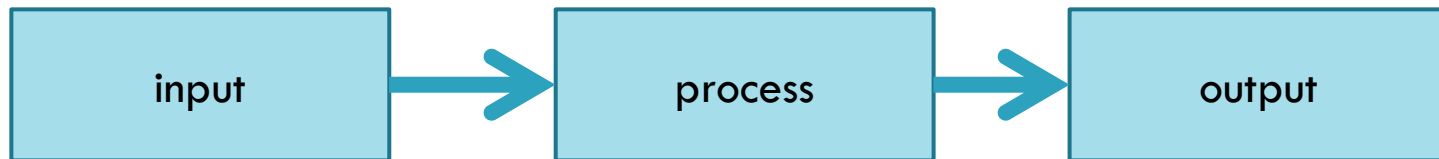


CHAPTER01

ความรู้เบื้องต้นเกี่ยวกับภาษาจาวา

ระบบคอมพิวเตอร์

- หมายถึงกระบวนการหรือขั้นตอนการทำงานของคอมพิวเตอร์
- ประกอบไปด้วยการรับข้อมูล (**input**) การประมวลผล (**process**) และการแสดงผล (**output**)



บล็อกไดอะแกรมระบบคอมพิวเตอร์

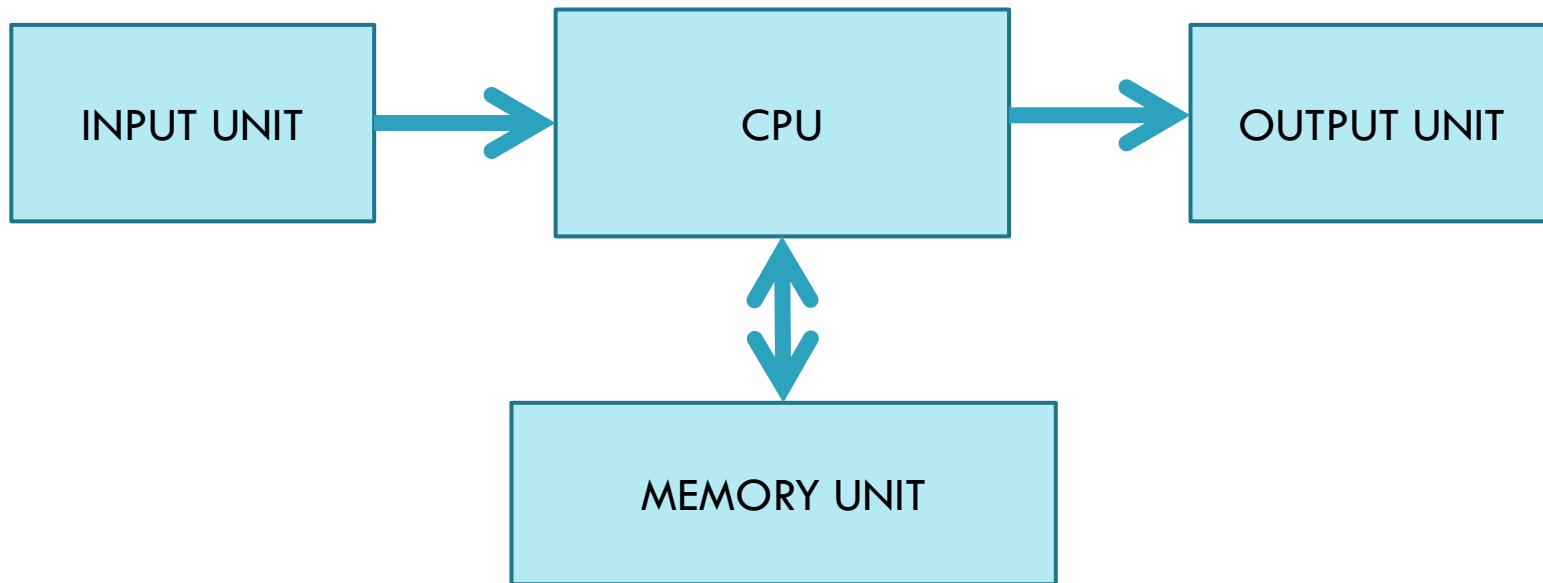
องค์ประกอบสำคัญของระบบคอมพิวเตอร์

- ระบบคอมพิวเตอร์มีองค์ประกอบที่สำคัญ 3 ส่วน คือ
- Hardware
- Software
- Peopleware

สถาปัตยกรรมคอมพิวเตอร์

- องค์ประกอบพื้นฐานของคอมพิวเตอร์ประกอบไปด้วย 4 ส่วนหลักที่สำคัญ คือ
 - 1. Input Unit
 - 2. Output Unit
 - 3. CPU
 - 4. Memory Unit

