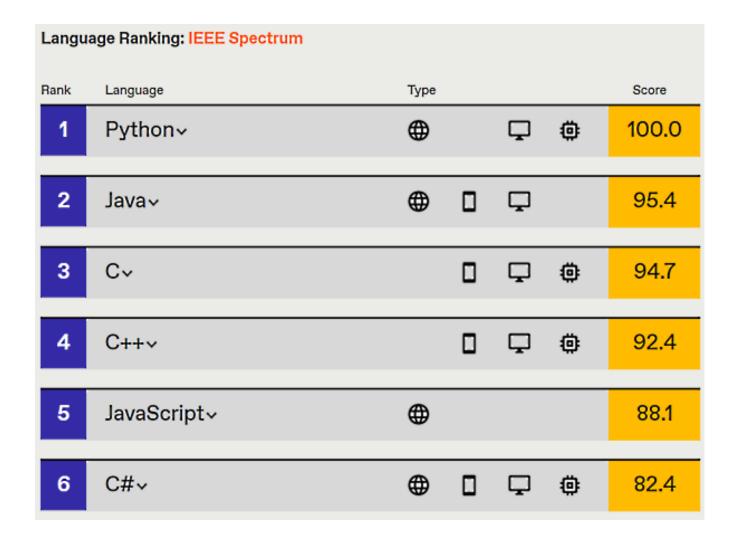
Java Programming

SC363204

Java Web Application Development การพัฒนาโปรแกรมประยุกต์บนเว็บด้วยภาษาจาวา

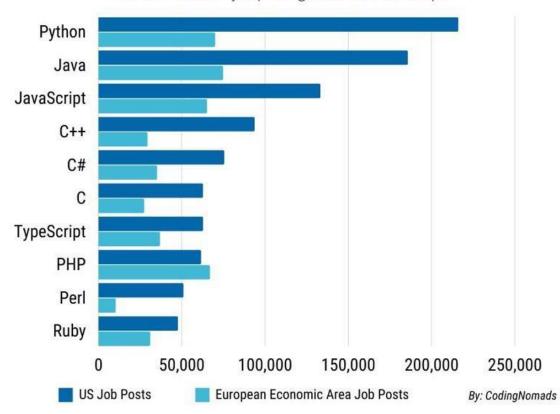


Top Programming Language 2022

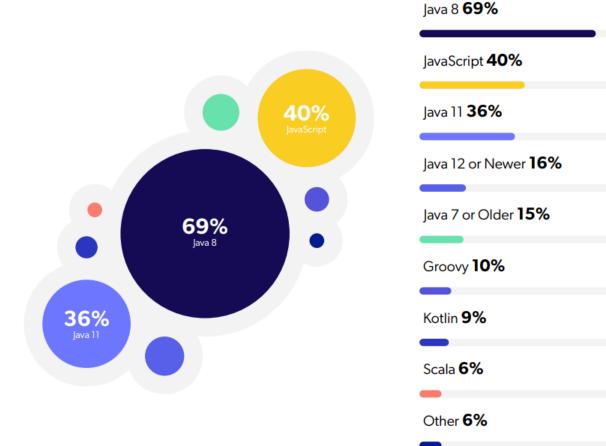


Most in-demand programming languages of 2022

Based on LinkedIn job postings in the USA & Europe



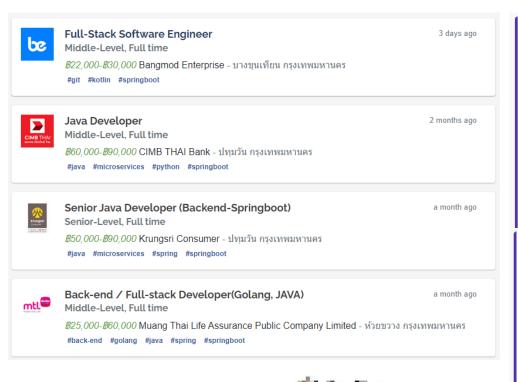
What Java programming language are you using in your main application?



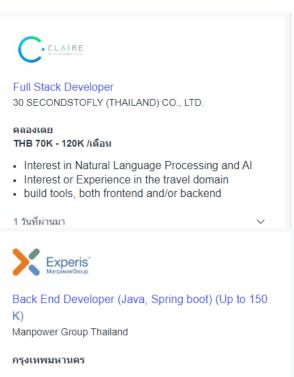
www.jrebel.com

JRebel by Perforce © Perforce Software, Inc. All trademarks and registered trademarks are the property of their respective owners. (0120JB21)

Java Developer Jobs







· Experience in JAVA - Spring boot

Near by BTS JatuchakPostgreSQL, MySQL,



↑ AZURE

Backend Developer

Azure Computers (Thailand) Co., Ltd.

