



Software Park Thailand  
</Code Camp>

SOFTWARE PARK  
THAILAND



# Basic Java Programming



Software Park Thailand  
</Code Camp>

# Agenda

1.วิธีการติดตั้งโปรแกรม **Eclipse**

2.Basic Syntax

3.Operator

4.Control Flow

5.Method

6.Strings

7.Arrays

8.Maven

9.Homework



Software Park Thailand  
</Code Camp>

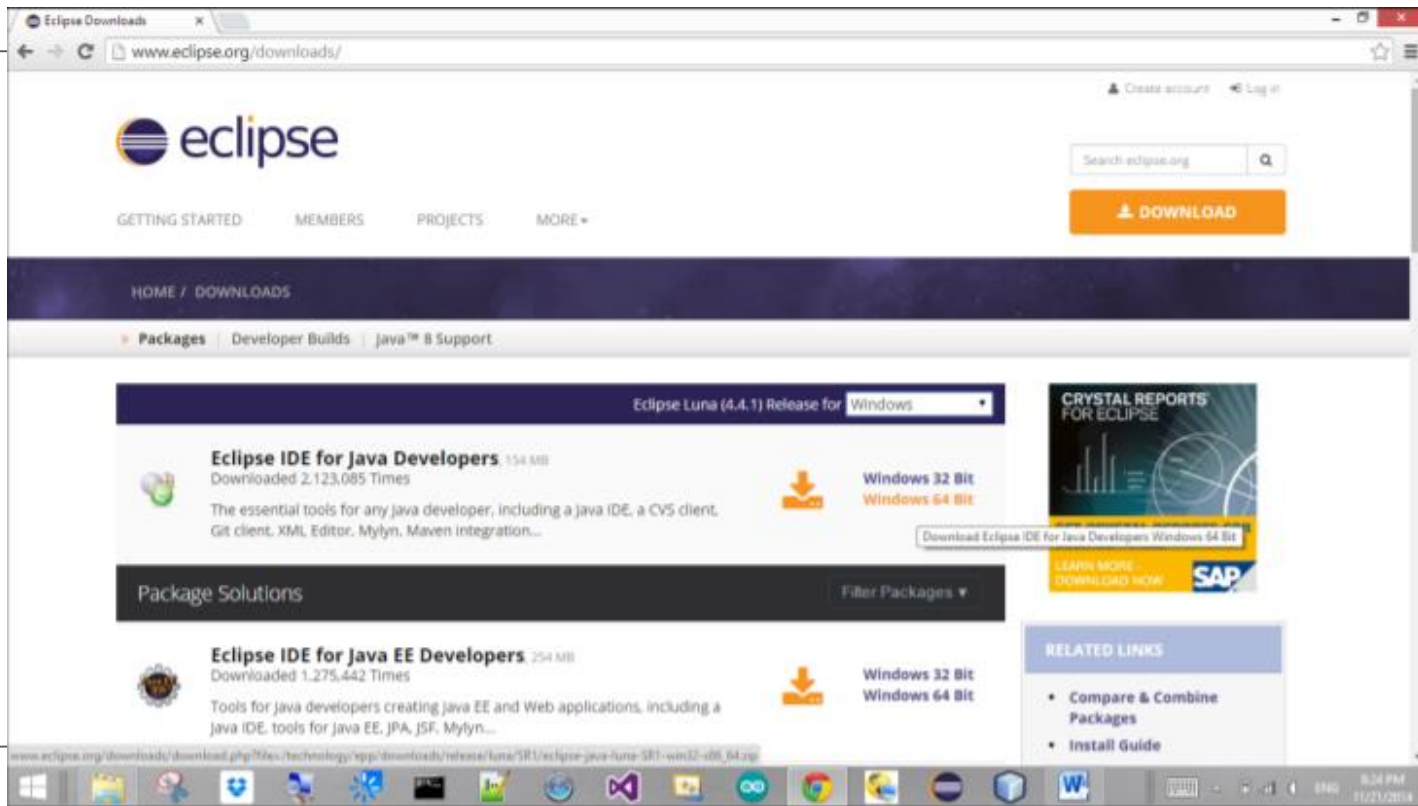
# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

1. ดาวน์โหลดโปรแกรม Eclipse IDE for Java Developers จากเว็บไซต์ [www.eclipse.org/downloads/](http://www.eclipse.org/downloads/) โดยเลือกดาวน์โหลดได้ทั้ง รุ่นสำหรับ Windows 32 Bit และ Windows 64 Bit รุ่นสำหรับ Linux และรุ่นสำหรับ Mac OS



Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse





Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

2. ทำการแตกไฟล์โดยการคลิกขวาที่ชื่อไฟล์(eclipse-java-luna-SR1-win32-x86\_64.zip) แล้วเลือกตำแหน่งที่จะเก็บไฟล์ ตัวอย่างเช่น D:\Java\eclipse หรือ D:\57019999\eclipse

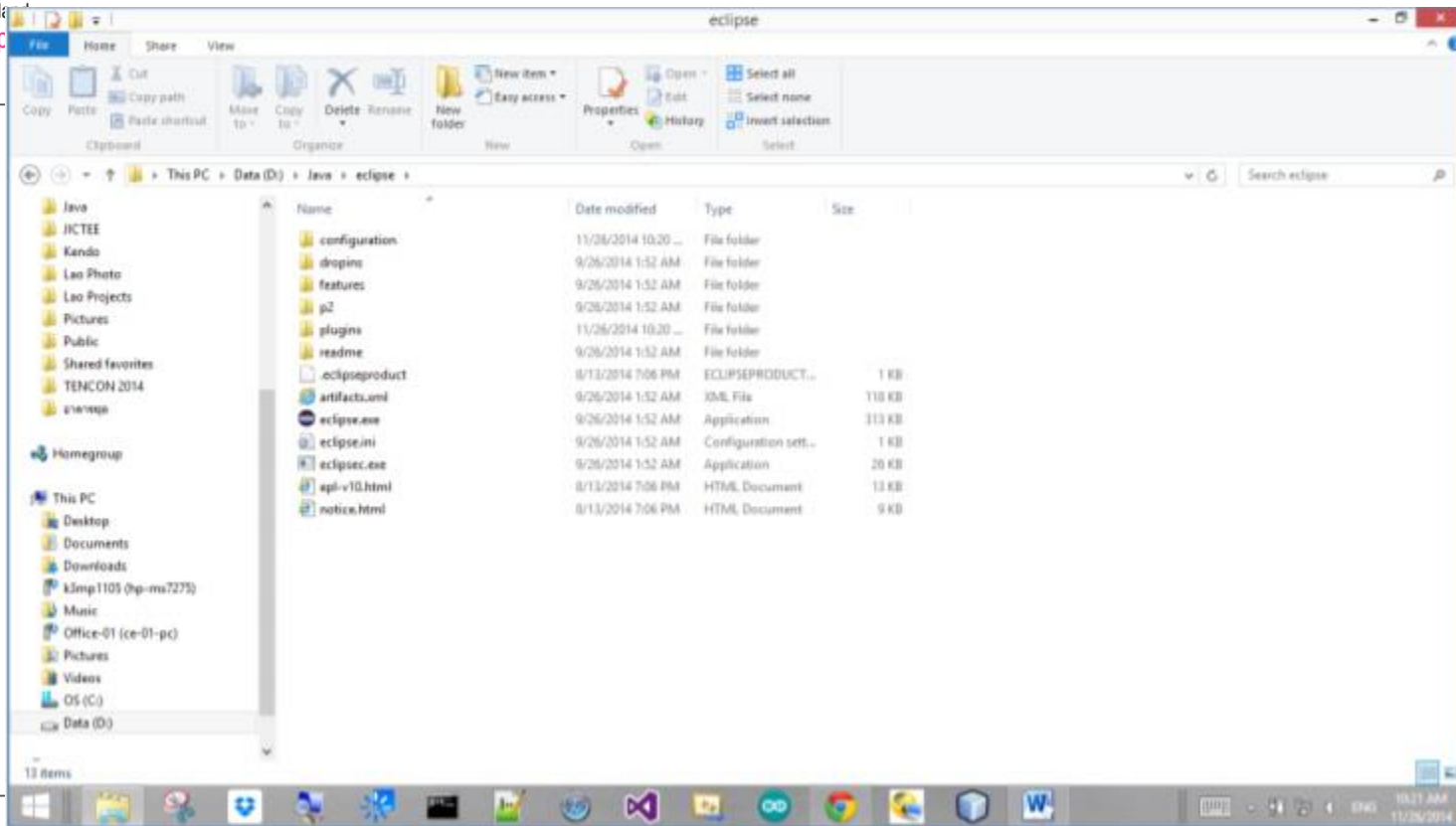
3. ทำการตรวจสอบไฟล์ที่แตกออกมาโดยคลิกเข้าไปดูในแฟ้มที่เก็บไฟล์ที่แตกออกมา



Software Park Thailand

</Code Camp

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse





Software Park Thailand  
</Code Camp>

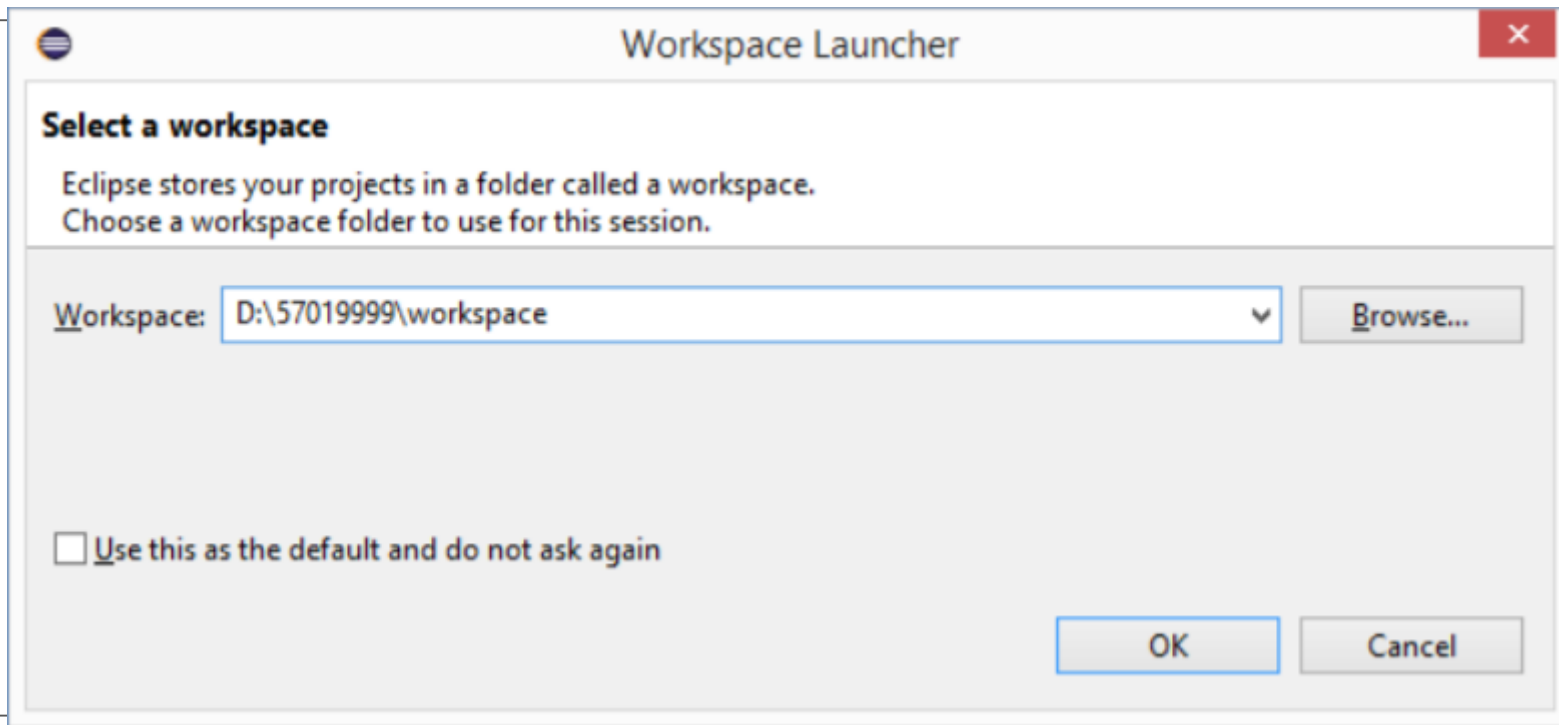
# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

4. สร้าง short cut แล้วนำไปเก็บไว้ที่ Desktop หรือนำไปแปะไว้ที่ Taskbar
5. ทำการสร้างแฟ้มข้อมูลเปล่า (กำหนดให้ใช้ชื่อ workspace) เพื่อใช้เป็นพื้นที่สำหรับทำงานร่วมกับโปรแกรม Eclipse ไว้ที่ Drive D: ตัวอย่างเช่น D:\Java\workspace หรือ D:\57019999\workspace
6. สั่งรันโปรแกรม Eclipse แล้วเลือกพื้นที่ทำงานตามแฟ้มที่ได้สร้างไว้ ตัวอย่างเช่น D:\57019999\workspace



Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

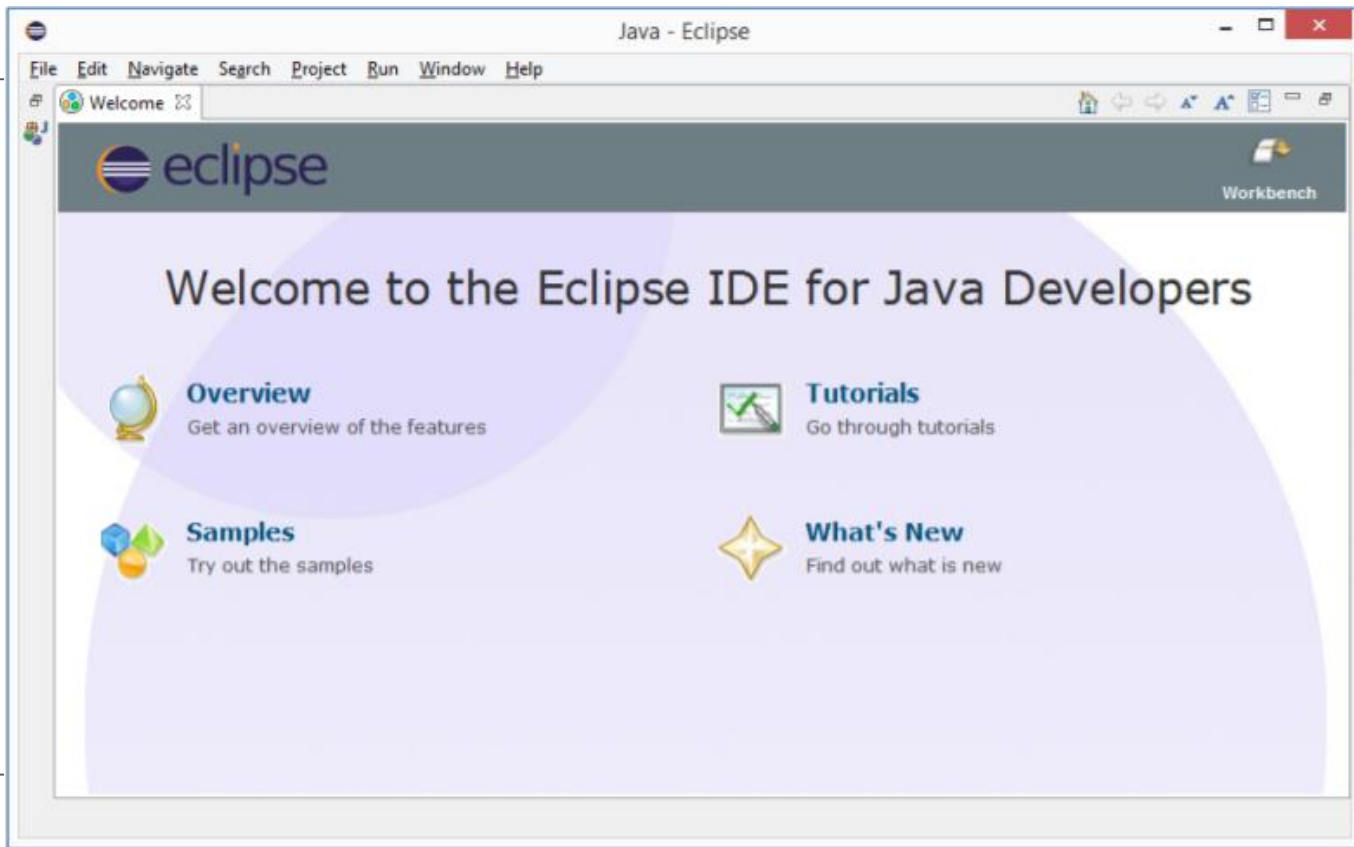






Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

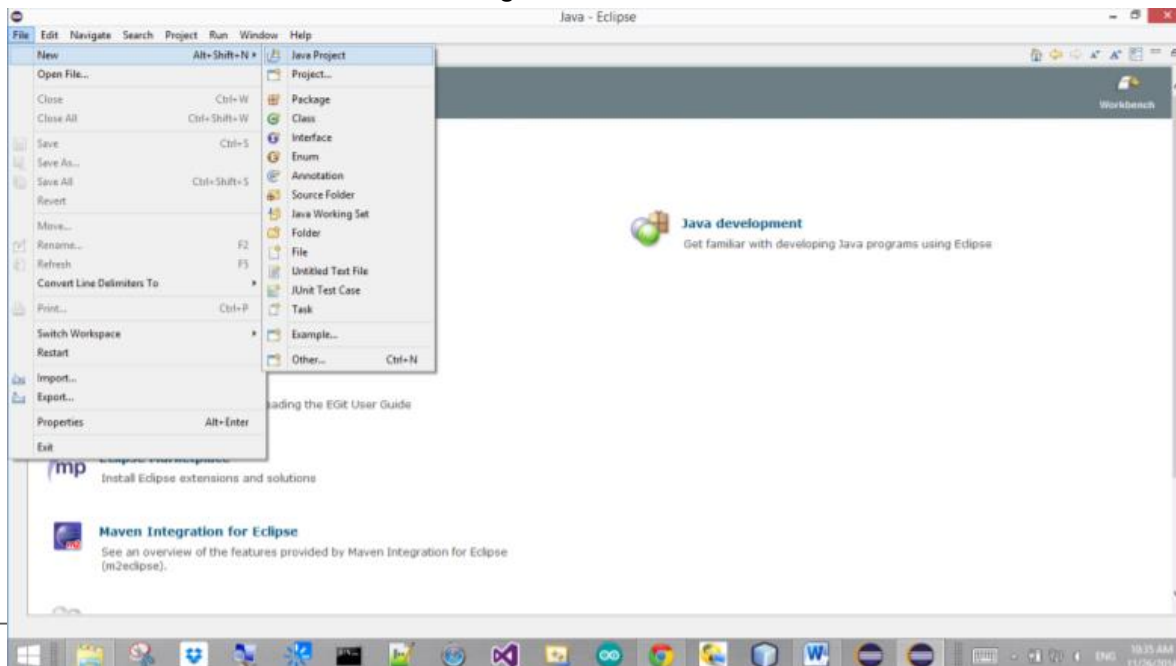




Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

## 7. สร้างโปรเจกเพื่อทำงาน โดยการเลือกเมนู File -> New -> Java Project

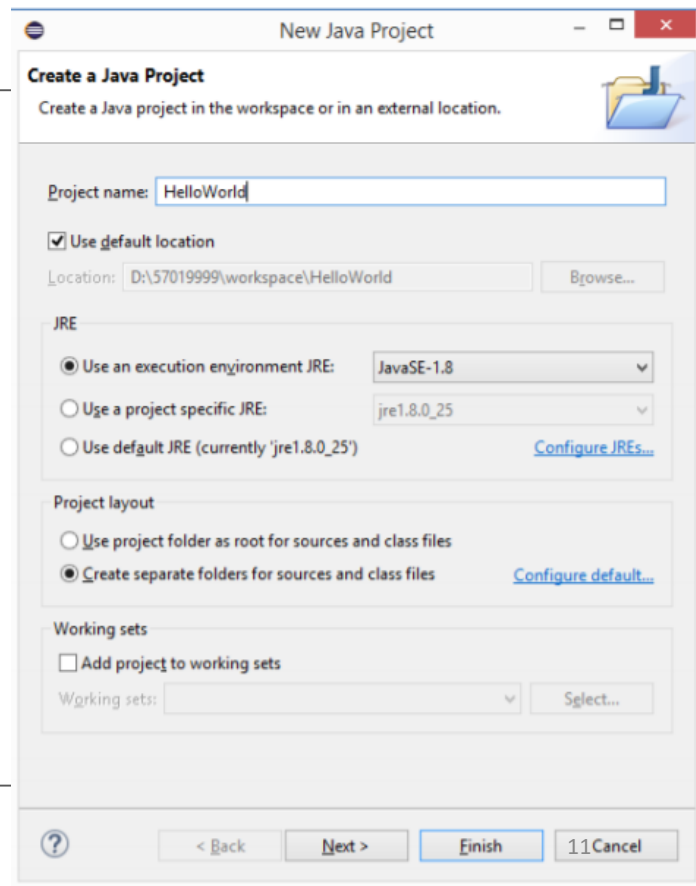




Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

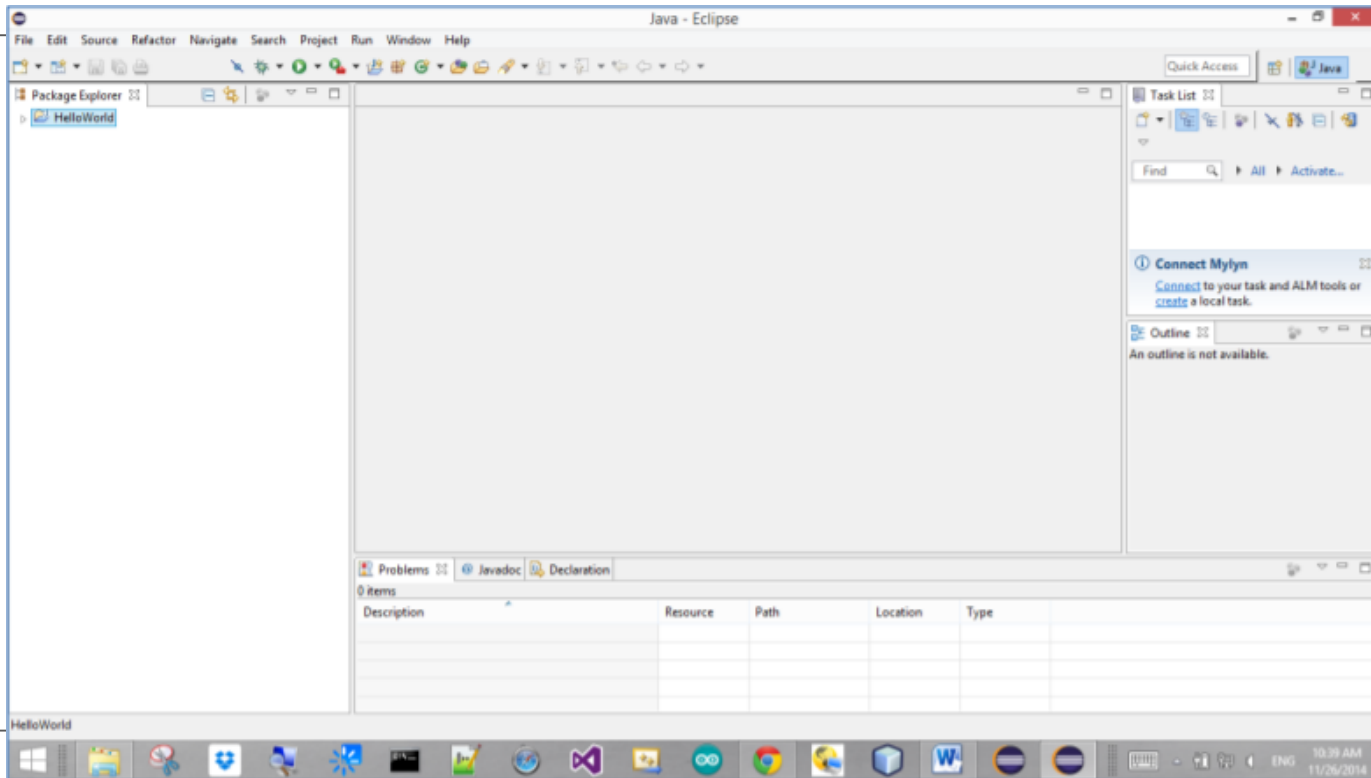
8. ป้อนชื่อโปรเจกต์ ตัวอย่างเช่น HelloWorld แล้วกด Finish





Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

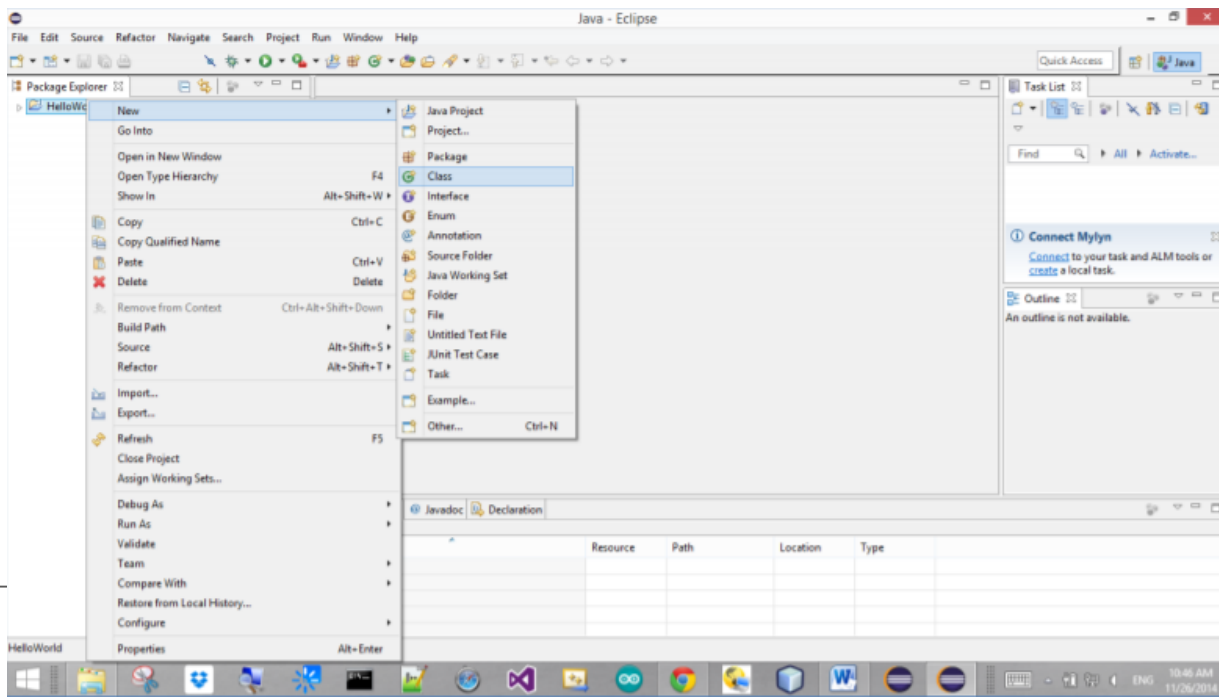




Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

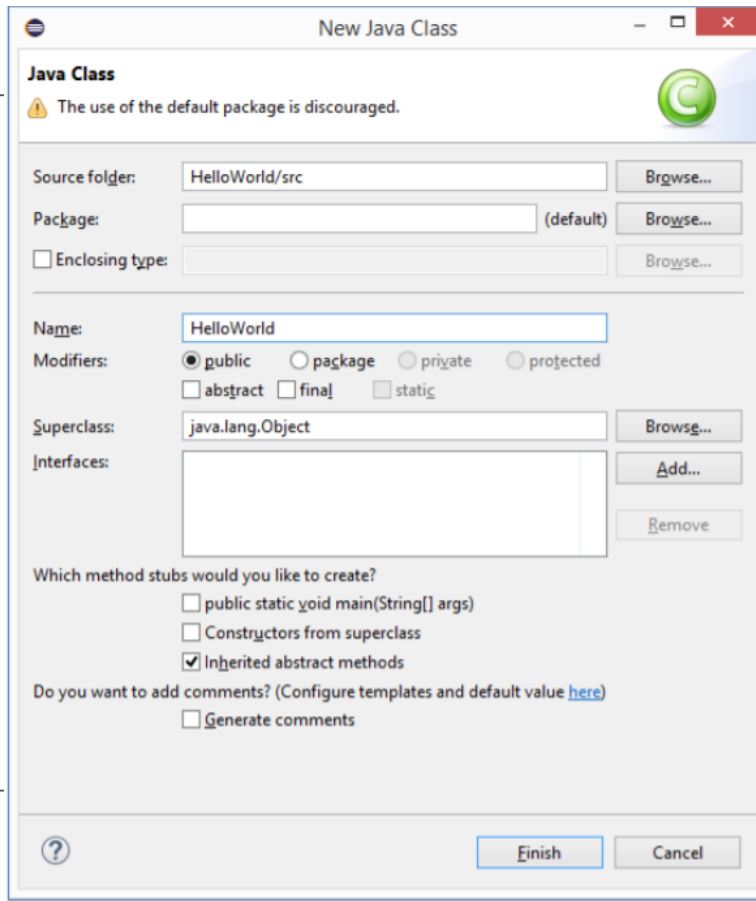
9. ที่หน้าจอ Package Explorer คลิกที่ชื่อแฟ้มโปรเจค HelloWorld แล้วคลิกขวา เลือก New -> Class ตั้งชื่อคลาสโดยให้ขึ้นต้นด้วยตัวใหญ่ HelloWorld แล้วกด Finish





Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse





# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

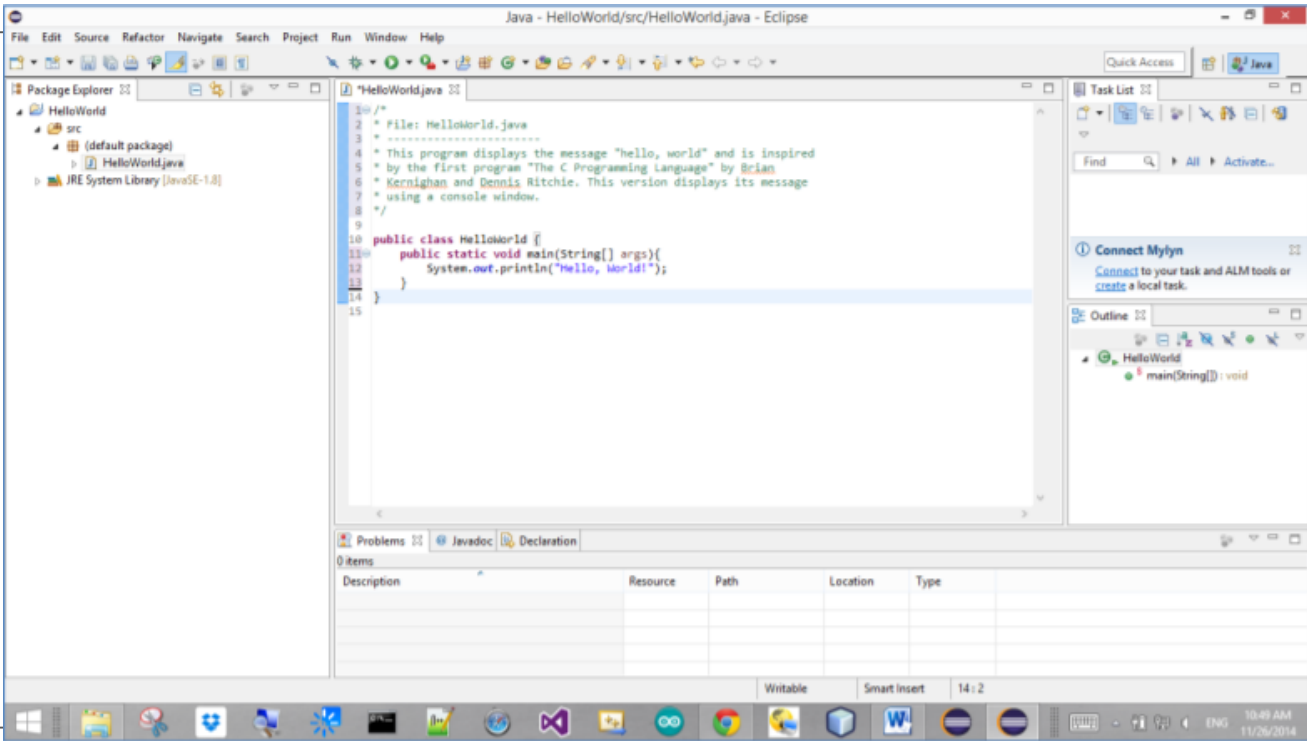
10. พิมพ์โปรแกรมตามตัวอย่างดังต่อไปนี้ (ดัดแปลงมาจากคู่มือ ACM Java Task Force Tutorial, Chapter 1, Page 2)

```
/*
 * File: HelloWorld.java
 * -----
 * This program displays the message "hello, world" and is inspired
 * by the first program "The C Programming Language" by Brian
 * Kernighan and Dennis Ritchie. This version displays its message
 * using a console window.
 */

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



# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse



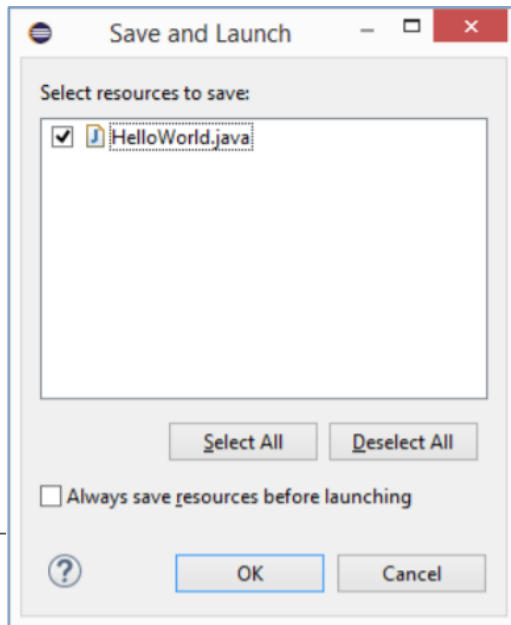




Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

11. เลือกเมนู Run แล้วเลือกคำสั่ง Run (หรือกด Ctrl+F11) จะปรากฏหน้าจอถามว่าต้องการบันทึกโปรแกรมก่อนรันหรือไม่ ให้เลือก OK ในกรณีที่ต้องการให้บันทึกโปรแกรมก่อนรันทุกครั้ง ให้คลิกที่ Always save resources before launching

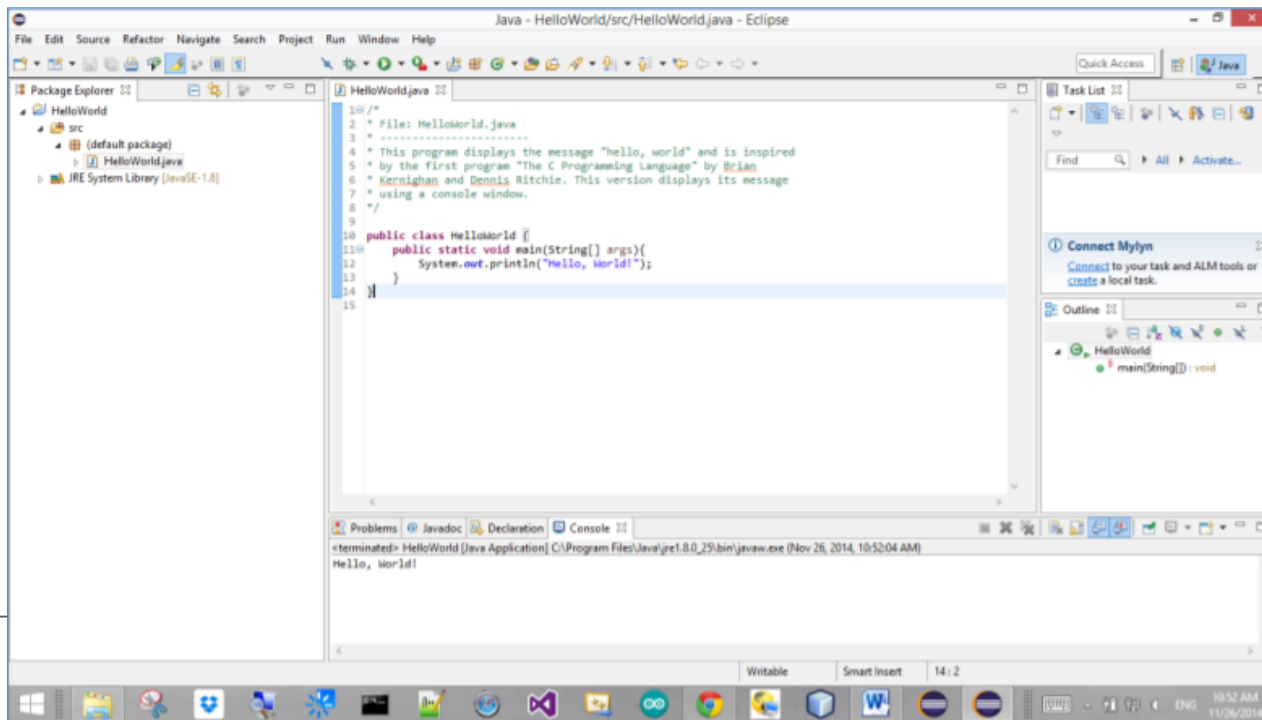




Software Park Thailand  
</Code Camp>

# วิธีการติดตั้งและใช้งานโปรแกรม Eclipse

## 12. จะปรากฏหน้าจอ Console ที่ด้านล่างแสดงผลการทำงานของโปรแกรม





Software Park Thailand  
</Code Camp>

# What is Java?



**Java** เป็นภาษาที่ถูกพัฒนาขึ้นเพื่อวัตถุประสงค์ทั่วไป เป็นภาษาที่สนับสนุนการเขียนโปรแกรมแบบเชิงวัตถุ (Object-Oriented Programming) มีไลบรารีต่าง ๆ อำนวยความสะดวกในการพัฒนา ลดการพัฒนาโปรแกรมน้อยลง เพราะสามารถเขียนเพียงครั้งเดียว แต่สามารถนำไปใช้งานได้หลากหลายแพลตฟอร์ม

# History of Java

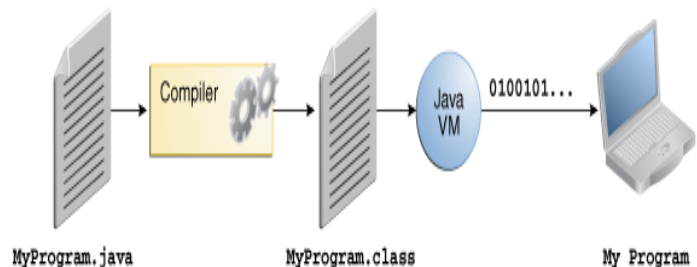


James Gosling

- พัฒนารึ้นในปี ค.ศ. 1991
- เดิมชื่อภาษา "Oak"
- เปลี่ยนชื่อเป็นภาษา Java ใน เวอร์ชัน 1.0.2
- จุดประสงค์ของการพัฒนาภาษา เพื่อให้สามารถทำงานได้ในหลากหลายระบบปฏิบัติการ และหลากหลายเครื่องมือ



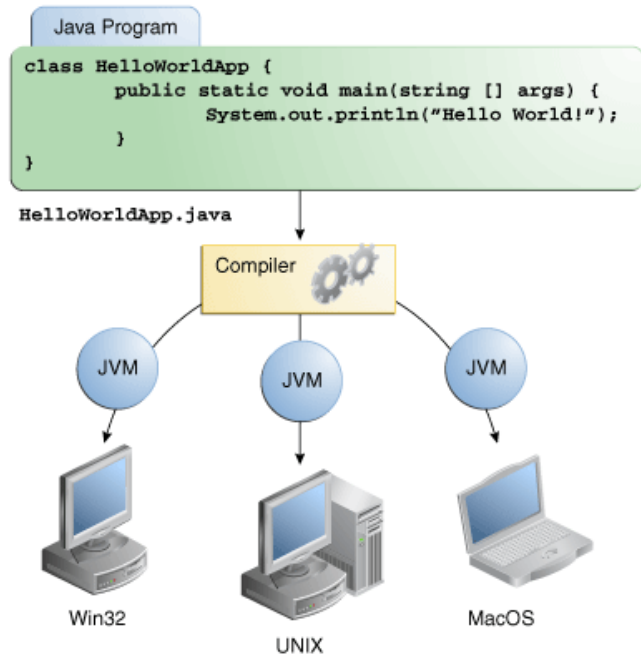
# How does Java Work?



1. เขียนคำสั่งลงในไฟล์ และบันทึกเป็นนามสกุล .java
2. สั่งคอมไพล์เลอร์คอมไพล์ไฟล์ .java สร้างเป็นไฟล์นามสกุล .class ด้วยคำสั่ง `javac`
3. ไฟล์นามสกุล .class ที่ถูกคอมไพล์จะกลายเป็นไฟล์ที่เก็บไบต์โค้ดเอาไว้
4. จาาเวอร์ชัวแมชชีนจะทำหน้าที่รันโปรแกรมนามสกุล .class ให้แสดงผลบนอุปกรณ์



# How does Java Work?



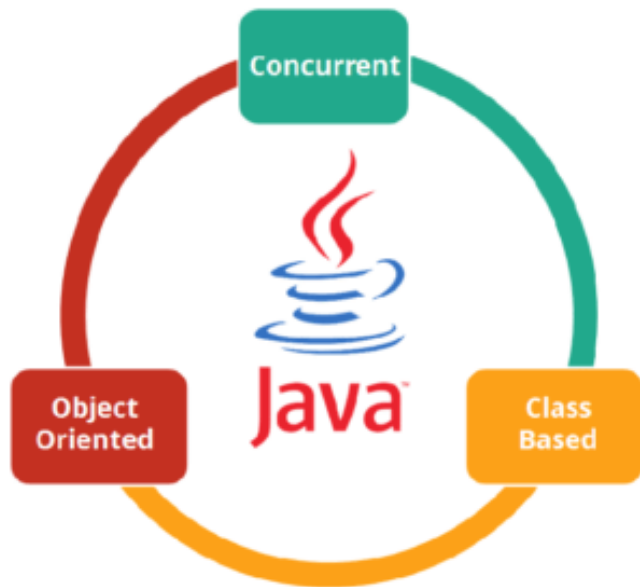
## Java SE Runtime Environment 8u201

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	68.1 MB	<a href="#">jre-8u201-linux-i586.rpm</a>
Linux x86	83.8 MB	<a href="#">jre-8u201-linux-i586.tar.gz</a>
Linux x64	64.91 MB	<a href="#">jre-8u201-linux-x64.rpm</a>
Linux x64	80.73 MB	<a href="#">jre-8u201-linux-x64.tar.gz</a>
Mac OS X x64	76.18 MB	<a href="#">jre-8u201-macosx-x64.dmg</a>
Mac OS X x64	67.77 MB	<a href="#">jre-8u201-macosx-x64.tar.gz</a>
Solaris SPARC 64-bit	46.27 MB	<a href="#">jre-8u201-solaris-sparcv9.tar.gz</a>
Solaris x64	50.14 MB	<a href="#">jre-8u201-solaris-x64.tar.gz</a>
Windows x86 Online	1.87 MB	<a href="#">jre-8u201-windows-i586-iftw.exe</a>
Windows x86 Offline	63.53 MB	<a href="#">jre-8u201-windows-i586.exe</a>
Windows x86	66.51 MB	<a href="#">jre-8u201-windows-i586.tar.gz</a>
Windows x64	71.44 MB	<a href="#">jre-8u201-windows-x64.exe</a>
Windows x64	71.29 MB	<a href="#">jre-8u201-windows-x64.tar.gz</a>

# Feature of Java



- **Concurrent:** สามารถรองรับการทำงานได้หลาย ๆ คำสั่งพร้อมกัน
- **OOP-based:** ภาษาถูกพัฒนามาบนหลักการการพัฒนาโปรแกรมเชิงวัตถุ ทำให้ซอฟต์แวร์ที่พัฒนามีโครงสร้างที่ดี
- **Independent:** เป็นอิสระไม่ผูกมัดกับแพลตฟอร์มใด สามารถเขียนโปรแกรมเพียงครั้งเดียว แต่สามารถทำงานได้หลากหลายแพลตฟอร์ม

# Compare version

Java SE 5	Java SE 6	Java SE 7	Java SE 8
Generic*	<b>No new features,</b> Improve libraries	Switch on Strings*	Lambda expression
Annotation		Try-with-resources	Annotation on Java types
Enum types*		Diamond operator*	
Varargs		New abstract file system (NIO2)	Improve Date and Time API
Enhanced For Loop*			
Concurrency utilities			



# ทำไปพร้อมๆ กัน - Hello World!

## Java source

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

## Hello World - Anatomy

**class HelloWorld:** ตั้งชื่อคลาส "HelloWorld"

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

**public:** เป็น modifier ให้คลาสอื่น ๆ สามารถเรียกใช้งานได้

**static:** เรียกใช้งานเมธอดได้โดยตรง ไม่จำเป็นต้องสร้างคลาสเพื่อเรียกใช้งาน ในกรณีนี้เมธอด main จะถูกเรียกให้ทำงานโดย JVM

**main:** เป็นเมธอดที่ให้โปรแกรมทำงาน

**String[] args:** เป็นการระบุค่าประเภทข้อความ เพื่อให้ข้อความแสดงผลบน console

**System.out.println:** เมธอด println ถูกเรียกใช้งานเพื่อให้แสดงผลข้อความ และทำการขึ้นบรรทัดใหม่  
System เป็นคลาส และ out เป็น object ของคลาส PrintStream

# นามสกุลไฟล์ของ Java ที่พบบ่อย

นามสกุลไฟล์	ความหมาย และการนำไปใช้
.java	ไฟล์ Source code ของภาษา Java
.class	ไฟล์ bytecode ของภาษา Java จะได้จากการ compile ไฟล์ .java
.jar	(Java Archive) เป็นได้ทั้งไฟล์ execute เพื่อให้โปรแกรมสามารถทำงานได้ หรือทำหน้าที่เหมือน zip ไฟล์ เก็บรวบรวมไฟล์ .class ทั้งหมดเอาไว้
.mf	ไฟล์ MANIFEST เป็นไฟล์ที่บอกข้อมูลของไฟล์ .jar ปรกติจะถูกสร้างขึ้นตอนสร้าง package และเก็บไว้ใน .jar ไฟล์
.war	(Web Application Archive) เปรียบเสมือน zip ไฟล์ แต่เก็บไฟล์ทั้งหมดที่เกี่ยวข้องทั้งหมดของ web application



Software Park Thailand  
</Code Camp>

# Agenda

1.วิธีการติดตั้งโปรแกรม **Eclipse**

2.**Basic Syntax**

3.Operator

4.Control Flow

5.Method

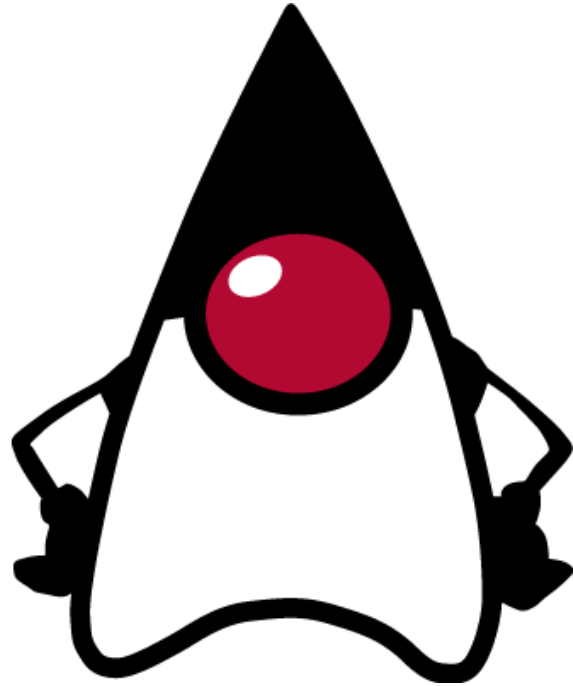
6.Strings

7.Arrays

8.Maven

9.Homework

# Basic Syntax





# Comments

## Java source

## วิธีการใส่คอมเมนต์ใน Java

