

## ปฏิบัติการที่ 4

### คำแนะนำทั่วไป

1. ให้นักศึกษา ทำปฏิบัติการภายใต้ directory c:\javallab (ต้องสร้างขึ้นเอง)
2. หลังเสร็จสิ้นปฏิบัติการ หากต้องการเก็บ files ไว้ให้ copy ไปยังที่เก็บสำรองอื่น (thumbdrive, mail server, ฯลฯ)

### จุดมุ่งหมายของปฏิบัติการ

1. เพื่อเรียนรู้และใช้งาน IDE Eclipse สำหรับการพัฒนาโปรแกรม
2. เพื่อให้สามารถเขียนโปรแกรมที่ใช้คำสั่งทางเลือกได้
3. เพื่อให้สามารถเขียนโปรแกรมที่ใช้คำสั่งการทำซ้ำได้

การส่ง ให้นำเฉพาะ java files ทั้งหมด zip (อยู่ใน src directory ของโปรเจกต์ที่สร้างขึ้น) แล้ว submit บน courseweb  
 หมายเหตุ โค้ดที่ต้องเขียนในปฏิบัติการ เริ่มต้นที่หน้า 12

### ปฏิบัติการ

#### I. เพื่อเรียนรู้และใช้งาน IDE Eclipse สำหรับการพัฒนาโปรแกรม

เครื่องมือที่ใช้ในการพัฒนาโปรแกรมจาวา ทั่วไปประกอบด้วย Text Editor, Compiler และ JVM สำหรับการรัน ในปฏิบัติการก่อน ๆ เราใช้เครื่องมือพื้นฐานที่พบได้จากระบบปฏิบัติการทั่วไปพร้อมกับเครื่องมือของ Java Development Kit (JDK) ในปฏิบัตินี้เราจะใช้เครื่องมือประเภท IDE (Integrated Development Environment) ที่รวม Text Editor, Compiler, Debugger, เครื่องมือสำหรับรัน รวมถึงตัวช่วยการพัฒนาอื่นไว้เป็นโปรแกรมเดียว IDE ช่วยให้การพัฒนา การตรวจสอบ การทดสอบโค้ดเป็นไปอย่างมีประสิทธิภาพ IDE สำหรับจาวามีหลายตัว เช่น Eclipse, Netbeans, JBuilder เป็นต้น ในที่นี้เราเลือกใช้ Eclipse ที่เป็นเครื่องมือซึ่งไม่มีค่าใช้จ่าย และได้รับความนิยมกว้างขวาง

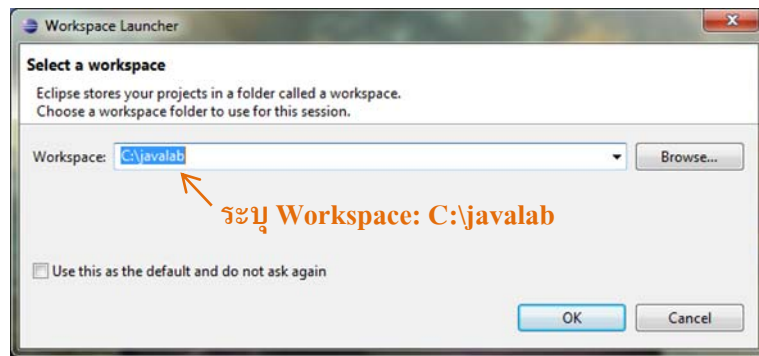
#### การติดตั้ง Eclipse

สามารถ download Eclipse ได้จาก <http://www.eclipse.org/downloads/> โดยเลือกเวอร์ชันที่เหมาะสมกับการใช้งาน ในที่นี้ใช้ Eclipse IDE for Java Developers (128 MB)

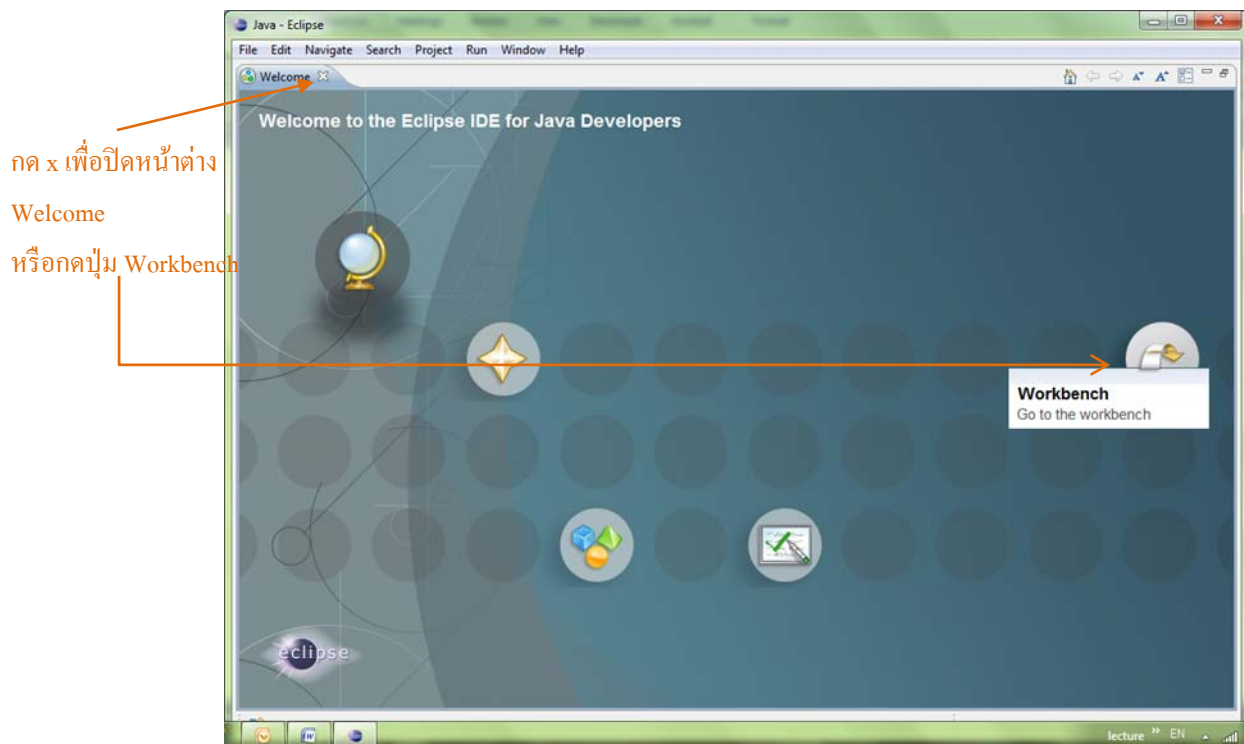
การติดตั้ง Eclipse ไม่มีกระบวนการพิเศษใด ๆ เพียงนำไฟล์ที่ download ได้ (zip file) ทำ unzip ไว้ใน directory ที่เหมาะสม เช่น C:\eclipse จากนั้นก็สามารถเริ่มต้นใช้งานได้