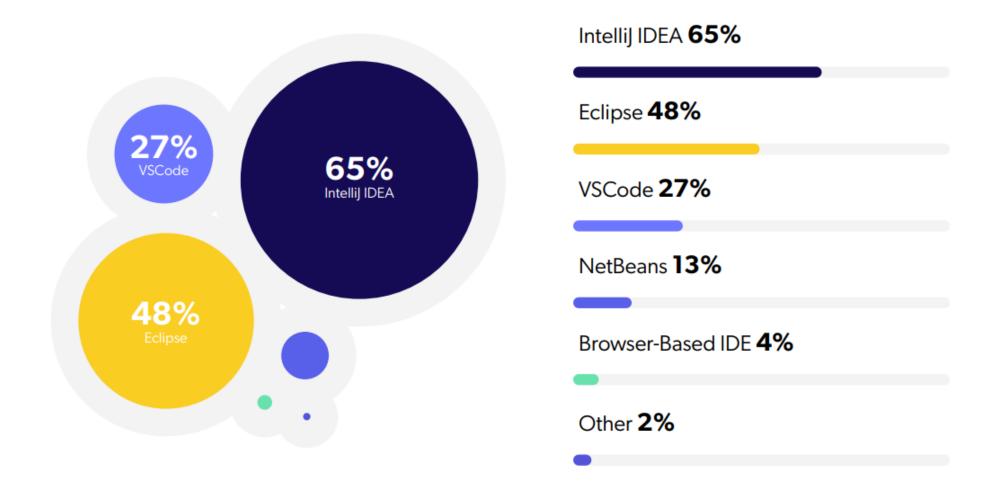
กรุงเทพมหานคร THB 55K - 90K /เดือน

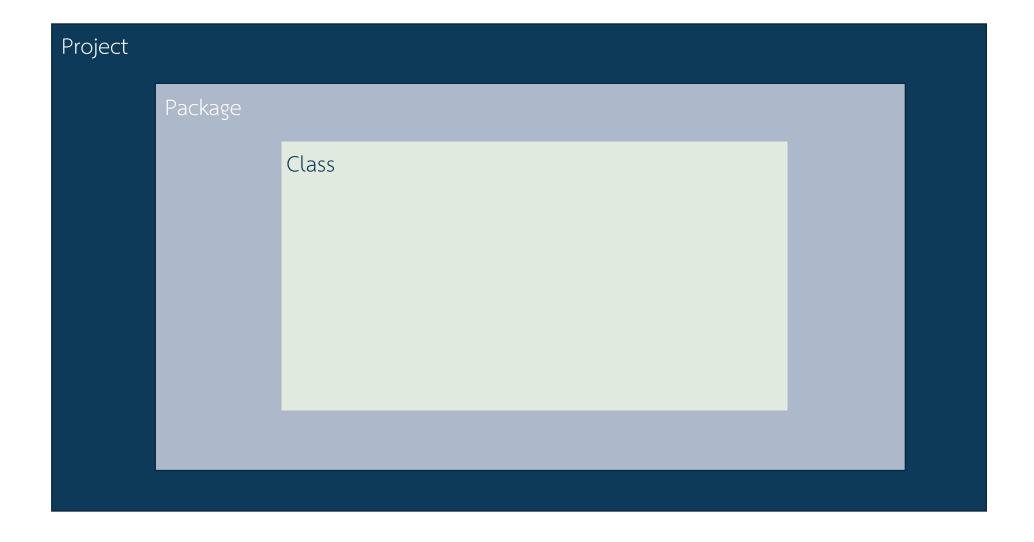
- · Backend with C#.Net
- · Backend with Java Spring boot
- Thai National

Java สร้างอะไรได้บ้าง?

- Android applications
- Back-end Systems
- Web applications
- Desktop Applications
- Cloud computing
- Robotics

What developer IDE do you use professionally?