```
/*
```

**Multi lines comment**

```
*/
```

**//Single line comment**

```
/*
First Java Program
*/
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World"); //Print out Hello World
        //System.out.println("Hello Mars");

        /*
        System.out.println("Hello Moon");
        System.out.println("Hello Venus");
        */
    }
}
```

# How does variable work?

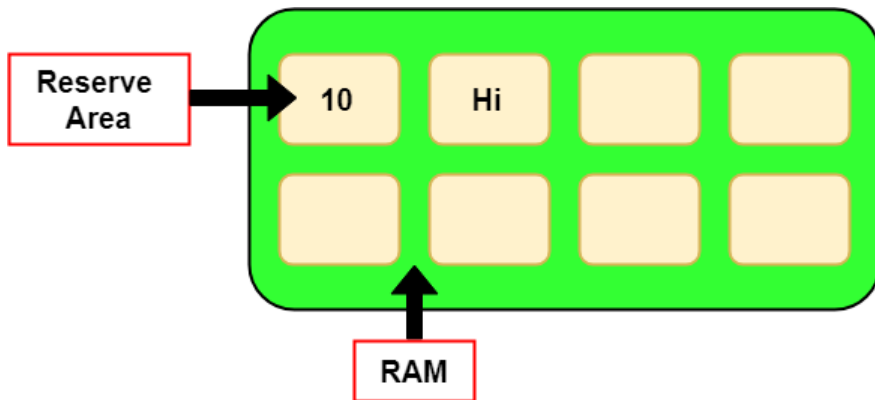
## Java source

```
int number = 10;  
String greeting = "Hi";
```

ประเภทอื่นๆ

double  
float  
boolean

## RAM





# Variable declaration

```
class Variable {  
    public static void main(String[] args) {  
        /* Data type: integer  
        Variable name: myInt  
        */  
        int myInt;  
        // Assign value to myInt = 20;  
        myInt = 20;  
        System.out.println("myInt value: " + myInt);  
        // Declare and assign  
        int myNewInt = 50;  
        System.out.println("myNewInt value: " +  
            myNewInt);  
    }  
}
```

## Syntax

<data type> [variable name] = value (option)

## การตั้งชื่อตัวแปร

- ตั้งชื่อแบบ Camel case เช่น **new**Variable, **first**String
- ตั้งชื่อให้สื่อความหมาย e.g. randomInt, displayNumber
- สามารถตั้งชื่อด้วย underscore (\_) หรือ dollar sign นำหน้าได้ เช่น \_nextScore, \$totalPrice
- ไม่สามารถ** ตั้งชื่อด้วยคำสงวน เช่น static หรือ new
- ไม่สามารถ** ตั้งด้วยตัวเลขนำหน้าได้ เช่น **1**num

# Lab 1 - ทำพร้อมกัน

1. ลอง Comment ทั้งแบบ Single Line และ Multiple Line
2. ประกาศตัวแปรประเภทต่างๆ
  - a. int
  - b. boolean
  - c. String
  - d. (Optional) หาตัวแปรประเภทอื่นๆ เองอีกสัก 2 ประเภทใน google
3. Print ทุกๆ ตัวแปร จากข้อ 2 ออกมา ในรูปแบบ “This is a **String**” + variable





# Variable Scope

```
class VariableScope {  
    // instance variable  
    public static int instanceInt = 20;  
  
    public static void main(String[] main) {  
        // local variable  
        int localInt = 20;  
    }  
}
```

## ประเภทของตัวแปร

### Instance variable

- ประกาศภายในคลาส แต่อยู่ภายนอกเมธอด
- ตัวแปรที่ประกาศขึ้นสามารถเรียกใช้งานได้ภายในคลาสที่สร้างขึ้นเท่านั้น ไม่สามารถเรียกใช้งานจาก instance อื่น ๆ ได้

### Local variable

- ประกาศใช้ภายในเมธอด และถูกเรียกใช้งานภายในเมธอดเท่านั้น
- ไม่สามารถระบุด้วยคำสงวน **static**



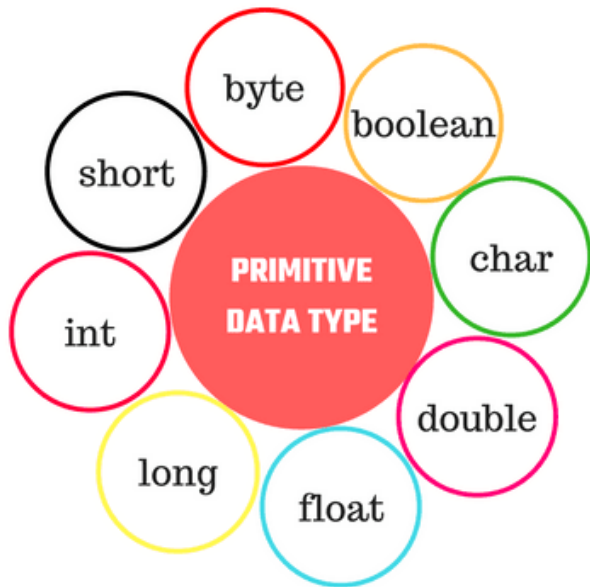
# Example of variable scope

```
class VariableExample {  
    public static int instanceInt = 50; //instance variable  
    public static void main(String[] args) {  
        System.out.println("Instance variable " +instanceInt);  
        myMethod();  
    }  
    public static void myMethod(){  
        int localInt = 20; // local variable  
        System.out.println("Instance variable " +instanceInt);  
        System.out.println("Local variable " + localInt);  
    }  
}
```

## Output

```
Instance variable 50  
Local variable 20
```

# Primitive Data Types



ประเภท	จำนวนพื้นที่ใน RAM	ค่าเริ่มต้น	ขอบเขตข้อมูล
boolean	1 bit of information	false	true / false
byte	8 bits	(byte) 0	-128 to 127
short	16 bits	(short) 0	$-2^{15}$ to $2^{15} - 1$
int	32 bits	0	$-2^{31}$ to $2^{31} - 1$
long	64 bits	0L	$-2^{63}$ to $2^{63} - 1$
char	16 bits unicode char	u0000	0 to 65535
float	32 bits	0.0f	$\pm 10^{38}$
double	64 bits	0.0d	$\pm 10^{308}$



# Example of Primitive Data Types

```
class DataType {  
    public static void main(String[] args) {  
        boolean myBoolean = false;  
        byte myByte = 0;  
        short myShort = 0;  
        int myInt = 0;  
        long myLong = 0L;  
        char myChar = '\u0000';  
        float myFloat = 0.0f;  
        double myDouble = 0.0d;  
    }  
}
```

## Output

```
myBoolean = false  
myByte = 0  
myShort = 0  
myInt = 0  
myLong = 0  
myChar =  
myFloat = 0.0  
myDouble = 0.0
```



# Primitive Conversion

```
class PrimitiveConversion {  
    public static void main(String[] args) {  
        int myInt = 5;  
        float myFloat = myInt; //Widening  
        System.out.println("myInt value: " + myInt);  
        System.out.println("myFloat value: " + myFloat);  
  
        float newFloat = 5.5f;  
        int newInt = (int) newFloat; //Narrowing  
        System.out.println("newFloat value: " + newFloat);  
        System.out.println("newInt value: " + newInt); // 5  
  
        int otherInt = 130;  
        byte myByte = (byte) otherInt; //Overflow  
        System.out.println("otherInt value: " + otherInt);  
        System.out.println("myByte value: " + myByte);  
    }  
}
```

คือการแปลงประเภทของตัวแปร จากประเภทที่กำหนดไว้ ไปอีกประเภทหนึ่ง การแปลงประเภทตัวแปรต้องคำนึงถึงค่าที่กำหนดไว้ให้ตัวแปรด้วย เช่น

1. แปลงค่าที่ระบุ**จุดทศนิยม**ได้ไปเป็น**จำนวนเต็ม** ค่าที่มีจุดทศนิยมจะไม่ถูกแปลงตามไปด้วย (Narrowing)
2. แปลงค่าที่มีเขตข้อมูล**มากกว่า** ไปเป็นข้อมูลที่มีขอบเขต**น้อยกว่า** อาจทำให้ได้รับค่าไม่ถูกต้อง (Overflow)



Software Park Thailand  
</Code Camp>

# Final variable

```
class FinalVar {  
    public static void main(String[] args) {  
        final int finalInt = 20;  
        //finalInt = 20;  
        System.out.println("finalInt value: " + finalInt);  
    }  
}
```

การประกาศ final ให้ตัวแปร เป็นการประกาศว่าจะ  
เป็นค่าจะไม่มีการเปลี่ยนแปลง หากทำการเปลี่ยน  
ค่าตัวแปรที่ประกาศ final ไว้แล้วจะ compile ไม่  
ผ่าน เหมือนการประกาศตัวแปรด้วย const ใน  
JavaScript

## Lab 2 ทำพร้อมๆ กัน - Variable Scope

1. ประกาศ method ชื่อว่า bark ขึ้นมา แล้วเอา method main ไปเรียก method bark โดย method bark ให้มี local variable ชื่อ dogName ที่กำหนดชื่อหมาเป็นอะไรก็ได้ แล้วก็ให้ print ว่า “The Dog name = **xxx** bark”
2. ให้ลองเปลี่ยนค่า primitive แล้ว print ออกมา โดยให้เปลี่ยนจาก
  - a. float -> int
  - b. int -> float
  - c. double -> float
  - d. char -> int
3. ทดลองประกาศ final variable ชื่อ hello ให้มีค่า = “Hello” แล้วบรรทัดถัดมาลองไป Assign ค่าใหม่ ให้ตัวแปรนี้เป็นค่า “World” แล้วลองดู run แล้วดูว่าเกิดไรขึ้น ?



Software Park Thailand  
</Code Camp>

# Agenda

1.วิธีการติดตั้งโปรแกรม **Eclipse**

2.Basic Syntax

3.Operator

4.Control Flow

5.Method

6.Strings

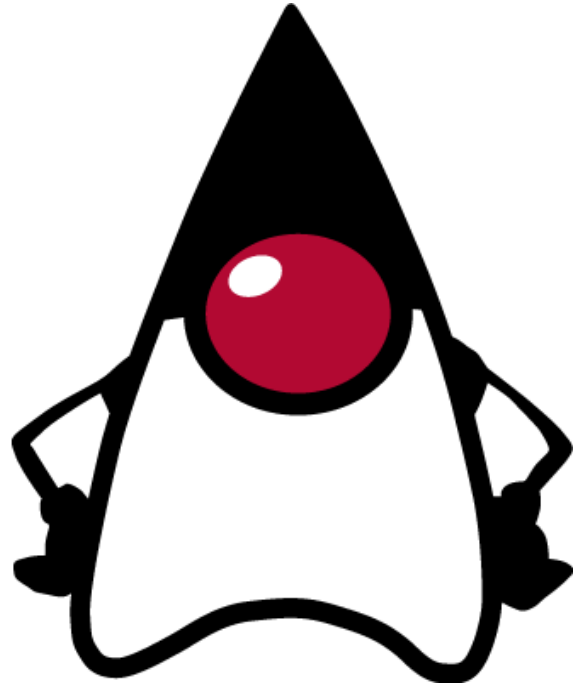
7.Arrays

8.Maven

9.Homework



# Operator





# Unary Operators