สถาปัตยกรรมคอมพิวเตอร์ (ต่อ)



องค์ประกอบของระบบคอมพิวเตอร์

INPUT UNIT

- เป็นหน่วยที่ทำหน้าที่รับข้อมูลจากภายนอกเข้าไปในคอมพิวเตอร์
- Keyboard
- Mouse
- Barcode Reader
- Light Pen เป็นต้น

CPU : Central Processing Unit

- ❑ คือหน่วยประมวลผลกลาง
- ❑ ภายใน CPU ประกอบไปด้วย
- ❑ CU : Control Unit ทำหน้าที่ควบคุมการทำงานของ CPU
- ❑ และ ALU : Arithmetic Logic Unit ทำหน้าที่ประมวลผลข้อมูลทางด้านคณิตศาสตร์ และตรรกะ
- ❑ นอกจากนี้ยังมี Register ซึ่งเป็นหน่วยความจำขนาดเล็กภายใน CPU

OUTPUT UNIT

- ❑ คือหน่วยที่ทำหน้าที่นำข้อมูลที่ถูกระประมวลผลแล้วไปแสดงผล อุปกรณ์ที่ใช้ในการแสดงผลได้แก่
- ❑ Monitor
- ❑ Speaker
- ❑ Printer
- ❑ Projector เป็นต้น

MEMORY UNIT

- แบ่งได้ 2 ประเภท คือ
- **Main Memory** : หน่วยความจำหลัก ได้แก่ **RAM , ROM** ทำหน้าที่เก็บข้อมูลที่ได้รับมาจาก **Input Unit** หรือเก็บโปรแกรมที่อ่านมาจาก **Secondary Memory** เพื่อส่งต่อไปยัง **CPU** ทำการประมวลผลต่อไป
- และ **Secondary Memory** : หน่วยความจำสำรอง ได้แก่ ฮาร์ดดิสก์ ,แผ่น **CD**, แผ่น **DVD** , **Flash Drive** เป็นต้น

ภาษาคอมพิวเตอร์

- หมายถึง ภาษาที่ใช้สำหรับเขียนโปรแกรม หรือชุดคำสั่งที่ใช้ควบคุมการทำงานของคอมพิวเตอร์ ให้ปฏิบัติตามความต้องการของผู้ใช้งาน
- ภาษาคอมพิวเตอร์แบ่งออกได้เป็น 2 ระดับ คือ
- 1. ภาษาระดับต่ำ (Low Level Language)
- 2. ภาษาระดับสูง (High Level Language)

Low Level Language

- เป็นภาษาคอมพิวเตอร์ที่ใช้กันในยุคแรกๆ
- มีความยุ่งยากในการเขียนมาก แบ่งได้เป็น 2 ประเภท คือ
- 1. Machine Language (ภาษาเครื่อง)
- 2. Assembly Language (ภาษาแอสเซมบลี)

Machine Language

- เป็นภาษาหรือคำสั่งที่ใช้ในการสั่งงาน หรือติดต่อกับเครื่องคอมพิวเตอร์โดยตรง
- ภาษาเครื่องประกอบไปด้วยรหัสเลขฐานสองซึ่งเทียบได้กับลักษณะของสัญญาณทางไฟฟ้า เข้ากับหลักการทำงานของเครื่องคอมพิวเตอร์พอดีทำให้คอมพิวเตอร์ทำงานตามคำสั่งได้ทันที

Assembly Language

- ❑ ภาษาแอสเซมบลี จัดเป็นภาษาสัญลักษณ์ (Symbolic Language)
- ❑ เป็นภาษาที่พัฒนาต่อจากภาษาเครื่อง
- ❑ ใช้สัญลักษณ์ข้อความแทนกลุ่มของเลขฐานสอง
- ❑ ทำให้การเขียนโปรแกรมสะดวกมากขึ้นกว่าภาษาเครื่อง
- ❑ แต่ก็ยังคงต้องจำความหมายของสัญลักษณ์ที่ใช้แทนภาษาเครื่อง

High Level Language

- เป็นภาษาคอมพิวเตอร์ที่ได้รับการพัฒนาให้สามารถใช้งานได้ง่ายและสะดวกมากขึ้น
- การเขียนภาษาไม่ขึ้นกับฮาร์ดแวร์หรือลักษณะการทำงานของเครื่องคอมพิวเตอร์
- โปรแกรมเมอร์ไม่จำเป็นต้องเข้าใจระบบการทำงานของเครื่องคอมพิวเตอร์มากนัก เพียงแต่เข้าใจกฎเกณฑ์ในการเขียนแต่ละภาษาให้ดีตัวภาษาคลายกับภาษาอังกฤษ

High Level Language (Continue)