- Package: เป็นกลุ่มของ class หรือ library มาตรฐานของภาษา Java ที่มีฟังก์ชันต่างๆ
- Class: ในส่วนของการประกาศคลาส **จะต้องประกาศคลาสให้ชื่อตรงกับไฟล์เสมอ** โดยชื่อ คลาสนั้น**ควรจะขึ้นต้นด้วยตัวใหญ่** และถ้ามีหลายคำให้ใช้ตัวพิมพ์ใหญ่แบ่ง
- Method: หลังจากคลาสสร้างแล้ว จะเป็นประกาศเมธอดภายในคลาส โดยในการที่จะรัน โปรแกรมได้**จะต้องมีเมธอดที่ชื่อว่า main**
- Statements: เป็นคำสั่งของโปรแกรมเพื่อให้โปรแกรมทำงานตามต้องการ

- คำสั่งต่าง ๆ จะต้องเขียนภายในคลาสเท่านั้น
- ภายในคลาส มีได้ทั้งตัวแปร และเมธอด
- คลาสที่รันเป็น Application ได้ต้อง
 - เป็นคลาสชนิด public
 - มีเมธอด main เป็นสมาชิก
 - ตั้งชื่อไฟล์ชื่อเดียวกับคลาส

Naming conventions

- ใช้รูปแบบ Camel case สำหรับคำที่มีตั้งแต่ 2 คำขึ้นไป (Compound Words)
- ชื่อคลาส คำนาม, คำแรกขึ้นต้นด้วย<mark>ตัวพิมพ์ใหญ่</mark> Circle EuropeCar GraduateStudent
- ชื่อ Attribute/ตัวแปร คำนาม, คำแรกขึ้นต้นด้วยตัวพิมพ์เล็ก firstName yearOfBirth studentName
- ชื่อ Method เป็นคำกริยา คำแรกขึ้นต้นด้วยตัวพิมพ์เล็ก getName() showDetails()
- ชื่อ Package <mark>ตัวพิมพ์เล็ก</mark> อาจนำชื่อโดเมนขององค์กรมาเขียนกลับ com.pantip.controller com.pantip.model

Package

- Package เปรียบเสมือนโฟลเดอร์ในการจัดหมวดหมู่ไฟล์โค้ดโปรแกรม
- ภาษาจาวาแนะนำให้เก็บไฟล์คลาสต่างๆไว้ใน package เสมอ

Output:

Welcome to Java Web

ยินดีต้อนรับ

ตัวแปร

- ภาษาจาวา<mark>ต้องระบุชนิดตัวแปร</mark> และประกาศตัวแปรก่อนใช้ทุกครั้ง
- ตัวแปรพื้นฐาน (Primitive Types)
 - 。 ชนิดตรรกะ (logical) <mark>Boolean</mark>
 - ชนิดอักขระ (textual) char, String
 - งนิดตัวเลขจำนวนเต็ม (integer) byte, short, int, long
 - ชนิดตัวเลขทศนิยม (floating point) float, double
- ชื่อทุกอย่างเป็น case-sensitive
- ไม่ใช้ reserve word มาตั้งชื่อ

If - else

```
int x = 20;
int y = 18;
if (x > y) {
    System.out.println("x is greater than y");
}
else {
    System.out.println("y is greater than x");
}
```

Output:

x is greater than y

Loop

```
กำหนดค่าตัวเริ่มต้นให้ตัวแปรควบคุม
while (เงื่อนไขการเข้าลูป) {
ชุดคำสั่งทำซ้ำ
ปรับปรุงค่าตัวแปรควบคุม
}
```

```
for (กำหนดค่าตัวเริ่มต้นให้ตัวแปรควบคุม; เงื่อนไขการเข้าลูป; ปรับปรุงค่าตัวแปรควบคุม;) {
ชุดคำสั่งทำซ้ำ
}
```

Loop

```
กำหนดค่าตัวเริ่มต้นให้ตัวแปรควบคุม
while (เงื่อนไขการเข้าลูป) {
ชุดคำสั่งทำซ้ำ
ปรับปรุงค่าตัวแปรควบคุม
}
```

```
int i = 0;
while (i < 5) {
System.out.println(i);
i++;
}</pre>
```

Loop

```
for (กำหนดค่าตัวเริ่มต้นให้ตัวแปรควบคุม; เงื่อนไขการเข้าลูป; ปรับปรุงค่าตัวแปรควบคุม;) {
ชุดคำสั่งทำซ้ำ
}
```

```
for (int i = 0; i < 5; i++;) {
    System.out.println(i);
}</pre>
```

Array

• ใช้เครื่องหมาย [] เพื่อประกาศตัวแปรให้อาร์เรย์

```
int score[] = new int [5]; หรือ
int[] score = new int[5];
```