#### เริ่มต้นการใช้ Eclipse

- ให้กด Double-click บนไฟล์ eclipse.exe ที่อยู่ใน directory ที่ได้ unzip ไว้ Eclipse จะเริ่มต้นทำงาน หน้าต่างแรกที่ปรากฏ เป็นหน้าต่างเพื่อให้ระบุที่ตั้งของ workspace (รูปที่ 1 ในที่นี้จะใช้ C:\javallab) เพื่อใช้เป็น directory สำหรับเก็บโปรเจกต์จาวา จากนั้น click OK จะเห็นหน้าต่าง Welcome page (รูปที่ 2) เพื่อเริ่มการเขียนโค้ด ให้ปิดหน้าต่าง Welcome นั้น โดยกดปุ่ม X หรือกดปุ่ม Workbench



รูปที่ 1 หน้าต่างเริ่มต้นเพื่อกำหนด Workspace directory



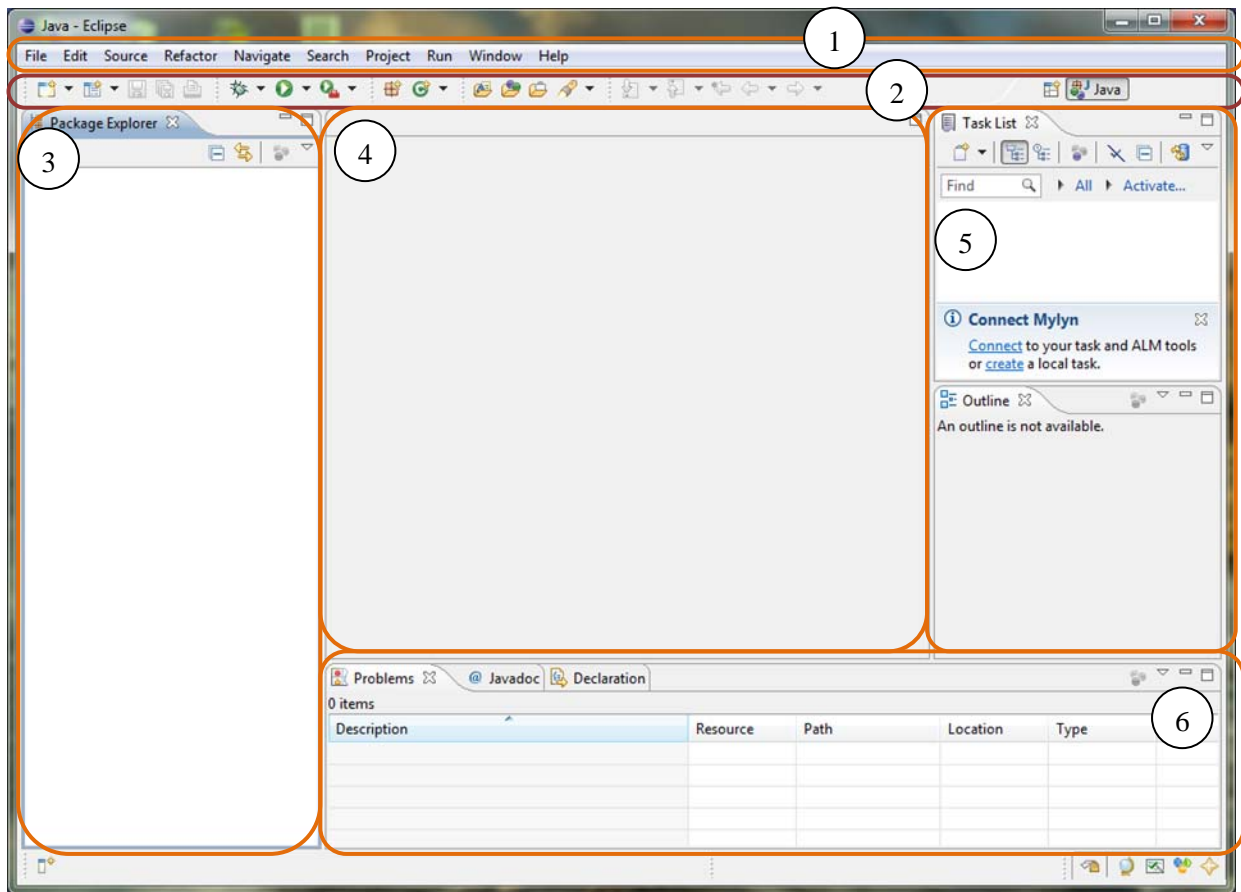
รูปที่ 2 หน้าต่าง Welcome ของ Eclipse

หมายเหตุ ถ้าน.ศ. ไม่เห็นหน้าต่างเริ่มต้นเพื่อกำหนด Workspace directory หมายความว่าได้มีการกำหนดค่าปริยาย (default) ของ Workspace Directory เอาไว้แล้ว ใ้น.ศ. ทำการเปลี่ยนค่า Workspace directory โดยเลือกเมนู File → Switch Workspace → Other...

หน้าต่างของ Eclipse แบ่งพื้นที่ได้เป็นส่วนต่าง ๆ 6 ส่วน (แต่ละส่วนใช้หมายเลขอ้างอิง ดังในรูปที่ 3) คือ

1. Menu เป็นเมนูสำหรับคำสั่งการทำงานตามประเภท เช่น File, Edit, Source ฯลฯ
2. Shortcut icon เป็นปุ่มทางลัดสำหรับคำสั่งการทำงานที่ใช้อยู่ เช่น New, การรันไฟล์, การสร้างคลาสใหม่
3. Package Explorer เป็นส่วนที่แสดงลำดับชั้นของ directory และไฟล์ภายในโปรเจกต์ ทั่วไปใช้เพื่อการท่อง (navigate) ไปยังหรือเปิดไฟล์เพื่อดูและแก้ไข
4. Editor เป็นพื้นที่สำหรับเขียนโค้ด โดยสามารถเปิดไฟล์เพื่อการดูหรือเขียนได้มากกว่า 1 ไฟล์พร้อมกัน