```
class Unary {  
    public static void main(String[] args) {  
        int unaryVar = +1;  
        System.out.println("Result " + unaryVar);  
        unaryVar--;  
        System.out.println("Result " + unaryVar);  
        unaryVar++;  
        System.out.println("Result " + unaryVar);  
        unaryVar = -unaryVar;  
        System.out.println("Result " + unaryVar);  
        boolean success = true;  
        System.out.println("Result " + success);  
        System.out.println("Result " + !success);  
    }  
}
```

## Output

```
Result 1  
Result 0  
Result 1  
Result -1  
Result true  
Result false
```

## Unary Operators

เครื่องหมาย	คำอธิบาย
+	แปลงตัวแปรเป็นค่าบวก
-	แปลงตัวแปรเป็นค่าลบ
++	เพิ่มจำนวนข้อมูล 1 ค่า
--	ลดจำนวนข้อมูล 1 ค่า
!	เปลี่ยนข้อมูลตรรกะ !(จริง) = เท็จ !(เท็จ) = จริง



# Prefix - Suffix

```
class PrePost {  
    public static void main(String[] args) {  
        int i = 3;  
        i++;  
        System.out.println(i);  
        ++i;  
        System.out.println(i);  
        System.out.println(++i);  
        System.out.println(i);  
    }  
}
```

## Output

```
4  
5  
6  
6  
7
```

### Prefix - ( ++/-- <variable>)

- เพิ่ม/ลด ค่าให้ตัวเอง 1 ค่าก่อนเริ่มทำงาน

### Suffix - ( <variable>++/--)

- เริ่มทำงานก่อน หลังจากนั้นเพิ่ม/ลด ค่าให้ตัวเอง 1 ค่า

# Comparison Operators

```
class Comparison {  
    public static void main(String[] args) {  
        int value_1 = 1;  
        int value_2 = 2;  
        if (value_1 == value_2)  
            System.out.println("value_1 == value_2");  
        if (value_1 != value_2)  
            System.out.println("value_1 != value_2");  
        if (value_1 > value_2)  
            System.out.println("value_1 > value_2");  
        if (value_1 < value_2)  
            System.out.println("value_1 < value_2");  
        if (value_1 <= value_2)  
            System.out.println("value_1 <= value_2");  
    }  
}
```

## Output

```
value_1 != value_2  
value_1 < value_2  
value_1 <= value_2
```

## Comparison Operators

เครื่องหมาย	คำอธิบาย
==	เท่ากับ
!=	ไม่เท่ากับ
>	มากกว่า
>=	มากกว่า หรือเท่ากับ
<	น้อยกว่า
<=	น้อยกว่า หรือเท่ากับ



# Conditional Operators

```
class Conditional {  
    public static void main(String[] args) {  
        int value_1 = 1;  
        int value_2 = 2;  
        if ((value_1 == 1) && (value_2 == 2))  
            System.out.println("value1 is 1 AND value2 is 2");  
        if ((value_1 == 1) || (value_2 == 1))  
            System.out.println("value1 is 1 OR value2 is 1");  
    }  
}
```

## Conditional Operators

เครื่องหมาย	คำอธิบาย
&&	และ
	หรือ

## Output

```
value_1 is 1 AND value_2 is 2  
value_1 is 1 OR value_2 is 1
```

## Lab 3 ทำพร้อมๆ กัน

1. ประกาศ `int i` ให้มีค่า = 20
  - a. เรียก `i++` 5 ครั้ง แล้ว print ค่า `i` ออกมา
  - b. เรียก `--i` 5 ครั้ง แล้ว print ค่า `i` ออกมา
2. ทดลองใช้ `&&` กับข้อมูลประเภท `float`
3. ทดลองใช้ `||` กับข้อมูลประเภท `char`



Software Park Thailand  
</Code Camp>

# Agenda

1.วิธีการติดตั้งโปรแกรม **Eclipse**

2.Basic Syntax

3.Operator

4.**Control Flow**

5.Method

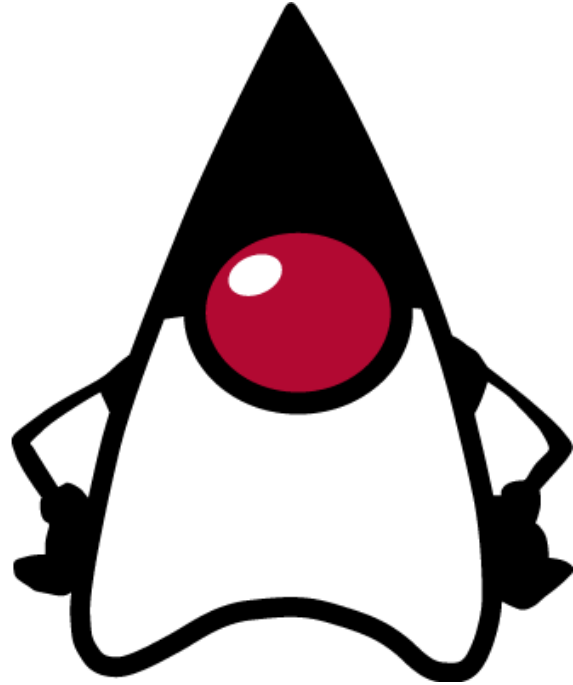
6.Strings

7.Arrays

8.Maven

9.Homework

# Control Flow





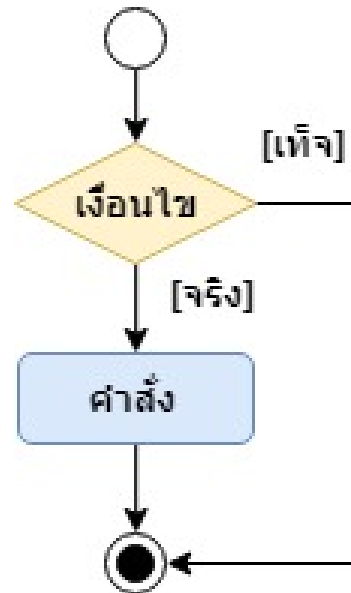


# If Statement

```
class IfExample {  
    public static void main(String[] args) {  
        int peopleAge = 20;  
        if ( peopleAge > 18 )  
            System.out.println("You can vote.");  
    }  
}
```

## Output

```
You can vote.
```



## Syntax

```
if (เงื่อนไข) {  
    คำสั่ง  
}
```

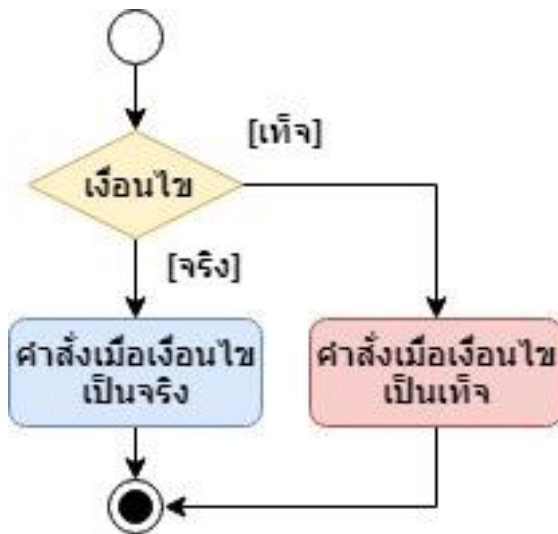


# If-else Statement

```
public class IfElseExample {  
    public static void main(String[] args) {  
        int number = 5;  
        if (number % 2 == 0) {  
            System.out.println("Even number");  
        } else {  
            System.out.println("Odd number");  
        }  
    }  
}
```

Output

Odd number



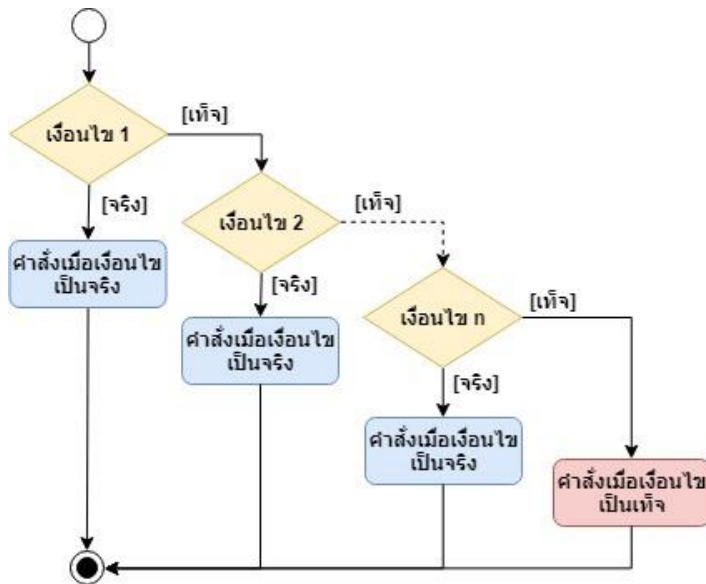
Syntax

```
if (เงื่อนไข) {  
    คำสั่งเมื่อเงื่อนไขเป็นจริง  
} else {  
    คำสั่งเมื่อเงื่อนไขเป็นเท็จ  
}
```



# If-else-if ladder Statement

```
public class IfElseIfExample {  
    public static void main(String[] args) {  
        int marks = 50;  
        if (marks < 60) {  
            System.out.println("fail");  
        } else if (marks >= 60 && marks < 70) {  
            System.out.println("C grade");  
        } else if (marks >= 70 && marks < 80) {  
            System.out.println("B grade");  
        } else if (marks >= 80 && marks < 100) {  
            System.out.println("A grade");  
        } else {  
            System.out.println("Invalid!");  
        }  
    }  
}
```



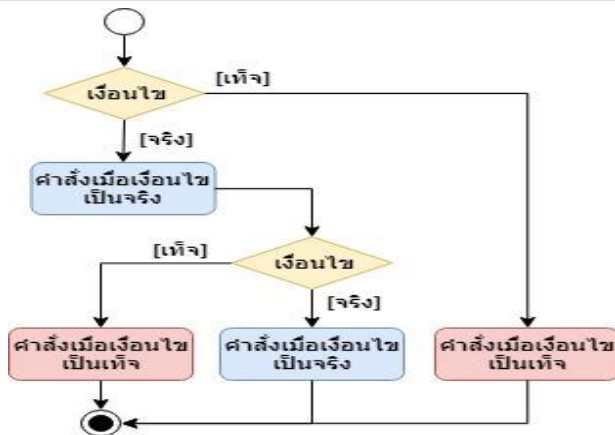
## Syntax

```
if (เงื่อนไข 1) {  
    คำสั่งเมื่อเงื่อนไขเป็นจริง  
} else if (เงื่อนไข 2) {  
    คำสั่งเมื่อเงื่อนไขเป็นเท็จ  
} else if (เงื่อนไข n) {  
    ....  
} else {  
    คำสั่งเมื่อเงื่อนไขเป็นเท็จ  
}
```



# Nested if Statement

```
public class NestedIfExample {  
    public static void main(String[] args) {  
        int age = 25;  
        int weight = 48;  
        if (age >= 18) {  
            if (weight > 50) {  
                System.out.println("OK");  
            } else {  
                System.out.println("Not OK");  
            }  
        } else {  
            System.out.println("Age should greater than 18");  
        }  
    }  
}
```



## Syntax

```
if (เงื่อนไข) {  
    if (เงื่อนไข) {  
        คำสั่งเมื่อเงื่อนไขเป็นจริง  
    } else {  
        คำสั่งเมื่อเงื่อนไขเป็นเท็จ  
    }  
} else {  
    คำสั่งเมื่อเงื่อนไขเป็นเท็จ  
}
```

# Lab 4 ทำพร้อมๆ กัน

1. ประกาศตัวแปร score = อะไรก็ได้
  - a. ถ้า score  $\geq 80$  ให้ print ว่า Good
  - b. ถ้า score  $\geq 50$  แต่น้อยกว่า 80 ให้ print ว่า normal
  - c. ถ้า score  $< 50$  ให้ print ว่า fail
2. จากข้อ 1 ให้ประกาศตัวแปร boolean ว่า isHandSome เพิ่มเข้ามาเป็นค่าอะไรก็ได้
  - a. แก้อัปเดตจาก ข้อ 2 ถ้าเกิด isHandSome = true แล้ว ไม่ว่าจะได้ score เท่าไหร่ ก็ให้ print Good ตลอด



Software Park Thailand  
</Code Camp>

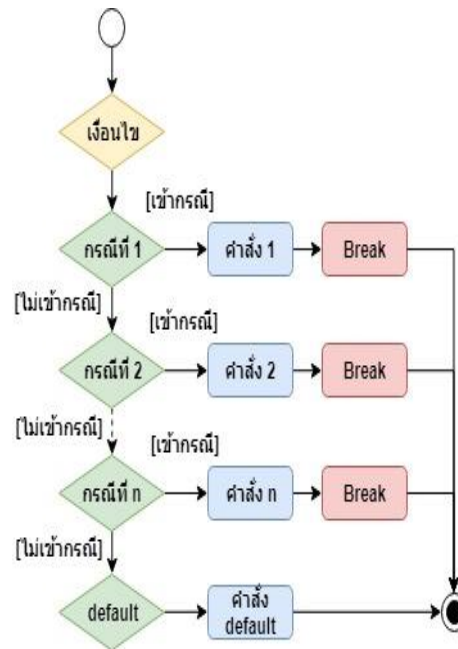
# Hotkey สำหรับ Eclipse ที่ใช้บ่อย

- พิมพ์ sysout แล้วกด control + space จะแปลงร่างเป็น System.out.println();
- เมื่อกด control + space หลังจุด มันจะมี auto complete โผล่ออกมา ทำให้เราเขียน code ได้ง่ายขึ้น
- กด control + shift + f จะทำการ format code ทำให้ code อ่านง่ายขึ้น มักใช้ก่อน commit และ push ขึ้น server



# Switch Statement