- ภาษาระดับสูงเครื่องคอมพิวเตอร์จะยังไม่เข้าใจ จึงต้องมีการแปลให้เป็นภาษาเครื่องเสียก่อน
- โปรแกรมที่ใช้แปลภาษาระดับสูง แบ่งออกได้เป็น 2 ประเภท คือ
- 1. interpreter
- 2. compiler

Interpreter

- ❑ การทำงานของอินเตอร์พรีเตอร์จะแปลภาษาทีละคำสั่ง แปลเสร็จก็จะปฏิบัติตามคำสั่งที่แปลโดยทันที หลังจากนั้นจึงไปแปลคำสั่งถัดไป
- ❑ ถ้าหากในขณะที่แปลคำสั่งเกิดข้อผิดพลาดก็จะหยุดให้โปรแกรมเมอร์แก้ไขข้อผิดพลาดคำสั่งนั้นก่อน
- ❑ ภาษาที่ใช้อินเตอร์พรีเตอร์ ได้แก่ **BASICA , GWBASIC** เป็นต้น

Compiler

- ❑ คอมไพเลอร์จะทำการแปลทั้งโปรแกรม แล้วเก็บโปรแกรมที่แปลได้ในรูปแบบของออปเจ็คโปรแกรม (**object program**)
- ❑ ถ้าโปรแกรมที่แปลได้ไม่มีข้อผิดพลาด ก็จะปฏิบัติตามคำสั่งนั้นๆ ทันที
- ❑ ถ้าโปรแกรมที่แปลมีข้อผิดพลาด คอมไพเลอร์ก็จะบอกข้อผิดพลาดทั้งหมดให้ทราบ
- ❑ ภาษาที่ใช้คอมไพเลอร์ ได้แก่ **C, Pascal, FORTRAN** เป็นต้น

วิวัฒนาการภาษาจาวา



- ปี คศ. 1991 ชันไมโครซิสเต็ม (Sun Microsystems) ได้ทำการวิจัยเพื่อพัฒนาซอฟต์แวร์ที่ใช้ควบคุมอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก ซึ่งได้ให้กำเนิดภาษาโอ๊ค (Oak)
- ปี คศ. 1993 ภาษาโอ๊คได้ถูกปรับปรุงใหม่เพื่อใช้ในการสร้างเว็บแอปพลิเคชัน (Web Application) พร้อมกับสร้างเว็บเบราว์เซอร์ (Web Browser) ที่รองรับ ชื่อว่าเว็บรันเนอร์ (Web Runner)

วิวัฒนาการภาษาจาวา (ต่อ)

- ❑ คศ. 1995 บริษัทซันไมโครซิสเต็มได้เปิดตัวภาษาจาวา (Java) โดยเปลี่ยนชื่อมาจากภาษาโอ๊ค (เนื่องจากไปซ้ำกับภาษาที่มีอยู่แล้ว) พร้อมกับเว็บเบราว์เซอร์ ที่รองรับภาษาจาวา ชื่อว่า **HotJava (Web Runner เดิม)**
- ❑ บริษัทซันไมโครซิสเต็มได้มีการแจกจ่าย **Java Development Kit (JDK)** ซึ่งเป็นชุดพัฒนาโปรแกรมภาษาจาวาในอินเทอร์เน็ต



ข้อดีของภาษาจาวา

- ❑ ทำงานบนเว็บเบราว์เซอร์ได้
- ❑ มีความปลอดภัยสูง
- ❑ สนับสนุนการทำงานหลายระดับ
- ❑ สามารถทำงานบนเครื่องคอมพิวเตอร์ต่างระบบได้
- ❑ เป็นภาษาเชิงวัตถุ
- ❑ มีความเรียบง่าย

ข้อดีของภาษาจาวา (ต่อ)

- มีระบบจัดการคืนพื้นที่ในหน่วยความจำอัตโนมัติ (automatic garbage collection)
- มีคลาสจำนวนมากมาให้ใช้งาน
- ฟรี

จาวาแพลตฟอร์ม

- ❑ โปรแกรมที่เขียนด้วยภาษาจาวาสามารถนำไปทำงานบนระบบปฏิบัติการต่างๆ กัน เช่น **Windows** หรือ **Linux** หรือ **MacOS**
- ❑ สามารถทำงานบนคอมพิวเตอร์ที่ใช้ซีพียูคนละแบบ เช่น ซีพียูของ **intel** หรือซีพียูของ **IBM**
- ❑ **Write Once Run Anywhere** นั่นเอง
- ❑ นั่นเพราะว่าจาวามีระบบที่เรียกว่า “จาวาแพลตฟอร์ม”

จาวาแพลตฟอร์ม (ต่อ)