5. Task List และ Outline แสดงส่วนของงาน (Task) และ โครงสร้างของคลาส (attributes, เมทอด)
6. หน้าต่างข้อมูล ส่วนนี้จะมีหน้าต่างย่อยเช่น
  - a. หน้าต่าง Problems แสดงข้อความปัญหา ที่พบในโค้ด
  - b. หน้าต่าง Javadoc ให้ข้อมูลเกี่ยวกับ API ของโค้ดส่วนที่ถูกเลือก (selected) ใน Editor
  - c. หน้าต่าง Declaration ให้รายละเอียดโค้ด (source codes) ในส่วนของโค้ดส่วนที่ถูกเลือกใน Editor
  - d. หน้าต่าง Console ซึ่งจะปรากฏกรณีเมื่อสั่งรันโค้ด ใช้เพื่อการแสดงผลหรือรับ input จากผู้ใช้



รูปที่ 3 หน้าต่างของ Workbench โดยอาจแบ่งเป็นพื้นที่ส่วนย่อย 6 ส่วน

### เริ่มต้นโปรเจก

การสร้างโปรแกรมจาวาใน Eclipse มีขั้นตอนดังนี้

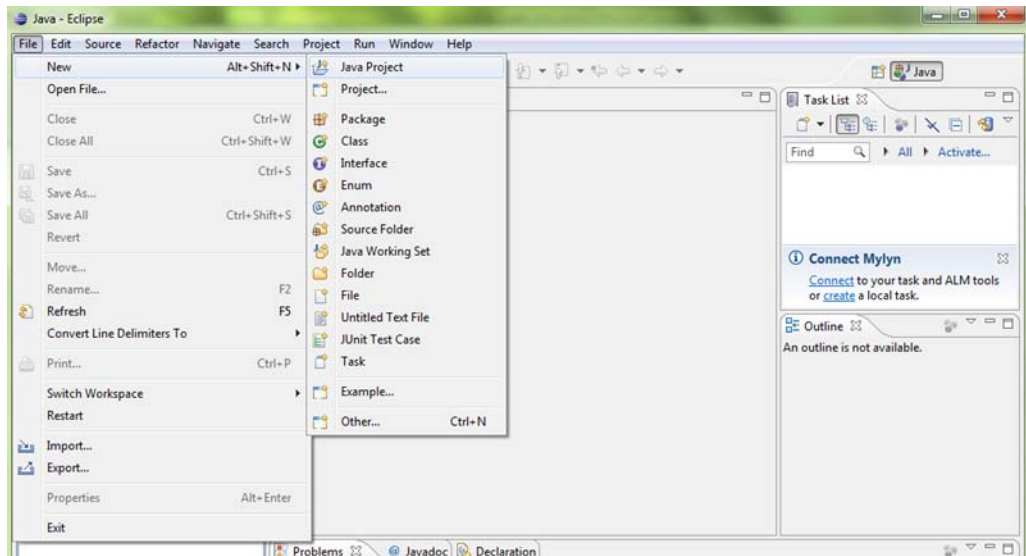
- 1 สร้างโปรเจกสำหรับโปรแกรม เพื่อใช้เก็บกลุ่มของคลาสที่ใช้ในโปรแกรมทั้งหมด
- 2 สร้างคลาสต่าง ๆ ที่ใช้ในโปรแกรม
- 3 ดิบักและทดสอบการทำงานของคลาสที่สร้างขึ้น

### การสร้างโปรเจก

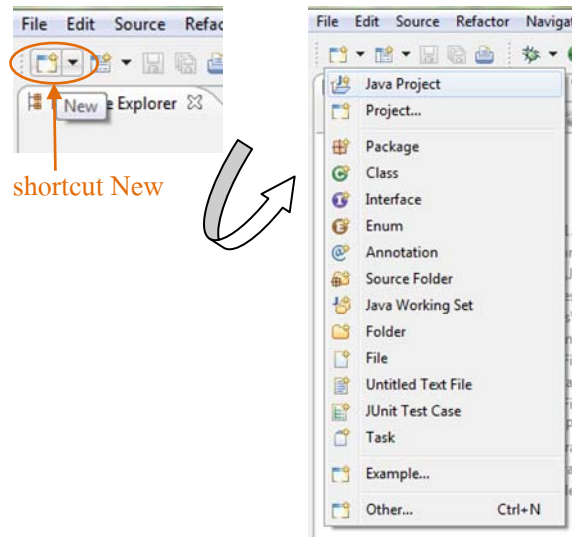
เมื่อเริ่มต้นจะต้องสร้างโปรเจก (Java Project) โดย

- เลือกเมนู File → New → Java Project (รูปที่ 4) หรือ

- กด shortcut icon New แล้วเลือก Java Project (รูปที่ 5)