```
public class SwitchExample {  
    public static void main(String[] args) {  
        int number = 20;  
        switch (number) {  
            case 10:  
                System.out.println("10"); break;  
            case 20:  
                System.out.println("20"); break;  
            case 30:  
                System.out.println("30"); break;  
            default:  
                System.out.println("Not in 10, 20 or 30");  
        }  
    }  
}
```



## Syntax

```
switch (เงื่อนไข) {  
    case (กรณีที่ 1) :  
        คำสั่ง; break;  
    case (กรณีที่ 2) :  
        คำสั่ง; break;  
    case (กรณีที่ n) :  
        คำสั่ง; break;  
    default:  
        คำสั่ง default  
}
```



# Switch Statement more examples

```
class SwitchTime {  
    public static void main(String[] args) {  
        int mealTime = 3;  
        String mealString = "";  
        switch (mealTime) {  
            case 1:  
                mealString = "1: Breakfast"; break;  
            case 2:  
                mealString = "2: Lunch"; break;  
            case 3:  
                mealString = "3: Dinner"; break;  
            case 4:  
                mealString = "4: Snack"; break;  
            default:  
                System.out.println("Invalid meal time!");  
        }  
        System.out.println(mealString);  
    }  
}
```

```
class SwitchString {  
    public static void main(String[] args) {  
        String levelString = "Expert";  
        int level = 0;  
        switch (levelString) {  
            case "Beginner":  
                level = 1; break;  
            case "Intermediate":  
                level = 2; break;  
            case "Expert":  
                level = 3; break;  
            default:  
                level = 0;  
        }  
        System.out.println("Your Level is: " + level);  
    }  
}
```





Software Park Thailand  
</Code Camp>

# Lab 5. Switch case

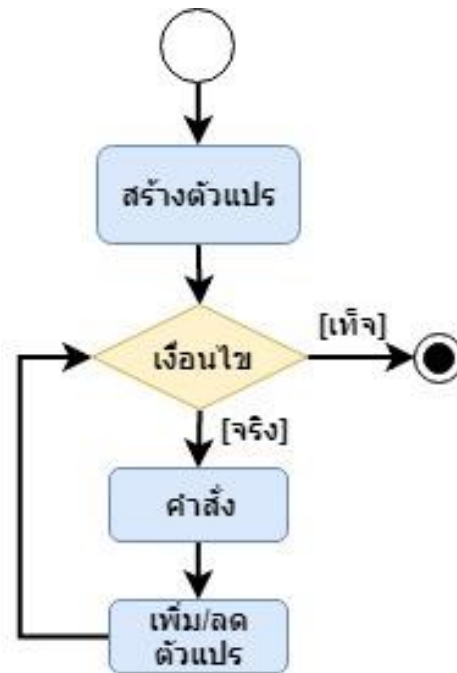
1. สร้างโปรแกรมตัดเกรด โดยที่
  - a. เกรด A ถ้าคะแนน = 80
  - b. เกรด B ถ้าคะแนน = 70
  - c. เกรด C ถ้าคะแนน = 60
  - d. เกรด D ถ้าคะแนน = 50
  - e. เกรด F ถ้าคะแนน = 40
  - f. เกรด E ถ้าคะแนนเป็นค่าอื่นๆ

# For Loop

## Syntax

```
for (สร้างตัวแปร; เงื่อนไข; เพิ่ม/ลด ตัวแปร) {  
    คำสั่ง  
}
```

```
class ForLoop {  
    public static void main(String[] args) {  
        for(int counter = 0; counter <= 10; counter++) {  
            System.out.println("Counter :" + counter);  
        }  
    }  
}
```





Software Park Thailand  
</Code Camp>

# For-each Loop

## Syntax

```
for ( <type> variable : Array | Collection ) {  
    คำสั่ง  
}
```

```
class ForEach {  
    public static void main(String[] args) {  
        int myArray[] = {1,2,3,4,5};  
        for(int counter : myArray) {  
            System.out.println("Counter :" + counter);  
        }  
    }  
}
```



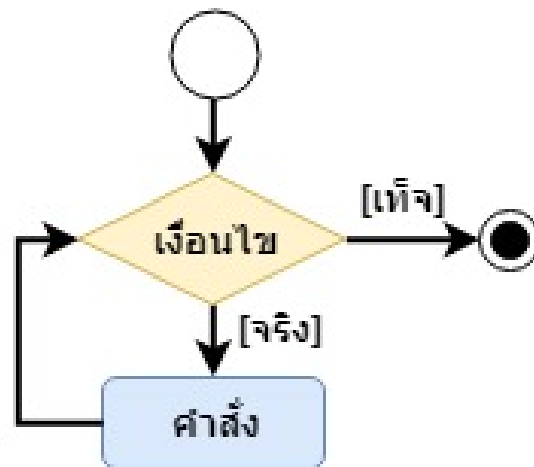
Software Park Thailand  
</Code Camp>

# While Loop

## Syntax

```
while (เงื่อนไข) {  
    คำสั่ง  
}
```

```
class While {  
    public static void main(String[] args) {  
        int counter = 0;  
        while (counter < 5) {  
            System.out.println("Counter" + counter);  
            counter++;  
        }  
    }  
}
```



## Lab 6. ทำพร้อมๆกัน While Loop

1. แสดงค่า  $i$  ในแต่ละรอบ โดยการประกาศตัวแปร  $i$  มาใช้ในการนับ ในเงื่อนไขของ while ถ้าหาก  $i$  น้อยกว่าหรือเท่ากับ 10 `while ( i <= 10 )` แสดงค่า  $i$  ออกมา แต่ละรอบ (  $i$  มีค่าเริ่มต้นเป็น 1 )
2. สร้างโปรแกรมหาผลรวมของตัวเลข 1 ถึง 10
3. สร้างโปรแกรมหาค่าระหว่าง 1-100 ที่หาร 12 ลงตัว
4. ให้ประกาศ array ที่มีค่า [1,2,3,4,5] แล้วใช้ foreach ในการวน loop แล้ว print ค่าออกมาทั้งหมด



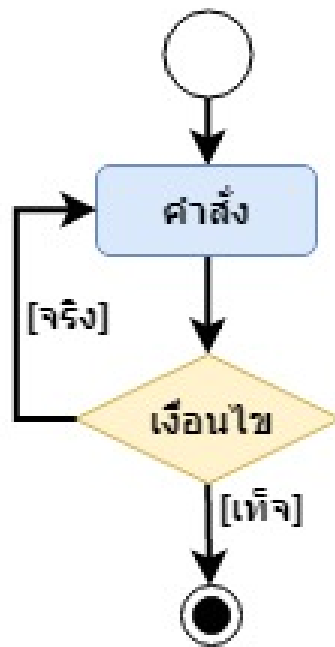
Software Park Thailand  
</Code Camp>

# Do While Loop

## Syntax

```
do {  
    คำสั่ง  
} while (เงื่อนไข)
```

```
class DoWhile {  
    public static void main(String[] args) {  
        int counter = 0;  
        do {  
            System.out.println("Counter : " + counter);  
            counter++;  
        } while (counter <= 5);  
    }  
}
```





Software Park Thailand  
</Code Camp>

# Lab 7. Do While Loop

1. print ค่าตัวเลข 20 โดยเริ่มจากมากไปน้อย
2. เขียนโปรแกรมตรวจสอบตัวเลข ว่าเป็นจำนวนคู่ หรือจำนวนคี่ โดยโปรแกรมจะทำงานในลูปเรื่อยๆ ถ้าหากเขายังคงกรอกเลขคู่ แต่ถ้ากรอกเลขคี่จะเป็นการออกจากลูป



# Break

## Syntax

เงื่อนไขให้หยุด Loop  
**break;**

```
class Break {  
    public static void main(String[] args) {  
        for (int counter = 0; counter <= 10; counter++) {  
            if (counter == 5)  
                break;  
            System.out.println("Counter : " + counter);  
        }  
    }  
}
```



## Lab 8. Break กับ For

1. กำหนดตัวแปร count สำหรับนับจำนวนรอบ
  - a. count เริ่มต้นที่ 0 และไปจบที่ 20 (รวม 20 ด้วย)
  - b. ถ้า count มีค่าเป็น 11 คำสั่ง break ภายใน if จะทำงาน
  - c. ให้แสดงค่า 11 และออกจาก loop โดยไม่ทำคำสั่งด้านล่างต่อ



# More Break

```
class MoreBreak {  
    public static void main(String[] args) {  
        for (int counter_1 = 0; counter_1 < 5; counter_1++) {  
            for (int counter_2 = 0; counter_2 < 3; counter_2++) {  
                if ( counter_1 == 2 && counter_2 == 2 )  
                    break;  
                System.out.println("Counter 1: " + counter_1 + " " + "Counter 2: " + counter_2);  
            }  
        }  
    }  
}
```



Software Park Thailand  
</Code Camp>

# Continue

## Syntax

เงื่อนไขที่ให้ Loop ข้ามคำสั่ง  
**continue;**

```
class Continue {  
    public static void main(String[] args) {  
        for (int counter = 0; counter < 5; counter++ ) {  
            if (counter == 2)  
                continue;  
            System.out.println("Counter :" + counter);  
            System.out.println("Hello");  
        }  
    }  
}
```

## Lab 9. Continue กับ For

1. กำหนดตัวแปร count สำหรับนับจำนวนรอบ
  - a. count เริ่มต้นที่ 0 และไปจบที่ 20 (รวม 20 ด้วย)
  - b. ถ้า count มีค่าเป็น 11 คำสั่ง continue ภายใน if จะทำงาน
  - c. ให้แสดงค่า 11 และเริ่มต้นรอบใหม่โดยไม่สนใจคำสั่งที่เหลือด้านล่าง



# More Continue

```
class MoreContinue {  
    public static void main(String[] args) {  
        for (int counter_1 = 0; counter_1 < 5; counter_1++) {  
            for (int counter_2 = 0; counter_2 < 3; counter_2++) {  
                if ( counter_1 == 2 && counter_2 == 2 )  
                    continue;  
                System.out.println("Counter 1: " + counter_1  
                    + " " + "Counter 2: " + counter_2);  
            }  
        }  
    }  
}
```



Software Park Thailand  
</Code Camp>

# Lab 10. ทำพร้อมๆกับ Continue

1. ให้สร้างโปรแกรมวน loop 10 รอบ โดยมีค่าเริ่มต้นที่ 1 และไปจบ 10 ให้แสดงทุกค่ายกเว้นค่า 2 โดยใช้คำสั่ง continue



Software Park Thailand  
</Code Camp>

# Agenda

1.วิธีการติดตั้งโปรแกรม **Eclipse**

2.Basic Syntax

3.Operator

4.Control Flow

5.**Method**

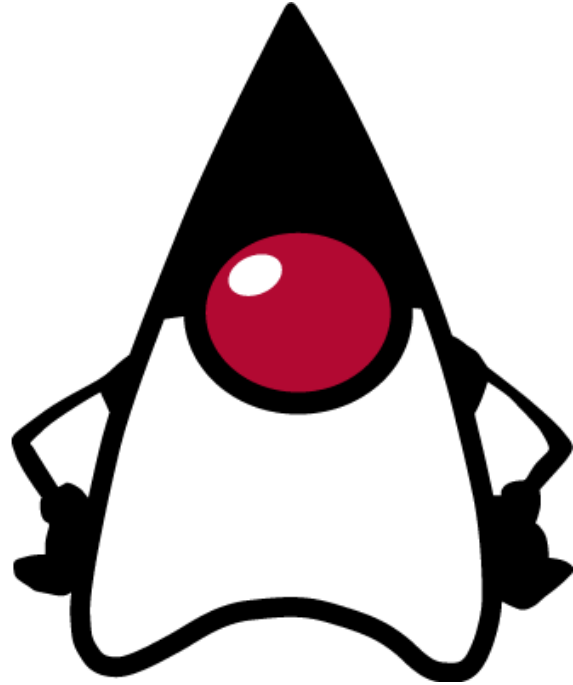
6.Strings

7.Arrays

8.Maven

9.Homework

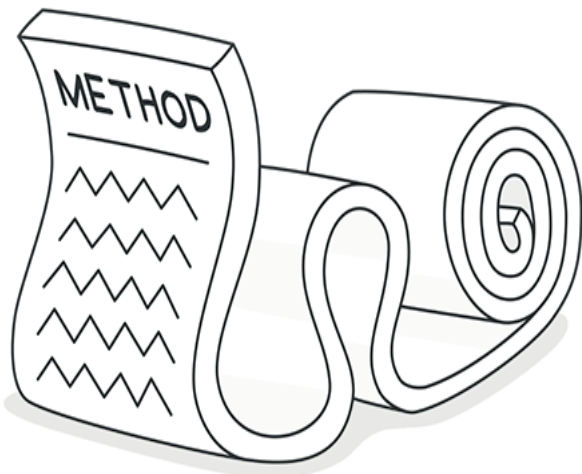
# Method







# What is method?



**Method** (เมธอด) เป็นวิธีการจัดกลุ่มของข้อมูลให้เป็นส่วนย่อย เพื่อให้ง่ายต่อการจัดกลุ่ม วิธีการใช้งานเมธอดมีทั้งแบบสร้างขึ้นเพื่อใช้งาน หรือเรียกใช้จาก library ของจาวา

## ข้อดีของเมธอด

1. ลดความซับซ้อนของโปรแกรม ทำให้ทำความเข้าใจโปรแกรมได้ง่ายขึ้น
2. ง่ายต่อการแบ่งโปรแกรมออกเป็น ส่วน ๆ แยกการทำงานออกจากกัน เพื่อลดทอนส่วนการทำงานโปรแกรม
3. สามารถนำเมธอดที่เขียนไปแล้ว กลับมาใช้งานใหม่ได้อีกครั้ง



# Return value method

## Create return value method Syntax

```
<modifier> <return type> method name (<type> name 1,  
... < type> name n ) {  
    return value;  
}
```

```
public static String greeting (String  
word) {  
    return word;  
}
```

```
class ReturnValMethod {  
    public static void main(String[] args) {  
        String myWord = "Hello World";  
        System.out.println(greeting(myWord));  
    }  
    public static String greeting(String word) {  
        return word;  
    }  
}
```



# Void method

## Create void method Syntax

```
<modifier> void method name (<type> name 1, ... < type> name n ) {  
}
```

```
public static void sayHello (String  
word) {  
    System.out.println(word);  
}
```

```
class NotReturnValMethod {  
    public static void main(String[] args) {  
        String greeting = "Hello World";  
        sayHello(greeting);  
    }  
    public static void sayHello(String word) {  
        System.out.println(word);  
    }  
}
```



# Method overloading