- ❑ เราเรียกระบบที่โปรแกรมอาศัยทำงานว่า “แพลตฟอร์ม (platform)”
- ❑ แพลตฟอร์มตีความหมายรวมไปถึงฮาร์ดแวร์และซอฟต์แวร์ด้วย หรือ
- ❑ แพลตฟอร์มอาจจะหมายถึงเฉพาะส่วนที่เป็นซอฟต์แวร์เพียงอย่างเดียว เช่น ระบบปฏิบัติการ และชุดของโปรแกรมย่อยที่ใช้ในการพัฒนาซอฟต์แวร์หรือไลบรารี (library)
- ❑ จาวาแพลตฟอร์มนั้นเป็นซอฟต์แวร์

จาวาแพลตฟอร์ม (ต่อ)

- ❑ จาวาแพลตฟอร์มประกอบไปด้วย ตัวแปลภาษา โปรแกรมเครื่องจักรเสมือน (JVM) และคลาสที่เป็นแกนหลักของภาษาจาวา
- ❑ จาวามี 3 แพลตฟอร์ม คือ
- ❑ 1. Java 2 Platform, Standard Edition (J2SE) ใช้สำหรับเขียนโปรแกรมทั่วไปหรือที่เรียกว่า “จาวาแอปพลิเคชัน (java application)” และโปรแกรมที่ทำงานบนเว็บเบราว์เซอร์ หรือที่เรียกว่า “แอปเพล็ต (java applet)”

จาวาแพลตฟอร์ม (ต่อ)

- ❑ **2. Java 2 Platform, Enterprise Edition (J2EE)** ใช้สำหรับเขียนโปรแกรมแบบมัลติเทียร์ (Multitiered) พัฒนามาจาก J2SE และเพิ่มคุณสมบัติสำหรับการพัฒนาโปรแกรมในระดับองค์กร
- ❑ **3. Java 2 Platform, Micro Edition (J2ME)** เป็นแพลตฟอร์มที่ทำงานบนสินค้าอิเล็กทรอนิกส์ เช่น โทรศัพท์มือถือรุ่นเก่า พีดีเอ และกล่องเคเบิลทีวี (TV set-top box)

เครื่องมือสำหรับเขียนโปรแกรมภาษาจาวา

- 1. ชุดพัฒนาภาษาจาวา (Java Development Kit : JDK)
 - ▣ โปรแกรมแปลภาษาจาวา (Java Compiler)
 - ▣ โปรแกรมรันภาษาจาวา (Java Runtime Environment : JRE)
- 2. โปรแกรมที่ใช้เขียนต้นฉบับโปรแกรม (Text Editor)

ชุดพัฒนาภาษาจาวา

- ❑ ชุดพัฒนาโปรแกรมภาษาจาวา (Java Development Kit : JDK) มีส่วนประกอบที่สำคัญคือ
- ❑ คอมไพเลอร์ และสภาพแวดล้อมสำหรับรันโปรแกรมจาวา
- ❑ ซึ่ง **jdk** สามารถดาวน์โหลดได้จากเว็บไซต์
<http://www.oracle.com>

Java compiler

- ทำหน้าที่แปลภาษาจาวาให้เป็นไบต์โค้ด (byte code) ซึ่งเป็นคำสั่งที่ไม่อิงกับซีพียูของเครื่องคอมพิวเตอร์ยี่ห้อใดยี่ห้อหนึ่ง
- เครื่องคอมพิวเตอร์ที่ประมวลผลไบต์โค้ดเป็น “เครื่องคอมพิวเตอร์เสมือน” หรือที่เราเรียกว่า เครื่องจักรเสมือนจาวา Java Virtual Machine



Java Runtime Environment : JRE

- เป็นโปรแกรมที่ใช้รับไบต์โค้ด สภาพแวดล้อมนี้ประกอบไปด้วยเครื่องจักรเสมือนจาวา (**Java Virtual Machine – JVM**) และคลาสที่เป็นแกนหลักของภาษาจาวา คอมพิวเตอร์ใดๆ ที่มี **JRE** ติดตั้งอยู่จะสามารถรันโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาได้