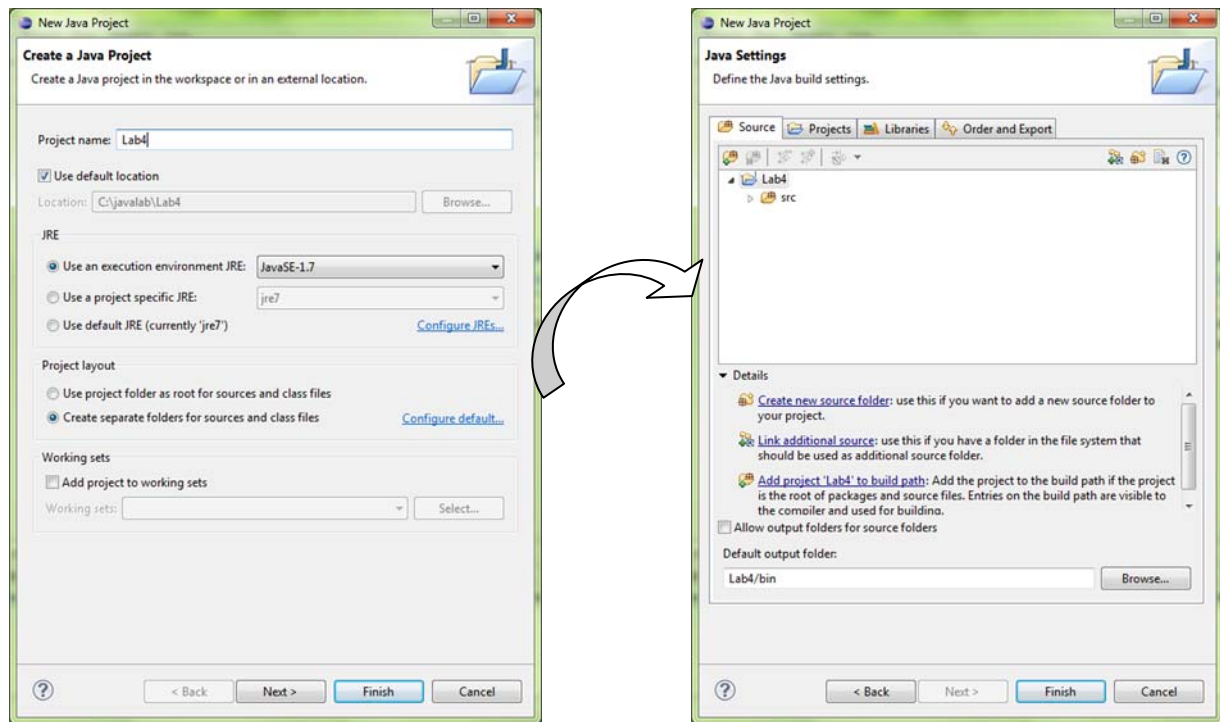


รูปที่ 4 เลือกเมนู File → New → Java Project สำหรับสร้างโปรเจกใหม่



รูปที่ 5 ใช้ short cut icon New → Java Project สำหรับสร้างโปรเจกใหม่

ในรูปที่ 6 เมื่อใส่ชื่อ Project name เช่น **Lab4** คลิกปุ่ม Next> จากนั้นคลิกปุ่ม Finish (เราใช้คำปรีขายสำหรับการ setup โปรเจก ซึ่งเหมาะกับการเขียนโค้ดเบื้องต้นทั่วไป)



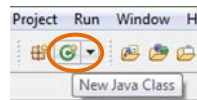
รูปที่ 6 หน้าต่างของขั้นตอนการสร้างโปรเจก

จะเห็นโปรเจก Lab4 ปรากฏในส่วน Package Explorer แต่ยังไม่มีคลาสใด ๆ ขึ้นต่อไปจึงต้องสร้างคลาสในโปรเจก

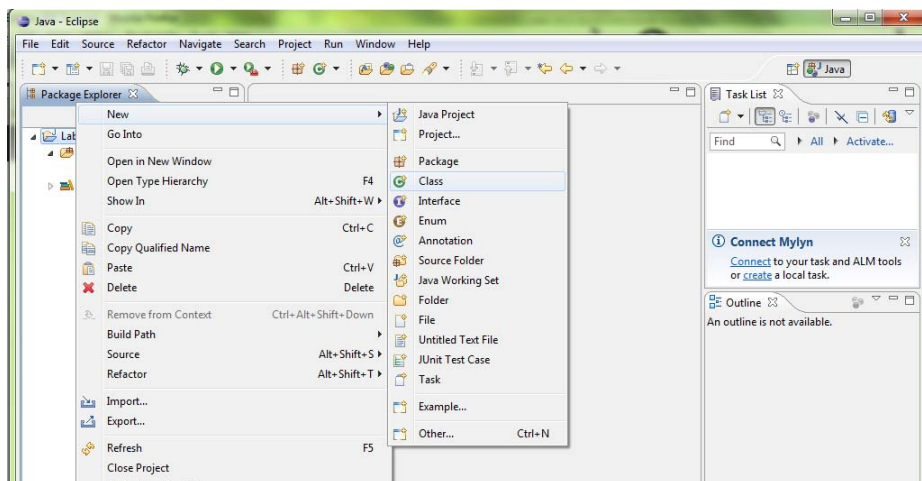
### การสร้างคลาสภายในโปรเจก

เราสามารถสร้าง Java Class ภายในโปรเจกโดย

- คลิกเมาส์ขวาที่ Lab4 เลือก New → Class (รูปที่ 7) หรือ
- กด shortcut New เลือก Class หรือ

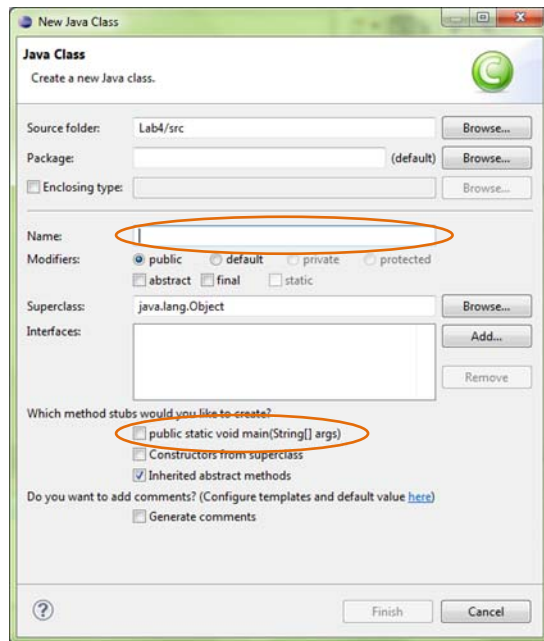


- จะใช้ shortcut New Java Class เลขก็ได้

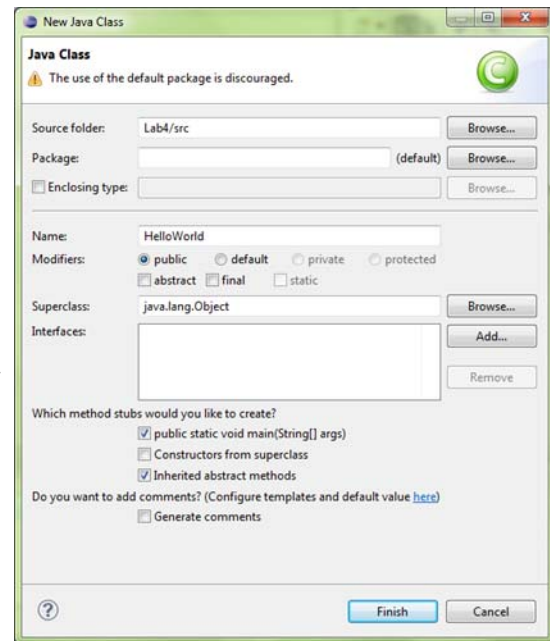


รูปที่ 7 หน้าต่างขั้นตอนการสร้างคลาสใหม่

จากนั้นหน้าต่าง New Java Class ปรากฏเพื่อให้ระบุรายละเอียด (รูปที่ 8.a) ให้ใส่ชื่อของคลาสในช่อง Name (ในที่นี้จะสร้าง HelloWorld ดังรูปที่ 8.b) เนื่องจากคลาสนี้จะเป็นคลาสทดสอบหรือคลาสในการรัน เราระบุให้สร้าง main method ด้วยโดยเลือก ☒ public static void main(String[] args)