```
class OverloadMethod {  
    public static void main(String[] args) {  
        int a = 11, b = 10;  
        String c = "Hello", d = "world";  
        int addedInt = addition(a, b);  
        String addedString = addition(c, d);  
        System.out.println("addedInt: " + addedInt);  
        System.out.println("addedString: " + addedString);  
    }  
  
    public static int addition(int n1, int n2) {  
        return n1 + n2;  
    }  
  
    public static String addition(String n1, String n2) {  
        return n1 + " " + n2;  
    }  
}
```

# Lab: 11 - ทดลองสร้าง Method



ทดลองสร้าง method แบบ Return และไม่  
Return ค่า และเรียกใช้งาน method ที่สร้างขึ้น  
โดยให้แสดงผลที่ Console



Software Park Thailand  
</Code Camp>

# Agenda

1.วิธีการติดตั้งโปรแกรม **Eclipse**

2.Basic Syntax

3.Operator

4.Control Flow

5.Method

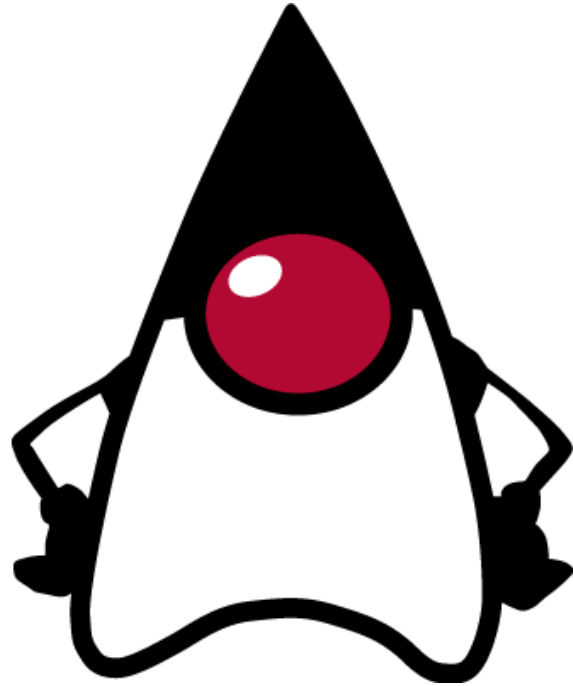
6.Strings

7.Arrays

8.Maven

9.Homework

# Strings





# What is String?

```
class StringExample {  
    public static void main(String[] args) {  
        char[] myChar = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd' };  
        String myWord = new String(myChar);  
        System.out.println(myWord);  
    }  
}
```

**String** คือข้อมูลประเภท char ที่นำมาเรียง  
ข้อมูลกันทั้งหมด ข้อมูลประเภท String ไม่  
นับว่าเป็น primitive type แต่เป็นออบเจ็กต์  
หนึ่ง



# Lab ทำพร้อมกัน

- สร้าง String จาก char อะไรก็ได้



# String .equals() method

```
class CompareStrings {  
    public static void main(String[] args) {  
        String myString = "FirstString";  
        String compareString_1 = "FirstString";  
        String compareString_2 = "firstString";  
        if ( myString.equals(compareString_1) )  
            System.out.println( "Equal" );  
        else  
            System.out.println("Not equal");  
        if ( myString.equals(compareString_2) )  
            System.out.println( "Equal" );  
        else  
            System.out.println("Not equal");  
    }  
}
```

**.equals()** เป็นเมธอดที่ใช้เปรียบเทียบข้อมูลประเภท String  
ผลลัพธ์ที่ได้เป็น boolean

**\* ไม่ควรใช้ == ในการเปรียบเทียบข้อมูลประเภท String** เพราะเป็นการเปรียบเทียบว่าเป็นข้อมูลที่มาจกตำแหน่งอ้างอิงเดียวกันใน Memory หรือไม่



# String .equals() method

```
class CompareTwoStrings { // Compare string with ==
    public static void main( String[] args ) {
        String firstString = "HelloWorld";
        String secondString = "HelloWorld";
        if (firstString == secondString)
            System.out.println("True");
        else
            System.out.println("False");
        if (firstString == (new String(secondString)))
            System.out.println("True");
        else
            System.out.println("False");
    }
}
```

**\* ไม่ควรใช้ == ในการเปรียบเทียบข้อมูลประเภท String** เพราะเป็นการเปรียบเทียบว่าเป็นข้อมูลที่มาจากการตำแหน่งอ้างอิงเดียวกันใน Memory หรือไม่



# String .contains() method

```
public class Employee {  
    public String firstname;  
    public String lastname;  
    private int age;
```

**.contains()** เป็นเมธอดที่ใช้ค้นหาคำที่อยู่ในข้อมูล  
String ผลลัพธ์ที่ได้เป็น boolean

```
    public Employee(String firstNameInput, String lastNameInput, int ageInput) {  
        firstname = firstNameInput;  
        lastname = lastNameInput;  
        age = ageInput;  
        int temp = 123;  
    }  
    public void hello() {  
        System.out.println("Hello " + firstname );  
    }  
    public int getAge() {  
        return age;  
    }  
}
```



# String .length() method

```
class MyClass {  
    public static void main(String[] args) {  
        String word_1 = "JavaScript";  
        String word_2 = "สวัสดีวันอาทิตย์";  
        System.out.println("Word 1 length : " + word_1.length());  
        System.out.println("Word 2 length : " + word_2.length());  
    }  
}
```

**.length()** เป็นเมธอดที่ใช้ค้นหาความยาวทั้งหมดของ String รวมถึงช่องว่าง (นับว่าเป็นตัวอักษรตามนิยามของ char) ผลลัพธ์ที่ได้เป็น int



# String .substring() method

```
class CutString {  
    public static void main(String[] args) {  
        String word = "Java is fun!";  
        System.out.println(word.substring(3));  
        System.out.println(word.substring(0, 4));  
        System.out.println(word.substring(0, 20));  
    }  
}
```

**.substring(startIndex, endIndex)** เป็นเมธอดที่ใช้**แบ่งคำ**จากตัวแปรประเภท String โดยคำที่ตัดออกมาขึ้นอยู่กับช่วงของคำที่ต้องการตัดโดยที่พารามิเตอร์แรก เป็นการกำหนดตำแหน่งเริ่มต้น และพารามิเตอร์ที่สอง กำหนดตำแหน่งสุดท้ายในการตัดคำ

**\*ข้อควรระวัง** ไม่สามารถตัดคำได้เกินความยาวของคำที่ต้องการตัด



# String .trim() method

```
class TrimSpace {  
    public static void main(String[] args){  
        String mySpaceWord = "  SPACE SHUTTLE  ";  
        System.out.println(mySpaceWord + " length: " + mySpaceWord.length());  
        String trimWord = mySpaceWord.trim();  
        System.out.println(trimWord + " length: " + trimWord.length());  
    }  
}
```

**.trim(string)** เป็นเมธอดที่ใช้**ตัดช่องว่าง**ที่อยู่  
ทาง**ด้านหน้า** หรือ**ด้านหลัง**ของคำ เพื่อตัดช่องว่าง  
และลดความยาวของคำลง



Software Park Thailand  
</Code Camp>

# String.toUpperCase() method

```
class UpperCaseWord {  
    public static void main(String[] args){  
        String myName = "john doe";  
        System.out.println(myName.toUpperCase());  
    }  
}
```

**.toUpperCase(string)** เป็นเมธอดที่ใช้แปลง  
ตัวอักษรภาษาอังกฤษเป็นตัวพิมพ์ใหญ่ทั้งหมด





# Method chaining

```
class MethodOneByOne {  
    public static void main(String[] args) {  
        String myName = "somchai";  
        System.out.println(setCapital(myName));  
    }  
    static String setCapital(String name) {  
        String firstChar = name.substring(0,1);  
        String upperCasefirstChar =  
firstChar.toUpperCase();  
        String restOfName = name.substring(1);  
        return upperCasefirstChar + restOfName;  
    }  
}
```

```
class MethodChaining {  
    public static void main(String[] args) {  
        String myName = "somchai";  
        System.out.println(setCapital(myName));  
    }  
    static String setCapital(String name) {  
        return name.substring(0,1).toUpperCase()  
            + name.substring(1);  
    }  
}
```

# Lab 12 - ทดลองใช้คำสั่ง

กำหนด String เป็นดังนี้

String1 = 'You and Me' , String2 = ' you and me '

1. ทดลองเปรียบเทียบ String 2 String ว่าเป็นค่าเดียวกันหรือไม่
2. ใช้คำสั่งค้นหาคำใน String และแสดงคำที่ค้นหบบนหน้าจอ
3. ใช้คำสั่งหาความยาวของ String นั้น และแสดงค่าความยาว String
4. ใช้คำสั่งตัดข้อความหรือตัด String ตำแหน่งที่ 1-4 ออก
5. ใช้คำสั่งตัดช่องว่างของประโยค
6. ใช้คำสั่งเปลี่ยน String เป็นพิมพ์ใหญ่ทั้งหมด
7. ใช้คำสั่งเปลี่ยน String2 เป็นพิมพ์ใหญ่ทั้งหมด และ ไม่มีช่องว่างซ้ายขวา ด้วยการเขียน code แคบสทัดเดียว (ใช้ Chaining นั้นเอง)

# Lab: 12 (Optional) - ทดลองแปลงชื่อจริง และนามสกุล



ทดลองสร้าง method และเรียกใช้งาน method ที่สร้างขึ้น ให้แสดงผลลัพท์บน Console เป็นชื่อตัวเอง ดังนี้

ชื่อนามสกุล ตัวอักษรสุดท้ายเป็นตัวพิมพ์ใหญ่

ตัวอย่าง john do**E**



Software Park Thailand  
</Code Camp>

# Agenda

1.วิธีการติดตั้งโปรแกรม **Eclipse**

2.Basic Syntax

3.Operator

4.Control Flow

5.Method

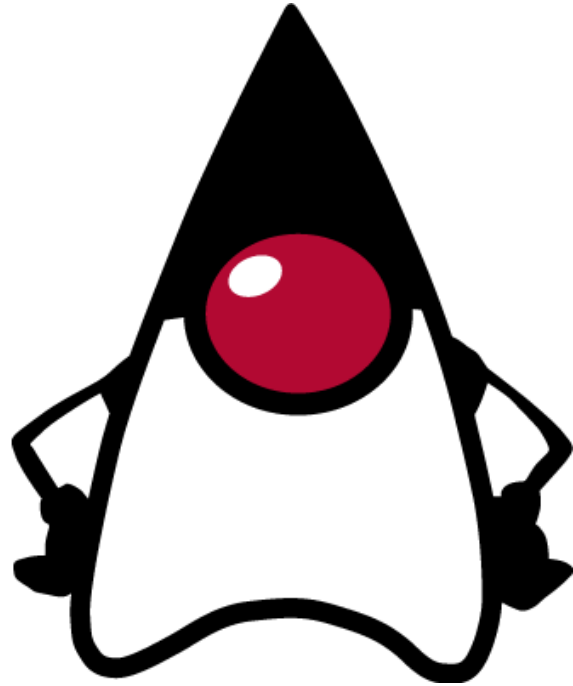
6.Strings

7.Arrays

8.Maven

9.Homework

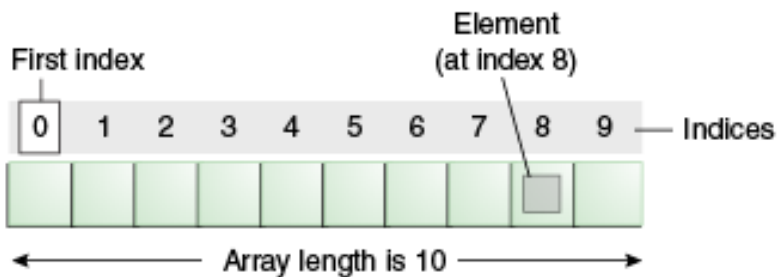
# Arrays





# What is Array?

**Array**(อาเรย์) คือประเภทของข้อมูลที่เป็น**ประเภท** เดียวกัน และมี**ลำดับ** ช่วยในการจัดการข้อมูล โดยที่ไม่ต้องประกาศตัวแปรซ้ำ ๆ



## Syntax

สามารถประกาศได้ดังนี้

1. `<type>[]` ชื่อตัวแปร;
2. `<type>[]` ชื่อตัวแปร = new `<type>[size];`
3. `<type>[]` ชื่อตัวแปร = {ค่า 1, ค่า 2, ค่า n}

```
int[] myArray_1;  
int[] myArray_2 = new int[5];  
int[] myArray_3 = {1, 2, 3, 4, 5};
```



# Initialization in Array

```
class InitArray {  
    public static void main(String[] args) {  
        int[] myIntArray = new int[5];  
        myIntArray[0] = 10;  
        myIntArray[1] = 30;  
        myIntArray[2] = 80;  
        myIntArray[3] = 5;  
        myIntArray[4] = 25;  
        for (int item : myIntArray) {  
            System.out.println("Item :" + item);  
        }  
    }  
}
```

## Syntax

ตัวแปร[ตำแหน่ง Index] = ค่าที่ต้องการระบุ



# Array Length

```
class CheckArrayLength {  
    public static void main(String[] g){  
        int[] intArray = new int [5];  
        char[] charArray = {'A', 'B', 'C', 'D'};  
        String[] StringArray = {"Dog", "Cat"};  
        System.out.println("intArray length: " + intArray.length);  
        System.out.println("charArray length: " + charArray.length);  
        System.out.println("StringArray length: " + StringArray.length);  
    }  
}
```

**.length** เป็นคำสั่งที่ใช้ตรวจสอบความยาวของ  
อาเรย์





# Passing Arrays to Methods

```
class PassingArrayToLoop {  
    public static void main(String[] args) {  
        int[] myArray = {1, 2, 3, 4, 5};  
        printArray(myArray);  
    }  
    public static void printArray(int[] _array) {  
        for (int item: _array) {  
            System.out.println(item);  
        }  
    }  
}
```



# 2-Dimensional Array

## Syntax

**<type>[ ][ ] ตัวแปร = new <type>[แถว][คอลัมน์];**

```
class TwoDimensionArray {  
    public static void main(String[] args) {  
        int[][] a = new int[3][4];  
        System.out.println("Number of row: " + a.length);  
        System.out.println("Number of column: " + a[0].length);  
    }  
}
```

	Column 1	Column 2	Column 3	Column 4
Row 1	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 2	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 3	a[2][0]	a[2][1]	a[2][2]	a[2][3]



# Loop 2-Dimensional Array (1)