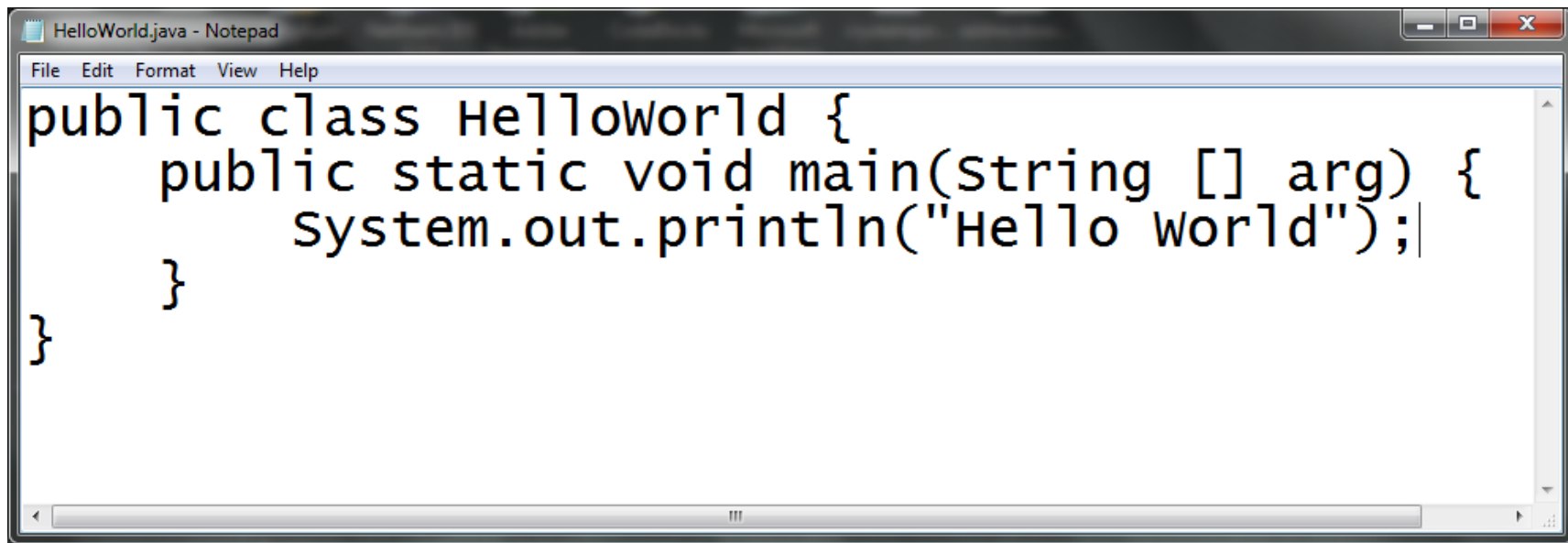
โปรแกรมที่ใช้เขียนต้นฉบับโปรแกรม (Text Editor)

- ❑ ชุดพัฒนาโปรแกรมภาษาจาวามีโปรแกรมสำหรับคอมไพล์และรันไบต์โค้ด แต่ไม่มี **Text Editor** ที่ใช้สำหรับเขียนไฟล์นามสกุล **java**
- ❑ แต่เราสามารถใช่ **Text Editor** ตัวใดก็ได้เพื่อใช้เขียนโปรแกรม
- ❑ **Text Editor** ได้แก่ Notepad, J-Lab, Netbeans, Eclipse, Visual Studio Code, IntelliJ

โปรแกรม HelloWorld

- โปรแกรมที่ผู้เขียนส่วนใหญ่เขียนเป็นโปรแกรมแรก ส่วนใหญ่ต้องเขียนโปรแกรม **HelloWorld**
- เริ่มต้นทดลองเขียนโดยเปิด **Notepad** ขึ้นมา
- สร้างไฟล์ชื่อว่า **HelloWorld.java**
- บันทึกไฟล์ไว้ที่ **D:\javaCode** (บันทึกในไดร์ฟอะไรก็ได้)

โปรแกรม HelloWorld (ต่อ)



