

**2110101 : การทำโปรแกรมคอมพิวเตอร์**

**แนวคิดการทำโปรแกรม**

**ภาควิชาวิศวกรรมคอมพิวเตอร์**

**จุฬาลงกรณ์มหาวิทยาลัย**

# หัวข้อ

---

---

- องค์ประกอบของโปรแกรม
- ขั้นตอนการแปลและสั่งงานโปรแกรม
- การเขียนโปรแกรมให้คนและเครื่องเข้าใจ
- การทำงานแบบลำดับ
- ขั้นตอนการทำโปรแกรม

# โปรแกรมแรก

---

---

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- โปรแกรมประกอบด้วยคำต่างๆ นำมาเขียนให้ถูกต้องตามไวยากรณ์
- คำต่างๆ ถูกค้นด้วยช่องว่าง ตัวอักษรพิเศษ หรือการขึ้นบรรทัดใหม่
- คำต่างๆ มีทั้งคำสงวนของภาษา คำที่ระบบใช้ คำที่ตั้งขึ้นใหม่ และอื่นๆ

# โปรแกรมประกอบด้วย class

---

---

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- โปรแกรมนี้ประกอบด้วย class หนึ่ง class ชื่อ Hello1
- รายละเอียดของ class Hello1 อยู่ภายในวงเล็บ { } หลังข้อความ `public class Hello1`

# Class ประกอบด้วย method

---

---

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- method บรรยายขั้นตอนการทำงาน
- รายละเอียดของ method อยู่ภายในวงเล็บ { } หลัง หัว method

# Method ประกอบด้วยชุดคำสั่ง

---

---

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- แต่ละบรรทัดคำสั่งปิดท้ายด้วย ; (semicolon)
- `System.out.print` แสดงข้อความภายในวงเล็บออกทางจอภาพ
- `println` ทำงานเหมือน `print` แต่ขึ้นบรรทัดใหม่ด้วย

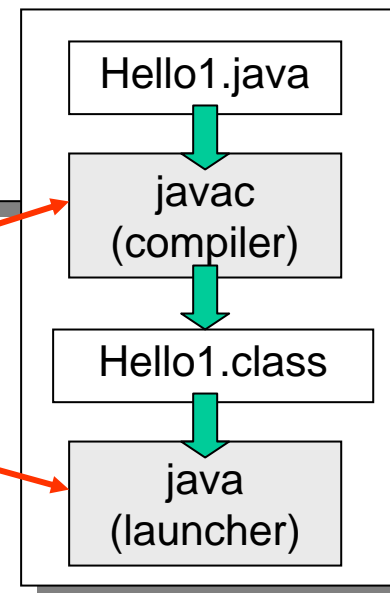
# การแปลและสั่งทำงาน

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

```
JLab>javac Hello1.java
JLab>java Hello1
Hello 2110101
JLab>
```



# main method

---

---

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- java launcher จะเรียกใช้ main method ของ class
- ดังนั้นการทำงานจะเริ่มที่ main method ของ class นั้น



# Comment

---

---

```
// 2110101 : Computer Programming
// First Java Program

public class Hello1 {

    // main method begins Java application execution
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
} // end of class Hello1
```

- Comment เริ่มต้นด้วย // (ยังมีวิธีเขียนแบบอื่น)
- โปรแกรมเมอร์เขียน comment ให้คนอ่าน เครื่องไม่สน
- ตัด comment ทิ้ง โปรแกรมก็ยังทำงานได้เหมือนเดิม

# การเขียนโปรแกรมให้อ่านง่ายๆ

```
public class Hello1{public static void
main(final String[] args){System.out.print("Hello ");
System.out.println                ("2110101");
} }
```

สองโปรแกรมนีทำงานเหมือนกัน  
แต่โปรแกรมไหนน่าอ่านกว่ากัน ?

```
public class Hello1 {
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.println("2110101");
    }
}
```

- หมั่นใช้ช่องว่าง การย่อหน้า และการขึ้นบรรทัดใหม่ เพื่อให้โปรแกรมอ่าน และอ่านง่ายขึ้น

# สวยเกินไปอาจผิดกฎ

---

---

```
*****
* public class Hello1 { *
* * *
*     PUBLIC STATIC VOID main(final String[] args) { *
* * *
*         System.out.print("Hello "); *
* * *
*         System.out.println("2110101"); *
* * *
*     } *
* * *
* } *
*****
```

# การใช้สีเพื่อให้อ่านง่ายขึ้นอีก

---

---

- สีเขียว : comment
- สีน้ำเงิน : คำสงวนของภาษาจาวา
- สีแดง : คำที่พบในคำสั่งมาตรฐานของจาวา
- สีแสง : อักขระที่อยู่ภายในเครื่องหมาย "
- สีม่วง : อักขระที่อยู่ภายในเครื่องหมาย '
- สีดำ : อื่นๆ

```
// 2110101 : Computer Programming
public class Hello1 {
    public static void main(final String[] args) {
        System.out.print("Hello ");
        System.out.print("2110101");
        System.out.print('A');
    }
}
```