```
class Loop2DimensionArray_1 {  
    public static void main(String[] args) {  
        int[][] twoD_Array = {  
            {1, 2, 3},{4, 5, 6, 7},{8, 9}  
        };  
        for (int row = 0; row < twoD_Array.length; row++){  
            for (int element = 0; element < twoD_Array[row].length; element++){  
                System.out.println(twoD_Array[row][element]);  
            }  
        }  
    }  
}
```

# Loop 2-Dimensional Array (2)

```
class Loop2DimensionArray_2 {  
    public static void main(String[] args) {  
        int[][] twoD_Array = {  
            {1, 2, 3},{4, 5, 6, 7},{8, 9}  
        };  
        for (int[] row : twoD_Array) {  
            for (int element: row) {  
                System.out.println(element);  
            }  
        }  
    }  
}
```

# Lab: 13 - ทดลองสร้างอาเรย์ 2 มิติ



1. ทดลองสร้างอาเรย์ 2 มิติ และวนลูปแล้ว print ให้ครบทุกจำนวน (หน้าที่แล้ว)
2. ทดลองตั้งค่าในอาเรย์ 2 มิติ เอาสมาชิกตัวสุดท้ายของแต่ละแถวจากการวนลูป มาบวกกัน และแสดงผลจากการบวกกันบน Console



Software Park Thailand  
</Code Camp>

# Agenda

1.วิธีการติดตั้งโปรแกรม **Eclipse**

2.Basic Syntax

3.Operator

4.Control Flow

5.Method

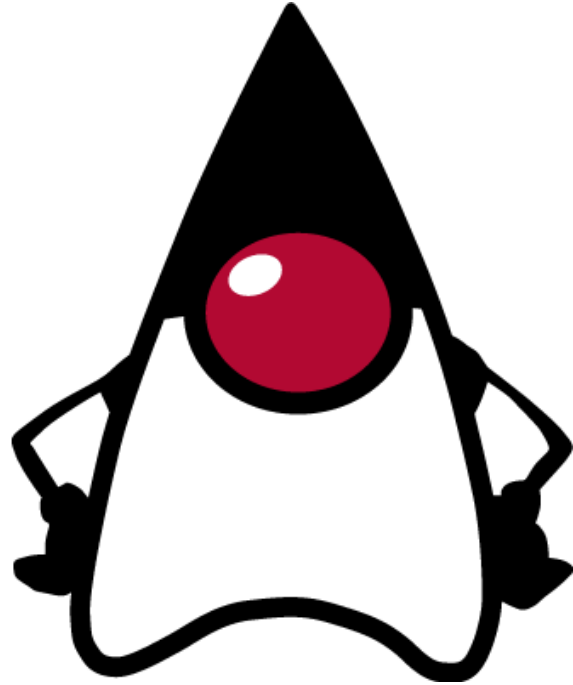
6.Strings

7.Arrays

8.Maven

9.Homework

# Homework



# ข้อตกลง

1. Build เป็นไฟล์ .jar มาส่งเท่านั้น ไม่รับ source code
2. Jar ไฟล์ต้องทำงานได้ ด้วยคำสั่ง `java -jar ชื่อไฟล์.jar`

หากติดขัด หรือสงสัยสิ่งใด หรืออยากให้อธิบายเพิ่มเติม







Software Park Thailand  
</Code Camp>

# HW: 1 - Loop 8 ข้อ

- จงเขียน method `draw(int n)` ให้รับค่าตัวเลขมา และทดลองเรียก
  - `draw(2);`
  - `draw(3);`
  - `draw(4);`
- เพื่อให้วาดรูปแสดงผลออกมาตามตัวอย่างในหน้าถัดๆ ไป ( $n$  มีค่ามากกว่า 4 ก็ได้)
- เพิ่ม `if else` ได้
- เพิ่มตัวแปรได้ ประกาศเพิ่มที่บรรทัดไหนก็ได้
- Loop ทั้ง 8 ข้อให้สร้าง Method `draw1`, `draw2`, `draw3` ตามที่โจทย์กำหนดลงในไฟล์เดียวกัน และเซฟชื่อไฟล์ `Homework8.java`

# 1.1

- จงเขียน method draw1(int n) ให้ print ออกมาในกรณีที่มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**\*\***

n = 2

**\*\*\***

n = 3

**\*\*\*\***

n = 4



## 1.2

- จงเขียน method `draw2(int n)` ให้ print ออกมาในกรณีที่มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**\*\***

**\*\***

`n = 2`

**\*\*\***

**\*\*\***

**\*\*\***

`n = 3`

**\*\*\*\*\***

**\*\*\*\*\***

**\*\*\*\*\***

**\*\*\*\*\***

`n = 4`



## 1.3

- จงเขียน method `draw3(int n)` ให้ print ออกมาในกรณีที่ `n` มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**12**

**12**

`n = 2`

**123**

**123**

**123**

`n = 3`

**1234**

**1234**

**1234**

**1234**

`n = 4`



## 1.4

- จงเขียน method `draw4(int n)` ให้ print ออกมาในกรณีที่ `n` มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**21**

**21**

`n = 2`

**321**

**321**

**321**

`n = 3`

**4321**

**4321**

**4321**

**4321**

`n = 4`



# 1.5

- จงเขียน method draw5(int n) ให้ print ออกมาในกรณีที่มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**11**  
**22**

n = 2

**111**  
**222**  
**333**

n = 3

**1111**  
**2222**  
**3333**  
**4444**

n = 4



## 1.6

- จงเขียน method `draw6(int n)` ให้ print ออกมาในกรณีที `n` มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**22**  
**11**

`n = 2`

**333**  
**222**  
**111**

`n = 3`

**4444**  
**3333**  
**2222**  
**1111**

`n = 4`



## 1.7

- จงเขียน method draw7(int n) ให้ print ออกมาในกรณีที่มีค่าต่างๆ ได้ผลลัพธ์ดังนี้
- ผลลัพธ์จริงไม่ต้องมีเว้นวรรคเลยก็ได้หาก n เป็น 4

**12**  
**34**

n = 2

**123**  
**456**  
**789**

n = 3

**1 2 3 4**  
**5 6 7**  
**8**  
**9 10 11 12**  
**13 14 15 16**

n = 4





## 1.8

- จงเขียน method `draw8(int n)` ให้ print ออกมาในกรณีที `n` มีค่าต่างๆ ได้ผลลัพธ์ดังนี้
- ผลลัพธ์จริงไม่ต้องมีเว้นวรรคเลยก็ได้หาก `n` เป็น 4

**43**  
**21**

`n = 2`

**987**  
**654**  
**321**

`n = 3`

**16 15 14 13**  
**12 11 10 9**  
**8 7 6 5**  
**4 3 2 1**

`n = 4`

# HW: 2 - Multiplication Table

จงเขียน method multiplyTable คำนวณค่าทุกค่าที่คูณ 2 แล้ว จาก Array 2 มิติที่กำหนดให้ และแสดงผลผ่าน Console ดังตัวอย่างทางขวามือ  
ใช้ Integer.parseInt("1") เพื่อแปลงจาก "1" (String) เป็น 1 (int)

```
String[][] table = {  
    { "1", "2", "3" },  
    { "4", "5", "6" },  
    { "7", "8", "9" }  
};  
  
multiplyTable(table);
```

## Output

```
2, 4, 6  
8, 10, 12  
14, 16, 18
```



Software Park Thailand  
</Code Camp>

## HW: 3 - Loop 9 ข้อ

- จงเขียน method `draw(int n)` ให้รับค่าตัวเลขมา และทดลองเรียก
  - `draw(2);`
  - `draw(3);`
  - `draw(4);`
- เพื่อให้วาดรูปแสดงผลออกมาตามตัวอย่างในหน้าถัดๆ ไป ( $n$  มีค่ามากกว่า 4 ก็ได้)
- เพิ่ม if else ได้
- เพิ่มตัวแปรได้ ประกาศเพิ่มที่บรรทัดไหนก็ได้



## 3.1

- จงเขียน method `draw9(int n)` ให้ print ออกมาในกรณีที `n` มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**0**  
**2**

`n = 2`

**0**  
**2**  
**4**

`n = 3`

**0**  
**2**  
**4**  
**6**

`n = 4`



## 3.2

- จงเขียน method `draw10(int n)` ให้ print ออกมาในกรณีที `n` มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**2**  
**4**

`n = 2`

**2**  
**4**  
**6**

`n = 3`

**2**  
**4**  
**6**  
**8**

`n = 4`



## 3.3

- จงเขียน method draw11(int n) ให้ print ออกมาในกรณีที่มีค่าต่างๆ ได้ผลลัพธ์ดังนี้
- ผลลัพธ์จริงไม่ต้องมีเว้นวรรคเลยก็ได้หาก n เป็น 4

**12**  
**24**

n = 2

**123**  
**246**  
**369**

n = 3

**1 2 3 4**  
**2 4 6 8**  
**3 6 9 12**  
**4 8 12 16**

n = 4

## 3.4

- จงเขียน method draw12(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

```
_*  
*_  
n = 2
```

```
_**  
*_*  
**_  
n = 3
```

```
_***  
**_*  
***_  
****_  
n = 4
```



## 3.5

- จงเขียน method draw13(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

\* \_

\_ \*

n = 2

\*\* \_

\* \_ \*

\_ \*\*\*

n = 3

\*\*\* \_

\*\* \_ \*

\* \_ \*\*\*

\_ \*\*\*\*

n = 4





## 3.6

- จงเขียน method draw14(int n) ให้ print ออกมาในกรณีที่มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

\* \_

\*\*

n = 2

\* \_ \_

\*\* \_

\*\*\*

n = 3

\* \_ \_ \_

\*\* \_ \_

\*\*\* \_

\*\*\*\*

n = 4



## 3.7

- จงเขียน method draw15(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**\*\***

**\*\_**

n = 2

**\*\*\***

**\*\*\_**

**\*\_\_**

n = 3

**\*\*\*\***

**\*\*\*\_**

**\*\*\_\_**

**\*\_\_\_**

n = 4



## 3.8

- จงเขียน method `draw16(int n)` ให้ print ออกมาในกรณีที `n` มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

\* \_

\*\*

\* \_

`n = 2`

\* \_ \_

\*\* \_

\*\*\*

\*\* \_

\* \_ \_

`n = 3`

\* \_ \_ \_

\*\* \_ \_

\*\*\* \_

\*\*\*\*

\*\*\* \_

\*\* \_ \_

\* \_ \_ \_

`n = 4`



## 3.9

- จงเขียน method draw17(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**1-**  
**22**  
**1-**

n = 2

**1--**  
**22-**  
**333**  
**22-**  
**1--**

n = 3

**1---**  
**22--**  
**333-**  
**4444**  
**333-**  
**22--**  
**1---**

n = 4



จงแสดงผลสูตรคูณตั้งแต่ 1 ถึง 12 ตามตัวเลขที่  
กรอกผ่าน Keyboard และแสดงผลผ่าน Console

**ห้ามลูปเกิน 1 ลูป**

[illegible]



Software Park Thailand  
</Code Camp>

# HW: 4 - Keyboard!

เพิ่ม config ที่ launch.json เพื่อให้กรอกรหัส  
ข้อมูลได้

```
{  
  "type": "java",  
  "name": "CodeLens (Launch) - App",  
  "request": "launch",  
  "mainClass": "jar.App",  
  "projectName": "my-maven-project",  
  "console": "externalTerminal",  
  "stopOnEntry": false  
}
```

Code ดัง  
นี้

```
//import java lib. to use keyboard  
import java.util.Scanner;  
  
public class App {  
  public static void main(String[] args) {  
    //Prepare to receive input from keyboard  
    Scanner scan = new Scanner(System.in);  
    System.out.println("Please enter number");  
  
    //Get value from entered values  
    int enterNumber = scan.nextInt();  
    System.out.println(enterNumber);  
    scan.close();  
  }  
}
```

## (Optional) HW 5 Loop 8 ข้อ

- จงเขียน method `draw(int n)` ให้รับค่าตัวเลขมา และทดลองเรียก
  - `draw(2);`
  - `draw(3);`
  - `draw(4);`
- เพื่อให้วาดรูปแสดงผลออกมาตามตัวอย่างในหน้าถัดๆ ไป ( $n$  มีค่ามากกว่า 4 ก็ได้)
- เพิ่ม if else ได้
- เพิ่มตัวแปรได้ ประกาศเพิ่มที่บรรทัดไหนก็ได้



# 5.1

- จงเขียน method draw18(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

```
-*  
**
```

n = 2

```
--*  
-**  
***
```

n = 3

```
---*  
--**  
-***  
****
```

n = 4





## 5.2

- จงเขียน method draw19(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**\*\***

**\_\***

n = 2

**\*\*\***

**\_\*\***

**\_\_\***

n = 3

**\*\*\*\***

**\_\*\*\***

**\_\_\*\***

**\_\_\_\***

n = 4



## 5.3

- จงเขียน method draw20(int n) ให้ print ออกมาในกรณีที่มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**\_\***

**\*\***

**\_\***

n = 2

**\_\_\***

**\_\*\***

**\*\*\***

**\_\*\***

**\_\_\***

n = 3

**\_\_\_\***

**\_\_\_\*\***

**\_\*\*\***

**\*\*\*\***

**\_\*\*\***

**\_\_\*\***

**\_\_\_\***

n = 4



## 5.4

- จงเขียน method draw21(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**-1**  
**23**  
**-4**

n = 2

**--1**  
**-23**  
**456**  
**-78**  
**--9**

n = 3

**---1**  
**--23**  
**-456**  
**78910**  
**-111213**  
**--1415**  
**---16**

n = 4



## 5.5

- จงเขียน method draw22(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

```
 _*_  
***
```

n = 2

```
  _*_  
 _***_  
*****
```

n = 3

```
   _*_  
  _***_  
 _*****_  
*****
```

n = 4



## 5.6

- จงเขียน method draw23(int n) ให้ print ออกมาในกรณีที่มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

\*\*\*

\_\*\_

n = 2

\*\*\*\*\*

\_\*\*\*\_

\_\_\*\_\_

n = 3

\*\*\*\*\*

\_\*\*\*\*\*\_

\_\_\*\*\*\_\_

\_\_\_\*\_\_\_

n = 4



## 5.7

- จงเขียน method draw24(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

```
_*_
```

```
***
```

```
_*_
```

n = 2

```
__*__
```

```
_***_*
```

```
*****
```

```
_***_*
```

```
__*__
```

n = 3

```
___*___
```

```
__***__
```

```
_*****_
```

```
*****
```

```
_*****_
```

```
__***__
```

```
___*___
```

n = 4



## 5.8

- จงเขียน method draw25(int n) ให้ print ออกมาในกรณีที่ n มีค่าต่างๆ ได้ผลลัพธ์ดังนี้

**-1-  
234  
-5-**

n = 2

**--1--  
-234-  
56789  
-10112-  
--13--**

n = 3

**---1---  
--234--  
-56789-  
1011213141516  
-1718192021-  
--222324--  
---25---**

n = 4