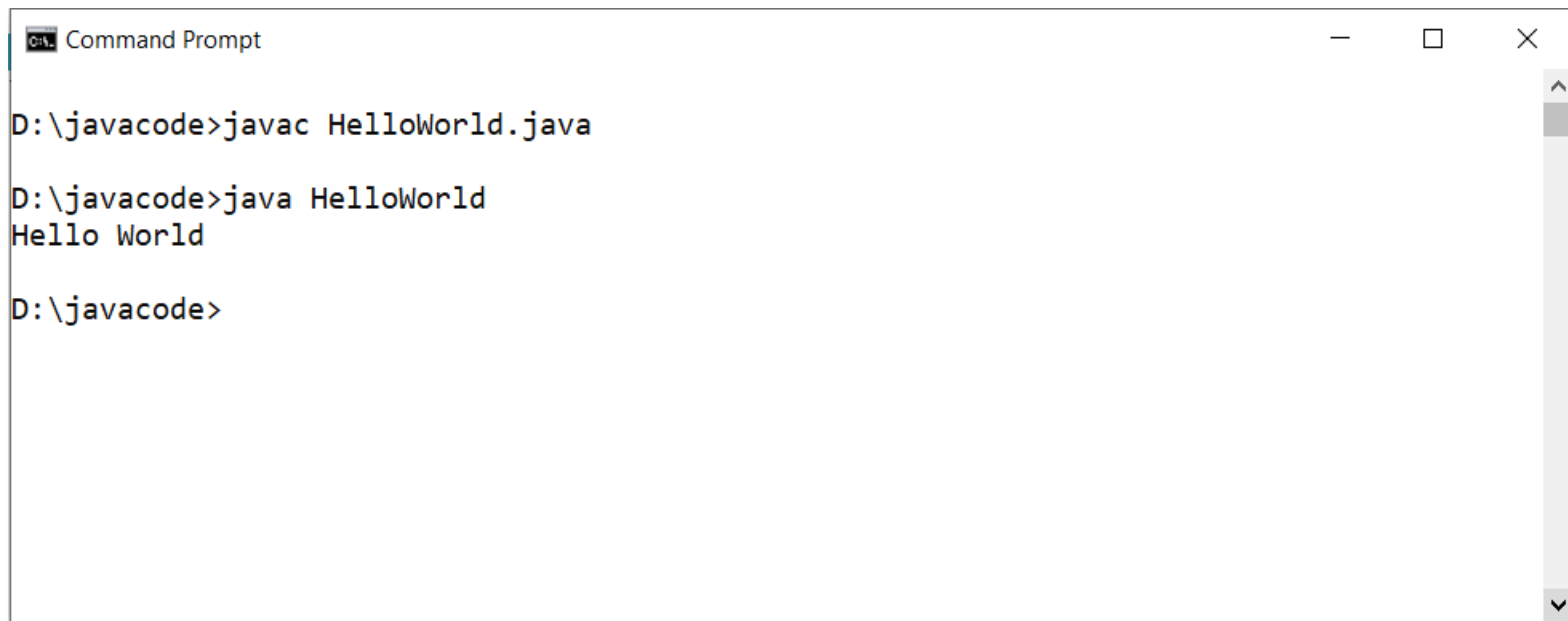
A screenshot of a Notepad window titled "HelloWorld.java - Notepad". The window contains the following Java code:

```
public class HelloWorld {  
    public static void main(String [] arg) {  
        System.out.println("Hello world");  
    }  
}
```


โปรแกรม HelloWorld (ต่อ)

- ❑ เมื่อเขียนโปรแกรมเสร็จแล้ว เราต้องคอมไพล์โปรแกรมก่อน จึงจะสามารถรันโปรแกรมได้
- ❑ การคอมไพล์และรันโปรแกรม **HelloWorld** ทำได้โดยการเปิดโปรแกรม **command prompt** จากนั้นก็พิมพ์คำสั่ง
- ❑ **javac HelloWorld.java** (คอมไพล์โปรแกรม จะต้องไปยังโฟลเดอร์ที่มีไฟล์นี้อยู่)
- ❑ **java HelloWorld** (รันโปรแกรม จะต้องไปยังโฟลเดอร์ที่มีไฟล์นี้อยู่)

โปรแกรม HelloWorld (ต่อ)



```
Command Prompt

D:\javacode>javac HelloWorld.java

D:\javacode>java HelloWorld
Hello World

D:\javacode>
```

The image shows a Windows Command Prompt window titled "Command Prompt". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command history shows the user navigating to the directory "D:\javacode" and running the command "javac HelloWorld.java" to compile the program. Then, they run "java HelloWorld" to execute it, which outputs "Hello World". The prompt returns to "D:\javacode>" after the execution.