• ประกาศพร้อมกำหนดค่าเริ่มต้น

```
Char vowels[] = {'A', 'E', 'I', 'O', 'U'};
```

Array

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars[0]);
```

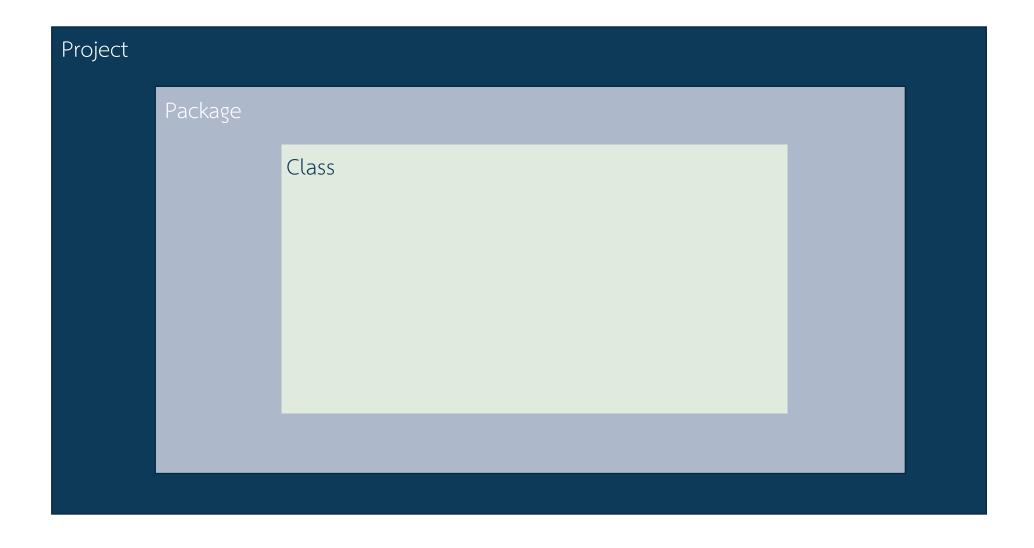
Output:

Volvo

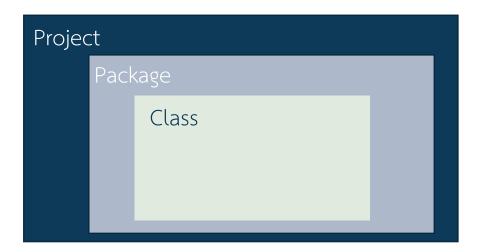
- หลักการเขียนโปรแกรมเชิงวัตถุ คือ มองทุกสิ่งทุกอย่างในโปรแกรมให้เป็นวัตถุ (object)
- ตัวอย่างระบบบัญชีเงินเดือน (Payroll System)







- Package: เป็นกลุ่มของ class หรือ library มาตรฐานของภาษา Java ที่มีฟังก์ชันต่างๆ
- Class: ในส่วนของการประกาศคลาส **จะต้องประกาศคลาสให้ชื่อตรงกับไฟล์เสมอ** โดยชื่อคลาสนั้น ควรจะขึ้นต้นด้วยตัวใหญ่ และถ้ามีหลายคำให้ใช้ตัวพิมพ์ใหญ่แบ่ง
- Method: หลังจากคลาสสร้างแล้ว จะเป็นประกาศเมธอดภายในคลาส โดยในการที่จะรันโปรแกรมได้ จะต้องมีเมธอดที่ชื่อว่า main
- Statements: เป็นคำสั่งของโปรแกรมเพื่อให้โปรแกรมทำงานตามต้องการ



Employee.java

```
package com.payroll;
public class Employee {
 public String name;
    public String department;
    public int salary;
    public int ot;
// method คำนวณเงินเดือน
 public int calculateSalary() {
        return salary + ot;
// method แสดงรายละเอียด
 public String display() {
        return "ชื่อ:" + name +
          , แผนก:" + department +
        ", เงินเดือน" + salary +
        ", OT" + ot +
        ", Total" + calculateSalary();
```

<u>Object</u>



ชื่อ:	อนันต์
แผนก:	พนักงานทั่วไป
เงินเดือน:	25000
OT:	2500
ຽງງາ	25000 + 2500 - 27500

0 0001.	25000 1 2500 - 21500
ผลลัพธ์:	ชื่อ:อนันต์ แผนก: พนักงานทั่วไป เงินเดือน: 25000 OT: 2500 รวม: 27500



เงนเดอน: 30000 OT: 0	ชื่อ: แผนก: เงินเดือน: OT:	นิมมาน พนักงานทั่วไป 30000 0
-------------------------	-------------------------------------	---------------------------------------

รวม:	30000 + 0 = 30000
ผลลัพธ์:	ชื่อ:นิมมาน แผนก: พนักงานทั่วไป เงินเดือน: 30000 OT: 0 รวม: 30000



ชื่อ:	จอมขวัญ
แผนก:	พนักงานทั่วไป
เงินเดือน:	45000
OT:	0
01.	45000 45000

ผลลัพธ์: ชื่อ:จอมชวัญ แผนก: พนักงานทั่วไป เงินเดือน: 45000 OT: 0 ร	າມ: 45000

• รูปแบบการสร้าง Object