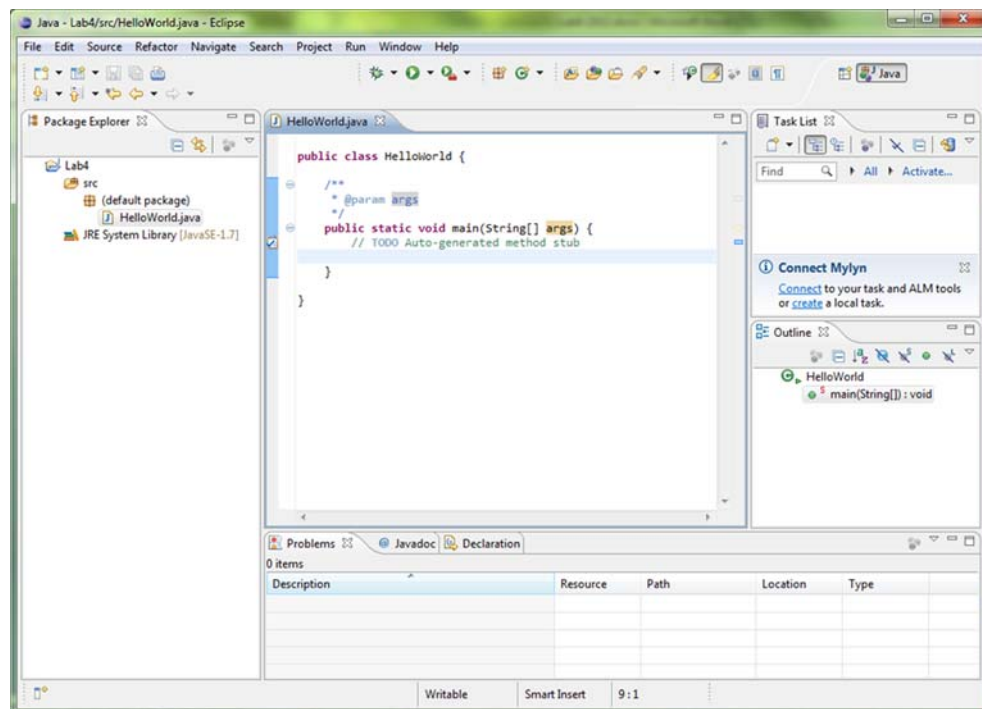


a) หน้าต่าง New Java Class เมื่อเริ่มต้น



b) หลังใส่ชื่อคลาสและเลือก main method ด้วย

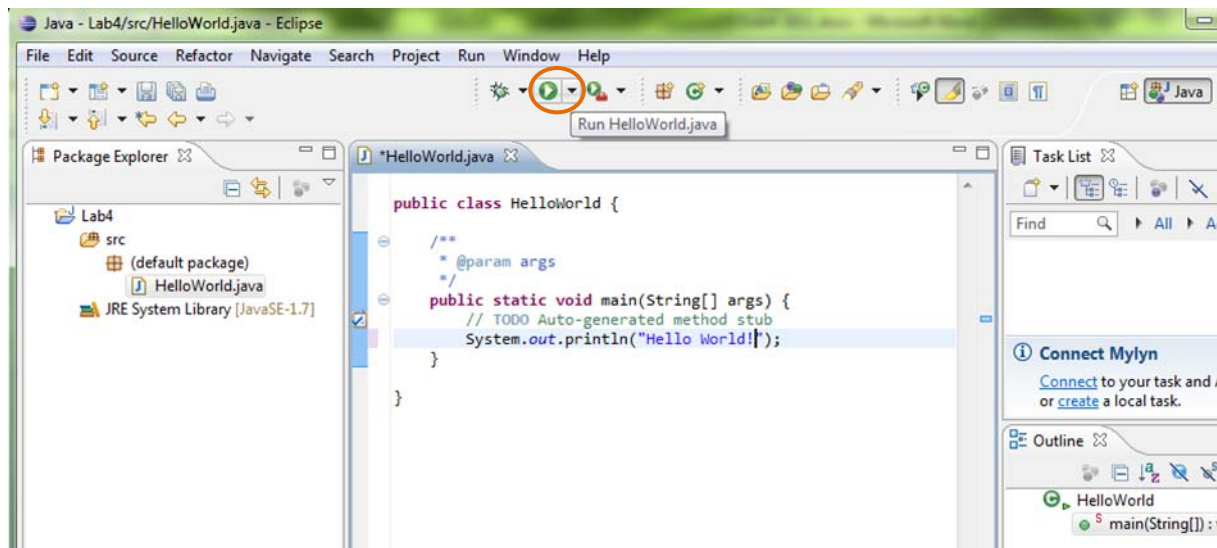
รูปที่ 8 หน้าต่างสำหรับระบุชื่อคลาส พร้อมรายละเอียดอื่นที่เกี่ยวข้อง จากนั้นคลิกปุ่ม Finish จะได้โค้ดจาวาของ HelloWorld ดังรูปที่ 9