การตั้งค่า PATH

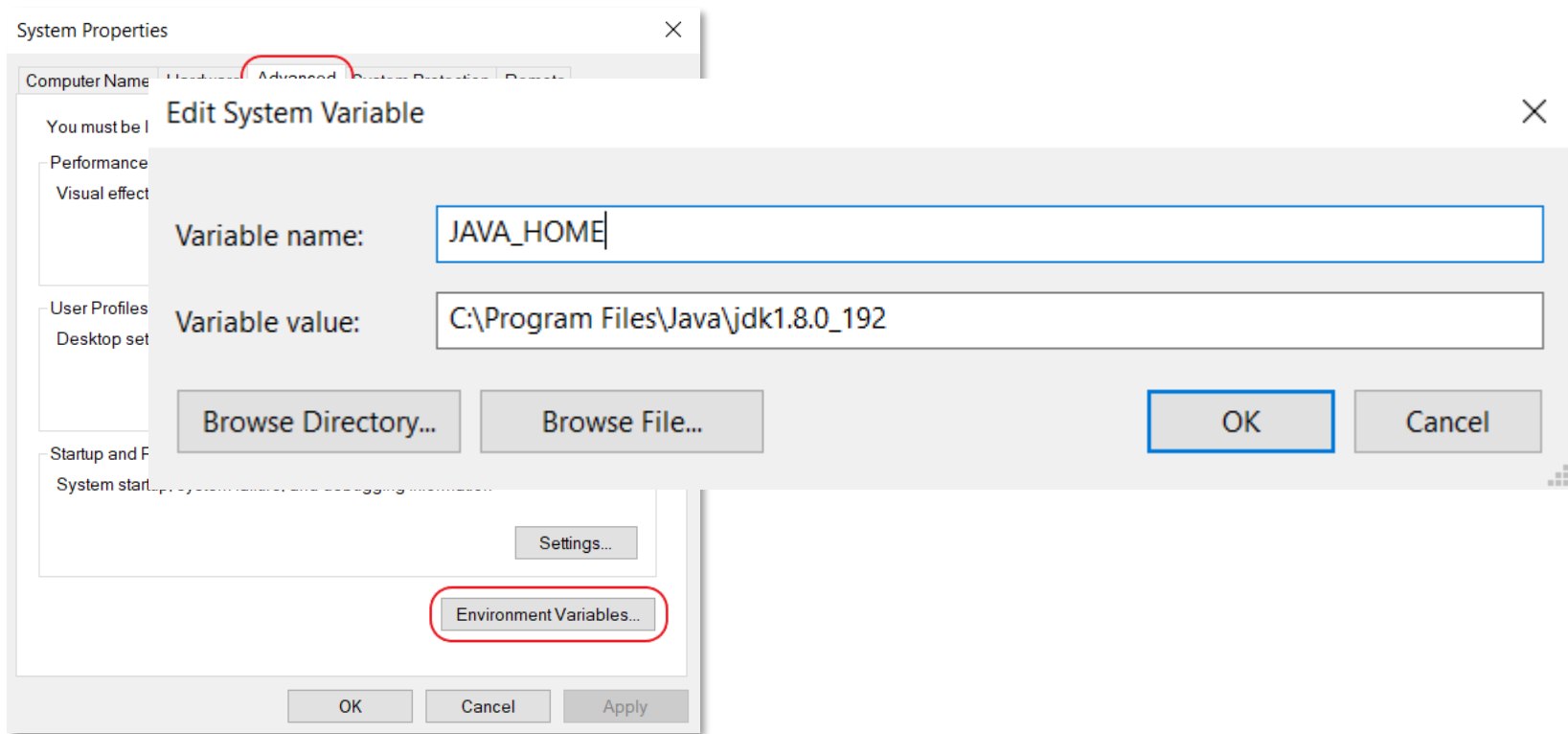
- โดยปกติแล้วโปรแกรม **command prompt** จะไม่รู้จักคำสั่ง **javac**
- วิธีการจะทำให้โปรแกรม **command prompt** รู้จักคำสั่งที่ใช้สำหรับพัฒนาโปรแกรมภาษาจาวาทำได้โดยการตั้งค่า **PATH** ให้แก่เครื่องคอมพิวเตอร์ มีขั้นตอนดังนี้
- 1. **copy path** ของ **jdk** ที่ติดตั้งในเครื่องคอมพิวเตอร์ เช่น

 > This PC > Windows 10 (C:) > Program Files > Java > jdk1.8.0_192

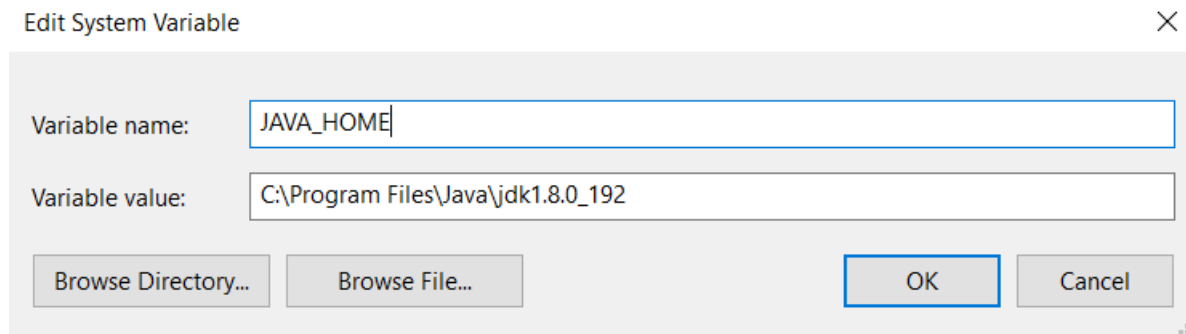
การตั้งค่า PATH (ต่อ)

- 2. กำหนดค่า **path** ที่ได้จากข้อ 1 ให้แก่ตัวแปร **path** ของระบบ
 - ▣ คลิกขวาที่ **My Computer > properties > Advanced system settings** เพื่อเปิด **system properties**