```
ชื่อคลาส ชื่อ object = new ชื่อคลาส();
Ex. Employee Obj1 = new Employee();
```

```
Employee.java

public class Employee {
//ประกาศตัวแปร

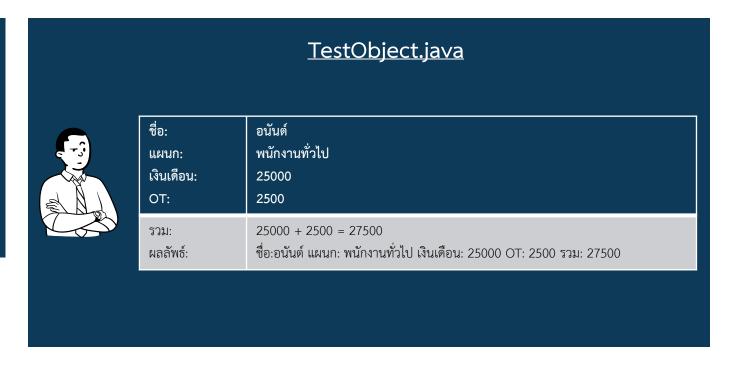
public String name;

public String department;

public int salary;

public int ot;

// ... }
```



```
    รูปแบบการสร้าง Object
    ชื่อคลาส ชื่อ object = new ชื่อคลาส();
    Ex. Employee somsak = new Employee();
```

```
Employee.java

public class Employee {
//ประกาศตัวแปร

public String name;

public String department;

public int salary;

public int ot;

// ... }
```

TestObject.java



```
    ชื่อ:
    แผนก:
    พนักงานทั่วไป
    เงินเดือน:
    25000
    OT:
    2500
    รวม:
    25000 + 2500 = 27500
    ผลลัพธ์:
    ชื่อ: สมศักดิ์ แผนก: พนักงานทั่วไป เงินเดือน: 25000 OT: 2500 รวม: 27500
```

```
package com.payroll;

public class TestObj {

public static void main(String[] args) {

    Employee obj1 = new Employee();
    obj1.name = "ตมศักดิ์";

obj1.department = "พนักงานทั่วไป";

obj1.salary = 25000;
    obj1.ot = 2500;
    System.out.println(obj1.display());

}
```

Employee.java

```
package com.payroll;
public class Employee {
 public String name;
    public String department;
    public int salary;
    public int ot;
// method คำนวณเงินเดือน
 public int calculateSalary() {
        return salary + ot;
// method แสดงรายละเอียด
 public String display() {
        return "ชื่อ:" + name +
          , แผนก:" + department +
        ", เงินเดือน" + salary +
        ", OT" + ot +
        ", Total" + calculateSalary();
```

TestObject.java



ผลลัพธ์:

ผลลัพธ์:

ชื่อ:	อนันต์
แผนก:	พนักงานทั่วไป
เงินเดือน:	25000
OT:	2500
รวม:	25000 + 2500 = 27500

ชื่อ:อนันต์ แผนก: พนักงานทั่วไป เงินเดือน: 25000 OT: 2500 รวม: 27500



ชื่อ:	นิมมาน
แผนก:	พนักงานทั่วไป
เงินเดือน:	30000
OT:	0
รวม:	30000 + 0 = 30000



ชื่อ:	จอมขวัญ
แผนก:	พนักงานทั่วไป
เงินเดือน:	45000
OT:	0
รวม:	45000 + 0 = 45000
ผลลัพธ์:	ชื่อ:จอมขวัญ แผนก: พนักงานทั่วไป เงินเดือน: 45000 OT: 0 รวม: 45000

ชื่อ:นิมมาน แผนก: พนักงานทั่วไป เงินเดือน: 30000 OT: 0 รวม: 30000

- การทำงานบางเมธอดในคลาส <mark>จำเป็น</mark>ต้องกำหนดค่าเริ่มต้นให้กับบาง attribute ก่อนเรียกใช้ เมธอด
- ช่วยในการสร้าง object และการกำหนดค่าให้กับตัวแปรใน object ภายในบรรทัดเดียว