รูปที่ 9 โค้ดของคลาส HelloWorld ที่ได้จากการสร้างคลาสใหม่




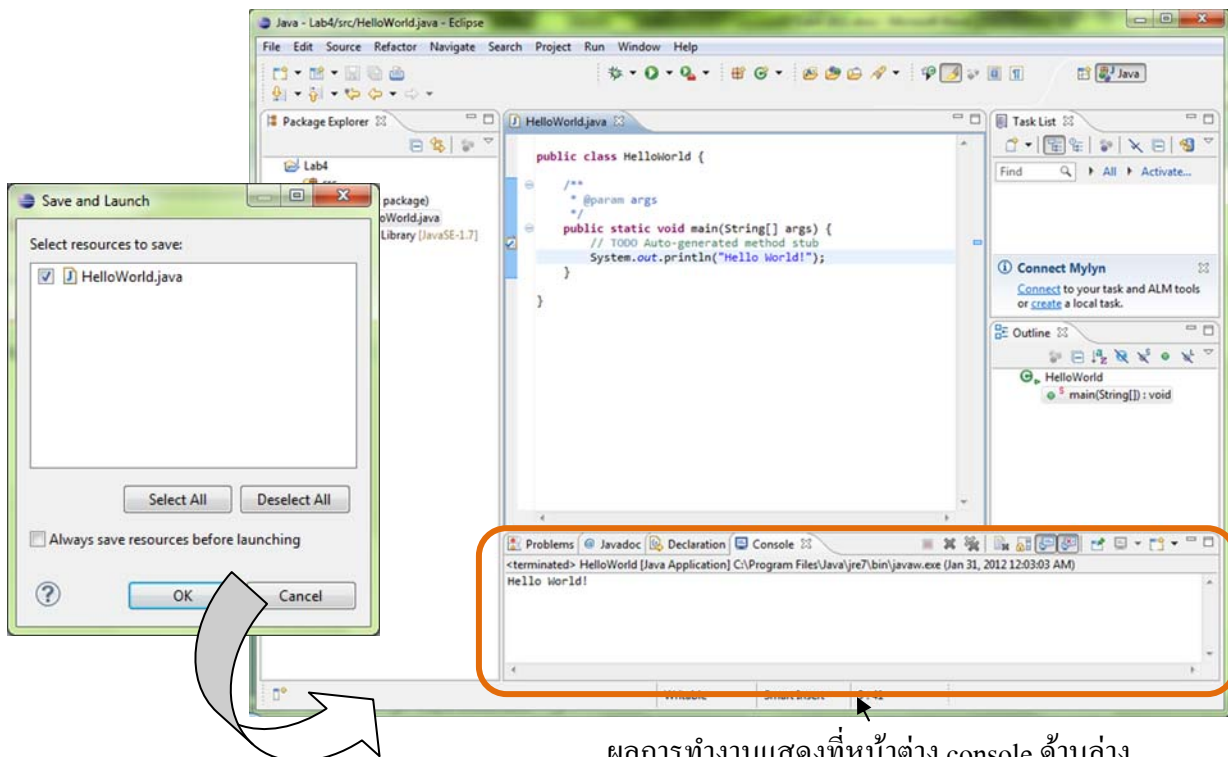
จากนั้นเพิ่มโค้ดเพื่อให้พิมพ์ข้อความ “Hello World!” ออกหน้าจอ ดังในรูปที่ 10



รูปที่ 10 หลังเพิ่มโค้ดใน HelloWorld

### การรันเพื่อทดสอบการทำงาน

การรันเพื่อทดสอบการทำงานของคลาส สามารถรันคลาสนั้นโดยคลิก  (shortcut icon ที่วงไว้ในรูปที่ 10) เนื่องจากยังไม่ได้บันทึกไฟล์ Eclipse จะขึ้นหน้าต่างถามให้บันทึกก่อนเริ่มรัน ให้คลิกปุ่ม OK เมื่อทำงานจะได้ข้อความ “Hello World” ที่หน้าต่าง Console (รูปที่ 11)



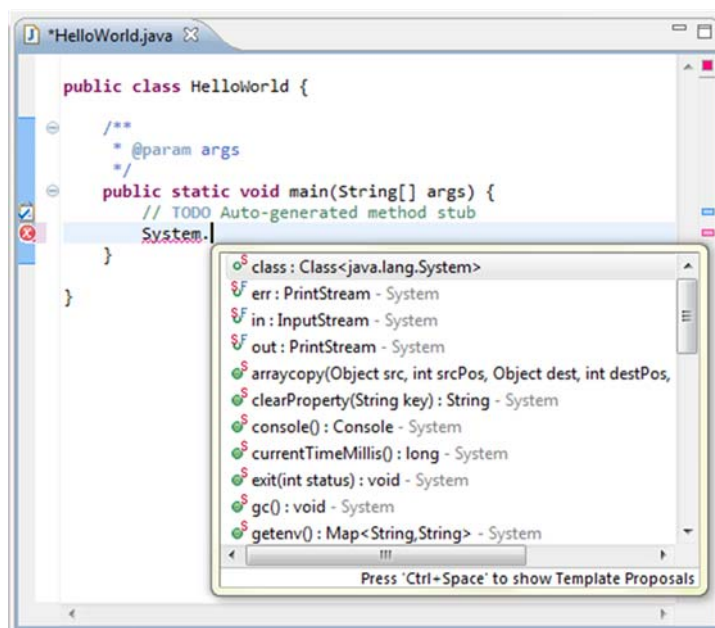
ผลการทำงานแสดงที่หน้าต่าง console ด้านล่าง

รูปที่ 11 แสดงการรันและข้อความผลลัพธ์ปรากฏที่หน้าต่าง console

## การใช้งาน Editor อย่างง่าย

### Code Completion



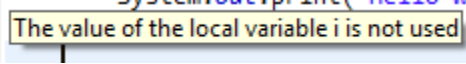

Editor ที่ใช้นอกจากจะช่วยให้เขียนโค้ดได้เช่นเดียวกับ Text Editor อื่นแล้ว ยังมีตัวช่วยในการเติมโค้ด โดยที่เมื่อเรากีย์โค้ดบางส่วน หาก Editor พบว่าเป็นวัตถุหรือคลาสที่รู้จัก (เช่นมีอยู่ใน API หรือเป็นคลาสในโปรเจกนั้น) มันจะแสดงตัวช่วยเช่น field หรือเมทอดที่มีให้ใช้งานได้ (รูปที่ 12) ในกรณีที่เครื่องช้า ตัวช่วยอาจขึ้นไม่ทันกับการคีย์โค้ด อาจสั่งให้แสดง list นี้ได้โดยกด CTRL-space



