มากมายให้เลือกใช้งาน เช่น
java.applet java.swing java.lang

ตัวอย่าง import java.lang.*; เรียกใช้ package ชื่อ java.lang
โดยเรียกใช้ทุกคลาสที่อยู่ใน package นี้

รูปแบบการเขียนคำสั่ง import

```
import ชื่อแพคเกจ.*;
```

```
import ชื่อแพคเกจ. ชื่อแพคเกจย่อย.*;
```

```
import ชื่อแพคเกจ. ชื่อคลาส;
```

```
import ชื่อแพคเกจ. ชื่อแพคเกจย่อย. ชื่อคลาส ;
```

Java มี package หลักๆ ที่ใช้เขียนโปรแกรมทั่วไป เรียกว่า **standard package** ดังนี้

java.lang : เก็บ class พื้นฐานในการสร้างโปรแกรมจาวา

java.io : เกี่ยวกับการจัดการ input/output

java.util : ประกอบด้วย class ที่ใช้งานทั่ว ๆ ไป

java.awt : ใช้สำหรับวาดภาพกราฟฟิก

java.net : ประกอบด้วย class สำหรับติดต่อทางเครือข่าย

java.applet : ประกอบด้วย class สำหรับสร้างโปรแกรมที่ทำงานบน browser

javax.swing : ประกอบด้วย class สำหรับช่วยสร้าง swing GUI สร้างกราฟฟิกสมัยใหม่

<http://docs.oracle.com/javase/6/docs/api/index.html>

การประกาศ method main

ทบทวนความรู้ function
การส่งค่าให้ function
ผลลัพธ์ที่ได้เมื่อfunctionทำงานเสร็จแล้ว
ex. void testCal();
 void testCal(1,2);
 int testCal(1,2);

```
public class ชื่อคลาส
```

```
{
```

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

```
    {
```

```
        ....ส่วนของโปรแกรม....
```

```
    }
```

```
}
```



ในส่วนนี้เราจะเริ่มจาก 1 class และ 1 method คือ method main
method main() จะเป็น method หลักที่ใช้ในการรันโปรแกรม
ดังนั้นการกระทำต่าง ๆ ที่ต้องการให้เกิดขึ้นในการรันโปรแกรม
จะต้องทำการเขียนคำสั่งไว้ใน method นี้

ดูโปรแกรมตัวอย่าง

```
import java.lang.*;
public class MyFirstApp
{
    public static void main( String []args)
    {
        system.out.print("Hello World");
    }
}
```

Package java.lang import อัตโนมัติ ไม่ต้องใส่

ชื่อคลาส ตัวอักษรตัวแรกควรใช้ตัวอักษรพิมพ์ใหญ่

ชื่อตัวแปร ชื่อmethod ชื่อobject ตัวอักษรตัวแรกควรใช้ตัวอักษรพิมพ์เล็ก

ภาษาจาวาจบประโยคคำสั่งด้วยเครื่องหมาย ;

การแสดงผลทางจอภาพ(Console Output)

ใช้ method ชื่อ "println" ซึ่งอยู่ใน คลาส System.out คำสั่งนี้จะนำข้อมูลที่เป็น String เพื่อนำมาแสดงผลทางจอภาพ

println พิมพ์ข้อความแสดงผลที่จอภาพแล้วขึ้นบรรทัดใหม่

ตัวอย่าง `System.out.println("Object-oriented");`

ตัวอย่าง `System.out.print("Object-oriented");`

```
----- Java Run -----  
Object-oriented  
Normal Termination  
Output completed (0 sec consumed).
```

```
----- Java Run -----  
Object-oriented  
Normal Termination  
Output completed (0 sec consumed).
```

ตัวอย่าง การเขียนโปรแกรม MyDog.java

```
public class MyDog
```

```
{
```

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

```
    {
```

```
        System.out.println("My dog's name is abc.");
```

```
        System.out.println("abc is my dog.");
```

```
    }
```

```
}
```

```
----- Java Run -----
```

```
My dog's name is abc.
```

```
abc is my dog.
```

```
Normal Termination
```

```
Output completed (0 sec consumed).
```


การแสดงผลที่จอภาพ

โดยใช้เครื่องหมาย +

ซึ่งข้อมูลทั้งหมดจะถูกแปลงให้เป็น string โดยอัตโนมัติ
เช่น

```
System.out.print("message"+ "message");
```

```
System.out.println("Specify what to do,"  
+ "not how to do it");
```

```
System.out.println("Specify what to do," +  
"not how to do it");
```

```

public class Test1
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to Java " + " and
                            Java is fun " );
    }
}

```

why error?

```

----- Java compile -----
Test1.java:5: unclosed string literal
                System.out.println("Welcome to Java " +           "and
                                   ^
Test1.java:6: unclosed string literal
                Java is fun " );
                ^
Test1.java:7: ')' expected
                ^
3 errors
Normal Termination
Output completed (0 sec consumed).