```
Employee.java

public class Employee {
//เพิ่มคำสั่ง
   public Employee(){
   }
// ... }
```

- หากไม่เขียน constructor ใดๆไว้ default constructor จะถูกสร้างโดยอัตโนมัติ
- แต่ถ้าหากเขียน constructor แบบใดก็ตามไว้ default constructor จะไม่ถูกสร้างอัตโนมัติ นักพัฒนาจะต้องเขียน default constructor เอง

TestConstructor.java



```
      ชื่อ:
      อนันต์

      แผนก:
      พนักงานทั่วไป

      เงินเดือน:
      25000

      OT:
      2500

      รวม:
      25000 + 2500 = 27500

      ผลลัพธ์:
      ชื่อ:อนันต์ แผนก: พนักงานทั่วไป เงินเดือน: 25000 OT: 2500 รวม: 27500
```

```
package com.payroll;

public class TestConstructor {
    public static void main(String[] args) {

    Employee obj1 = new Employee "อนันต์", "พนักงานทั่วไป",25000,2500);

    System.out.println(obj1.display());

}
```

<u>Object</u>



ชื่อ:	อนันต์
แผนก:	พนักงานทั่วไป
เงินเดือน:	25000
OT·	2500

รวม:	25000 + 2500 = 27500
ผลลัพธ์:	ชื่อ:อนันต์ แผนก: พนักงานทั่วไป เงินเดือน: 25000 OT: 2500 รวม: 27500



ชื่อ:	นิมมาน
แผนก:	พนักงานทั่วไป
เงินเดือน:	30000
OT:	0

ลัพธ์:	ชื่อ:นิมมาน แผนก: พนักงานทั่วไป เงินเดือน: 30000 OT: 0 รวม: 30000
--------	---



ชื่อ:	จอมขวัญ
แผนก:	พนักงานทั่วไป
เงินเดือน:	45000
OT:	0
รวม:	45000 + 0 = 45000
ผลลัพธ์:	ชื่อ:จอมขวัญ แผนก: พนักงานทั่วไป เงินเดือน: 45000 OT: 0 รวม: 45000

ผลลัพธ์:

<u>Object</u>



ชื่อ:	อนันต์
แผนก:	พนักงานทั่วไป
เงินเดือน:	25000
OT:	2500
รวม:	25000 + 2500 = 27500

ชื่อ:อนันต์ แผนก: พนักงานทั่วไป เงินเดือน: 25000 OT: 2500 รวม: 27500



ชื่อ:	นิมมาน
แผนก:	พนักงานทั่วไป
เงินเดือน:	30000
OT:	ถ
รวม:	30000 + 0 = 30000
ผลลัพธ์:	ชื่อ:นิมมาน แผนก: พนักงานทั่วไป เงินเดือน: 30000 OT: 0 รวม: 30000



ชื่อ:	จอมขวัญ
แผนก:	พนักงานทั่วไป
เงินเดือน:	45000
OT:	0
รวม:	45000 + 0 = 45000
ผลลัพธ์:	ชื่อ:จอมขวัญ แผนก: พนักงานทั่วไป เงินเดือน: 45000 OT: 0 รวม: 45000

ถ้าหากไม่ต้องการกำหนดค่าให้กับ ตัวแปร OT หรือไม่รับค่า ให้มีค่า เริ่ม ต้ น เ ป็ น O จ ะ เ ขี ย น Constructor อย่างไร

การใช้ Constructor และไม่ใช้ Constructor

```
package com.payroll;
public class TestObj {

public static void main(String[] args) {

    Employee obj1 = new Employee();
    obj1.name = "อนันต์";
    obj1.department = "พนักงานทั่วไป";
    obj1.salary = 25000;
    obj1.ot = 2500;
    System.out.println(obj1.display());
}
}
```

```
package com.payroll;
public class TestConstructor {
    public static void main(String[] args) {
          Employee obj1 = new Employee "อนันต์", "พนักงานทั่วไป",25000,2500);
          System.out.println(obj1.display());
```