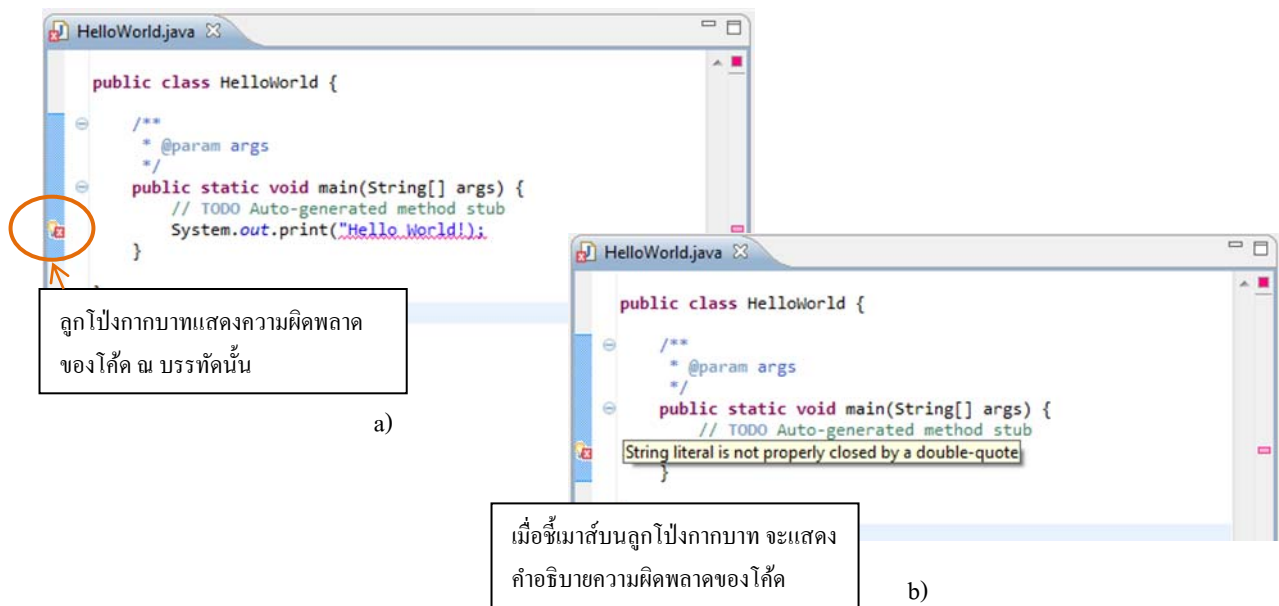
รูปที่ 12 list ของ field หรือเมทอดที่แสดงโดยตัวช่วย Code Completion

### Code Correction



ในขณะที่เขียนโค้ด Editor จะช่วยตรวจสอบความถูกต้องของ Syntax ที่เขียนไปด้วย หากพบมีโค้ดผิดพลาดจะแสดงสัญลักษณ์บนบรรทัดที่มีความผิดพลาด (ตัวอย่างดังรูปที่ 13) โดยสัญลักษณ์ที่ปรากฏแบ่งได้เป็น 3 แบบ คือ

- ลูกโป่งสีเหลืองกับเครื่องหมาย ! บนบรรทัด  แสดงการเตือน เช่นการประกาศตัวแปรโดยไม่ได้ใช้งาน และเมื่อเอาเมาส์ชี้บนลูกโป่ง จะแสดงข้อความบอกสาเหตุการเตือน  
- ลูกโป่งสีเหลืองพร้อมกากบาทแดงบนบรรทัด  แสดงโค้ดผิดพลาด (รูปที่ 13.a) และเมื่อเอาเมาส์ชี้ไปบนลูกโป่งจะแสดงข้อความบอกสาเหตุความผิดพลาด (ดังตัวอย่างรูปที่ 13.b) ในกรณีตัวอย่างเกิดจากลืมใส่ " ปิดค่า String literal

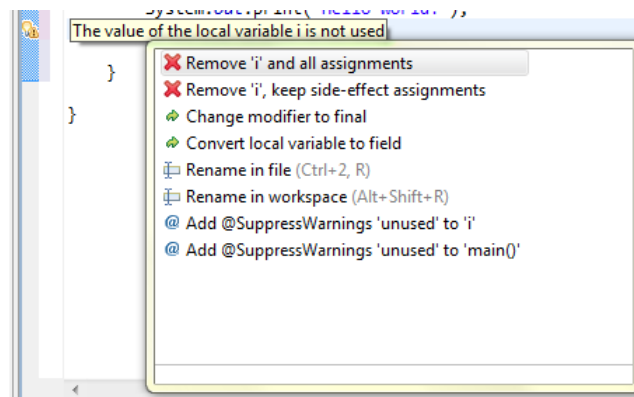




รูปที่ 13 ตัวอย่างความผิดพลาดและสัญลักษณ์ที่ปรากฏ

- c. กากบาทสีแดงบนบรรทัด  แสดงว่ามี syntax error และเมื่อเอาเมาส์ชี้บนกากบาทจะมีคำอธิบาย เช่น กรณีลืมใส่ปีกกาปิดคลาส  Syntax error, insert "]" to complete ClassBody


ในกรณีที่โค้ดผิดพลาด (ยกเว้นกรณีของกากบาทสีแดง) หากคลิกเมาส์บนสัญลักษณ์เหล่านี้ Eclipse จะแสดงทางเลือกที่เป็นได้ ดังตัวอย่างในรูปที่ 13



รูปที่ 14 ตัวอย่าง List ของทางเลือก

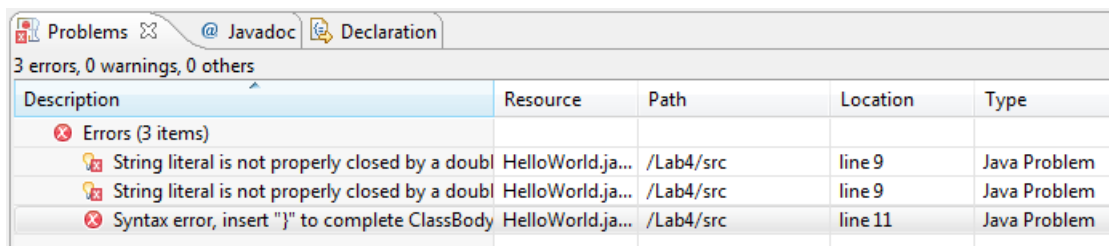
กรณีที่ต้องการเลือกทางเลือกอันใดอันหนึ่งกด double-click บนทางเลือกนั้น Eclipse จะดำเนินการแก้ไขโค้ดที่ผิดพลาดนั้นให้ (ตามวิธีที่เลือก)

## การบันทึกโค้ด การเปิด-ปิด โค้ดและที่ตั้งของโค้ด

- โค้ดที่เขียนควรต้องบันทึกไว้เป็นระยะ โดยกดปุ่ม icon  โดยรูปดิस्कหนึ่งอันเป็นการบันทึกเฉพาะไฟล์ ปัจจุบัน ในขณะที่รูปดิस्कหลายอันเป็นการบันทึกไฟล์ที่มีการเปิดอยู่ใน Editor ทั้งหมด
- สำหรับการเปิดโค้ดหรือเปิดโปรเจกต์ที่ปิดไปแล้วสามารถเปิดโดย double-click บนไฟล์ของ List ในหน้าต่างส่วน Package Explorer
- การปิดโค้ด ปิดได้โดยคลิก x บนขอบหน้าต่างข้างชื่อไฟล์ และการปิดโปรเจกต์ ทำได้โดยคลิกเมาส์ขวาแล้วเลือก Close Project
- ที่ตั้งของโค้ดทั้งหมดจะอยู่ภายใต้ directory ที่ระบุใน workspace โดย source codes ทั้งหมดจะอยู่ภายใต้ directory ของโปรเจกต์ใน directory ย่อย src ตัวอย่างเช่น ถ้า workspace อยู่ที่ C:\javallab และ โปรเจกต์ชื่อ Lab4 โค้ดทั้งหมดจะอยู่ที่ directory: **C:\javallab\Lab4\src**