```

```
// program for test print and newline \n
```

```
import java.lang.*;
```

```
public class HelloTest
```

```
{
```

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

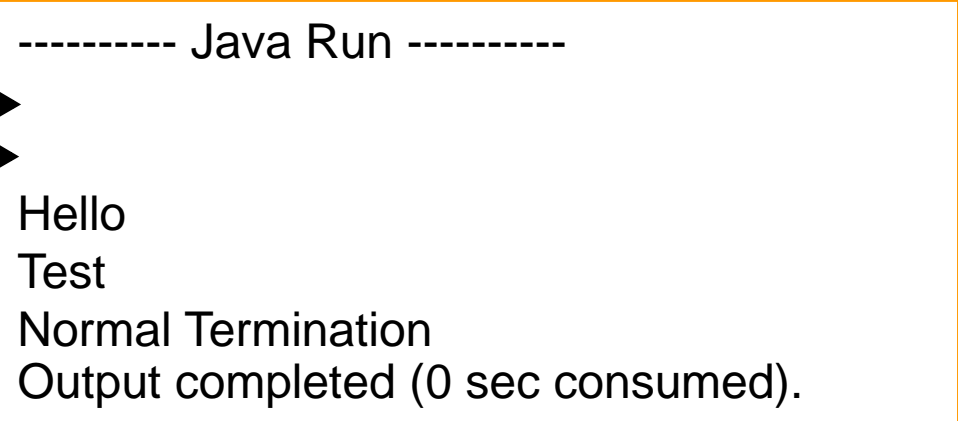
```
    {
```

```
        System.out.println();
```

```
        System.out.println("\nHello\nTest");
```

```
    }
```

```
}
```



```
----- Java Run -----  
→  
→  
Hello  
Test  
Normal Termination  
Output completed (0 sec consumed).
```

The terminal output shows the execution of the Java program. It starts with a separator line "----- Java Run -----". Two arrows point to the first two lines of output, "Hello" and "Test", which are printed on separate lines. The output then continues with "Normal Termination" and "Output completed (0 sec consumed)." on the following lines.

```
// program for test print and tab \t
```

```
public class HelloTest  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Yes\n\tI do");  
    }  
}
```

ผลลัพธ์ที่ได้จากโปรแกรม

```
----- Java Run -----  
Yes  
I do  
Normal Termination  
Output completed (0 sec consumed).
```

```
run:  
Yes  
I do  
BUILD SUCCESSFUL (total time: 0 seconds)
```

escape character ใช้สัญลักษณ์ \ ตามด้วยตัวอักษร ซึ่งมีความหมายในการใช้งานดังนี้

Escape code	ความหมาย
\n	Newline
\t	Tab
\b	backspace
\r	Carriage return
\f	Formfeed
\\	Backslash
\'	Single quote
\"	Double quote
\ddd	Octal decimal
\uxxxx	Hexa decimal

escape character

คือ การกำหนดค่าให้กับตัวแปรแบบ char โดยใช้รหัส Escape

ตัวอย่าง

```
public class Hello {  
    public static void main(String[] args)  
    {  
        System.out.println("Hello \"student\" \n"  
            + "Are \\you\\ ready");  
    }  
}
```

```
Hello "student"  
Are \\you\\ ready
```

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	.
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

วิธีหัดอ่านจากตาราง

ฐาน 8 oct \055 คือ -

ฐาน 8 oct \101 คือ A

ฐาน 16 hex \u002D คือ -

ฐาน 16 hex \U0041 คือ A

ส่วน 128-255 เป็น Extended ASCII Codes

```

public class EspSeq {
    public static void main(String[] args) {
        System.out.println('\055');           // - oct
        System.out.println('\u002d');         // - hex
        System.out.println("\101");           // oct is A
        System.out.println("\u0041");         // hex is A
        System.out.println("\137");           // _
        System.out.println("\u0020");         // space
        System.out.println("\057");
        System.out.println("\u002F");
        System.out.println("\u0048\u0065\u006c\u006c\u0020\u0041");
        System.out.println("\u0023\u0023\u0023\u0023\u0023\u0023\u0023\u0023");
    }
}

```

```

----- Java run -----
-
-
A
A
-
/
/
Hello
#####
Normal Termination
Output completed (0 sec consumed).

```



```
public class EspSeqFish
{
    public static void main(String[] args)
    {
        System.out.println("\u0020\u0020\u057\140\134\u057\n"+
                            "\074\060\n" +
                            "\u0020\u0020\134\140\u057\134\n");
    }
}
```

----- Java run -----

```
 /`V/
<0
 \`/\
```

Normal Termination
Output completed (0 sec consumed).