การตั้งค่า PATH (ต่อ)

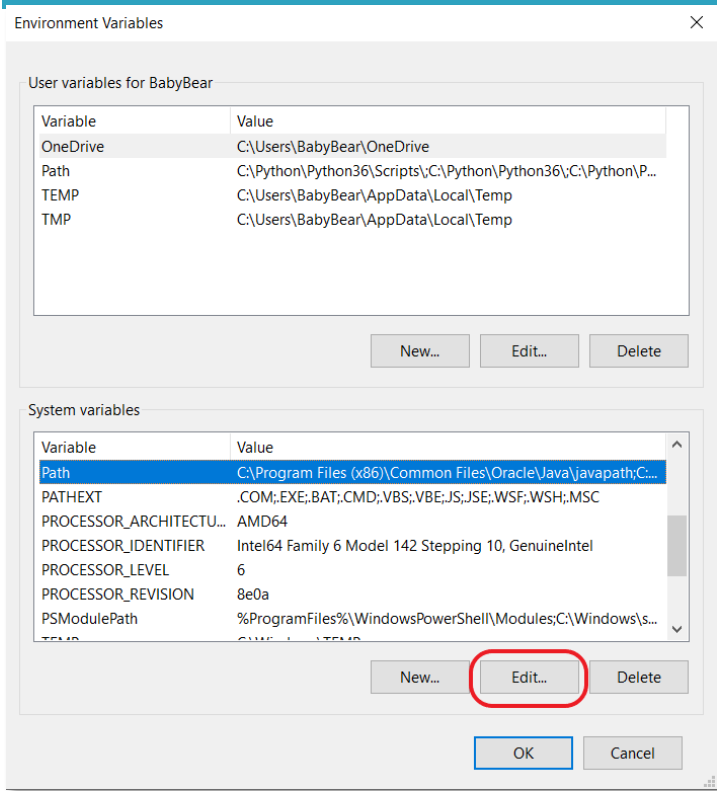


การตั้งค่า PATH (ต่อ)



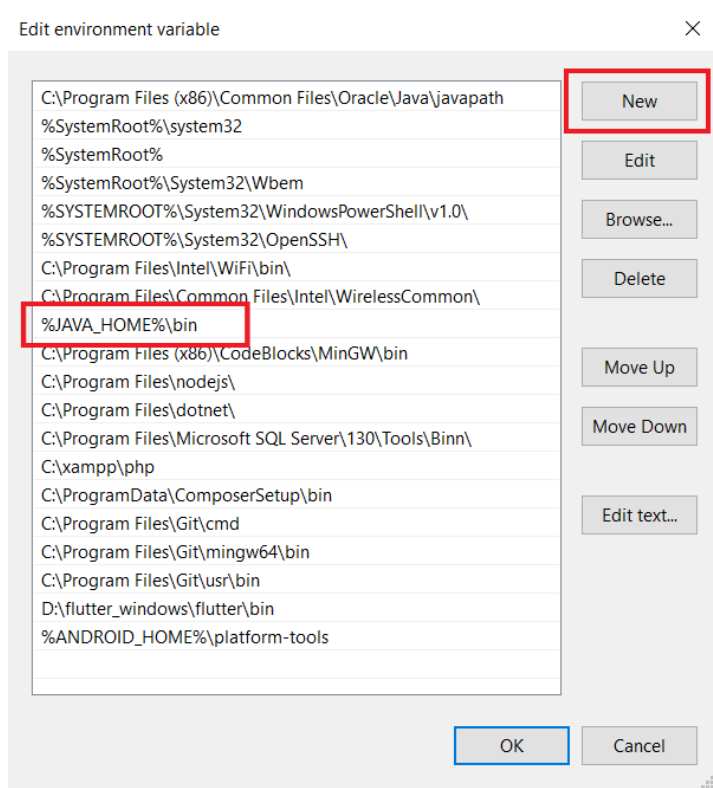
New Variable name

การตั้งค่า PATH (ต่อ)



เลือกตัวแปร **Path** แล้วคลิกที่ปุ่ม **Edit...**

การตั้งค่า PATH (ต่อ)



Paste พาทที่ก๊อปปี้ไว้ในข้อที่ 1 วางในตำแหน่ง **Variable value** ก่อน paste ให้ใส่เครื่องหมายเซมิโคลอน (;) คั่นระหว่างค่าของตัวแปรตัวอื่นก่อน หลังจากนั้นทำการคลิกที่ปุ่ม **OK**

เอกสารอ้างอิง

- วรเศรษฐ สุวรรณิก. เขียนโปรแกรม Java เบื้องต้น 2nd edition.
กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2556.

End.