## การใช้หน้าต่างข้อมูล

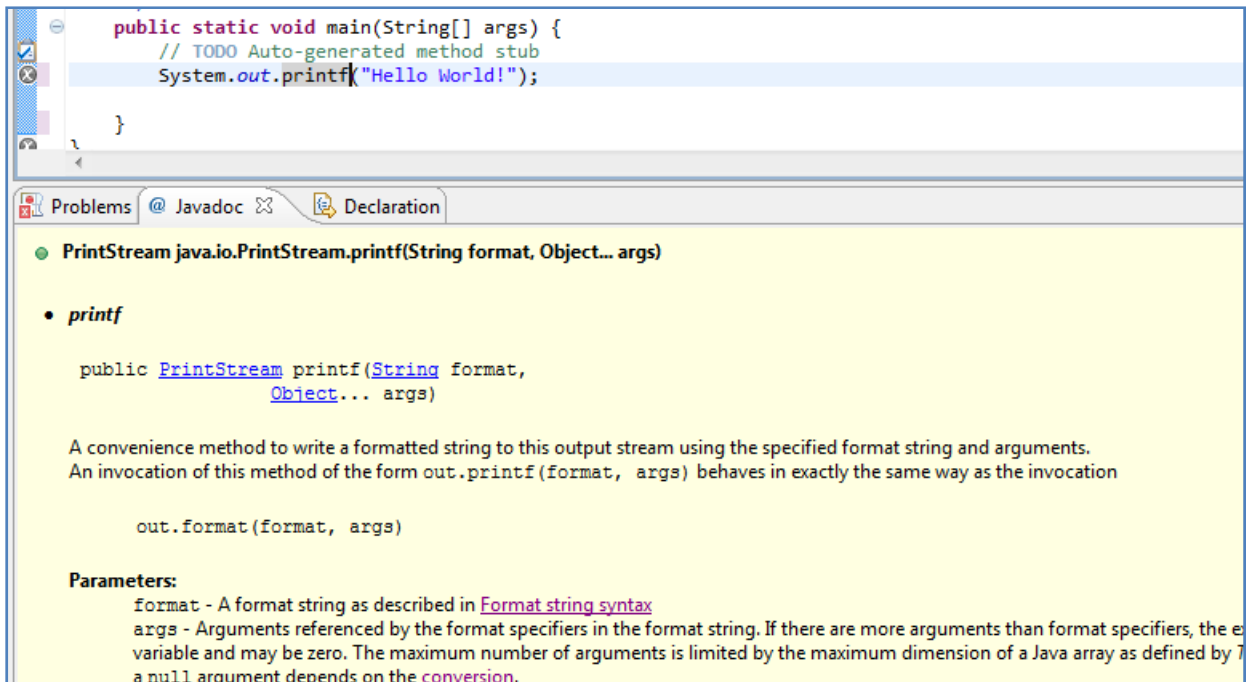
- **หน้าต่าง Problems** เป็นหน้าต่างแสดงข้อความปัญหา (Problems) ที่พบในโค้ด โดยเมื่อกดบันทึกไฟล์ รายการ Error ของโค้ดที่พบจะแสดงในหน้าต่างนี้ (รูปที่ 15) เมื่อกด double-click บนข้อความ error นั้น Eclipse จะกระโดดไปยังบรรทัดความผิดพลาดให้



Description	Resource	Path	Location	Type
<b>Errors (3 items)</b>				
String literal is not properly closed by a double quote	HelloWorld.java	/Lab4/src	line 9	Java Problem
String literal is not properly closed by a double quote	HelloWorld.java	/Lab4/src	line 9	Java Problem
Syntax error, insert ";" to complete ClassBody	HelloWorld.java	/Lab4/src	line 11	Java Problem

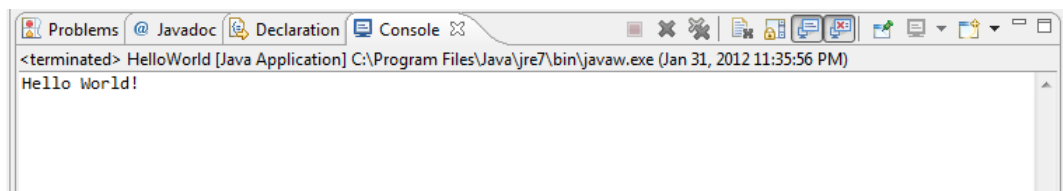
รูปที่ 15 แสดงหน้าต่าง Problem พร้อมรายการความผิดพลาด

- **หน้าต่าง Javadoc** ให้ข้อมูลเกี่ยวกับ API ของโค้ดส่วนที่ถูกเลือก (selected) ใน Editor (รูปที่ 16) โดย mark เลือกโค้ดที่มีคำอธิบาย javadoc เช่นเป็นเมทอดของคลาสใน API หรือมี javadoc ของโค้ดนั้น หากคลิกไปที่หน้าต่าง Javadoc จะปรากฏคำอธิบาย ในรูปที่ 16 mark เลือก printf ของ System.in จะขึ้นคำอธิบายเมทอดนั้น



รูปที่ 16 แสดงหน้าต่าง Javadoc ที่แสดงรายละเอียดของเมทอด printf ที่ถูกเลือกในโค้ด

- หน้าต่าง *Console* จะปรากฏกรณีที่มีโค้ดที่สามารถทำงานได้ และกดให้รัน หน้าต่างนี้ใช้ทั้งเพื่อการแสดงผลและรับ input จากผู้ใช้ (ในลักษณะเดียวกับ cmd.exe ที่เคยใช้ก่อนหน้านี้) รูปที่ 17 เป็นการแสดงผลลัพธ์เมื่อรันโปรแกรม HelloWorld



รูปที่ 17 แสดงหน้าต่าง console

นอกจากพื้นฐานการใช้งานที่กล่าวมาข้างต้นแล้ว Eclipse ยังมีความสามารถอื่น ๆ อีก เราสามารถศึกษาการใช้เพิ่มเติมได้จาก <http://www.eclipse.org/resources/?category=Getting%20Started>