# การใช้สีเพื่อสังเกตความผิดพลาด

---

---

```
// 2110101 : Computer Programming
public Class Hello1 {
    public static void main(fina1 String[] args) {
        System.out.print("Hello);
        System.out.print("2110101");
        System.out.print('A');
    }
}
```

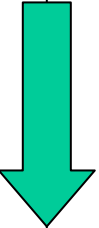
- **Class** ต้องเป็น **class** (ต้องเป็น c ตัวเล็ก)
- **fina1** ต้องเป็น **final** (ต้องเป็นตัวแอลเล็ก ไม่ใช่เลข 1)
- **"Hello);** ต้องเป็น **"Hello");** (ตัว " ปิดท้ายหายไป)
- **System** ต้องเป็น **System** (ต้องเป็น m)

การใช้สีเป็นความสามารถของส่วนป้อนโปรแกรม  
เพื่อคนอ่าน ตัวแปลภาษานั้นไม่สนใจสีเลย

# บรรทัดคำสั่งทำงานจากบนลงล่าง

- โดยทั่วไปโปรแกรมจะทำงานเป็นลำดับ จากบรรทัดคำสั่งบนลงล่าง

```
public class PrintStars {  
    public static void main(final String[] args) {  
        System.out.println( "*****" );  
        System.out.println("      *");  
        System.out.println("      *");  
        System.out.println("      *");  
        System.out.println(*****);  
    }  
}
```



```
JLab>javac PrintStars.java  
JLab>java PrintStars  
*****  
*      *  
*      *  
*      *  
*****  
JLab>
```

# การแสดงผลแบบวินโดว์

---

---

```
import javax.swing.JOptionPane;

public class HelloDialog {
    public static void main(final String[] args) {
        JOptionPane.showMessageDialog(null, "Hello");
        System.exit(0);
    }
}
```



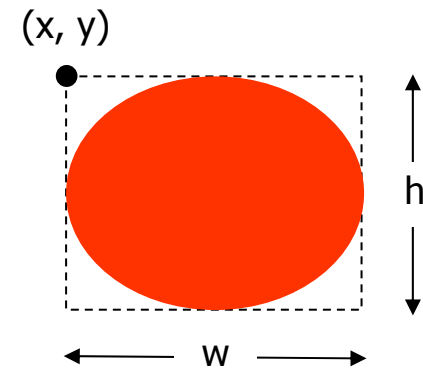
- เราใช้คลัง class มาตรฐานของจาวา `javax.swing.JOptionPane`
- บรรทัด `import` บอกว่าโปรแกรมนี้ต้องการใช้ `JOptionPane`
- ต้องเติม `System.exit(0)` เพื่อให้โปรแกรมหยุดทำงาน

# ตัวอย่างคำสั่งวาดรูป

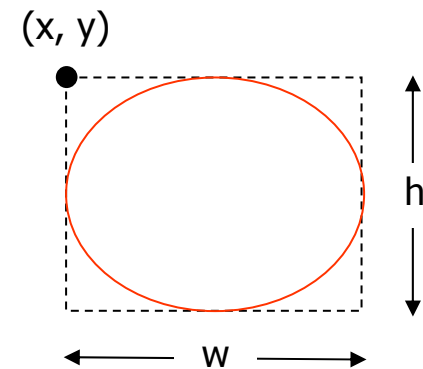
---

---

```
g.setColor(Color.red);  
g.fillOval(x, y, w, h);
```



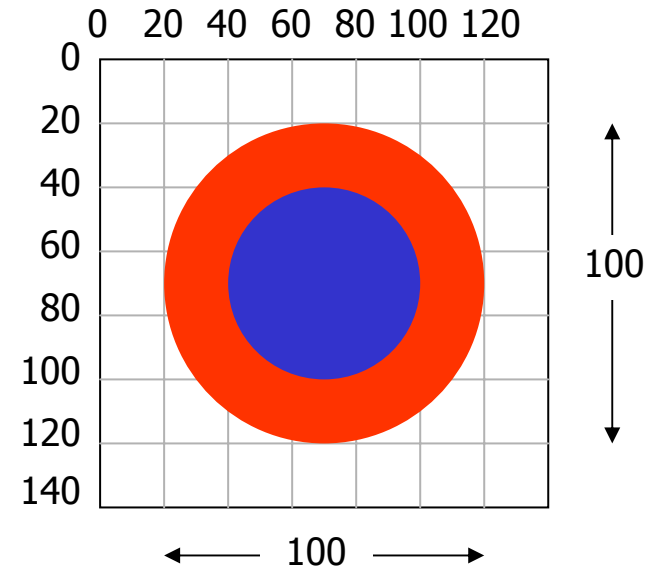
```
g.setColor(Color.red);  
g.drawOval(x, y, w, h);
```



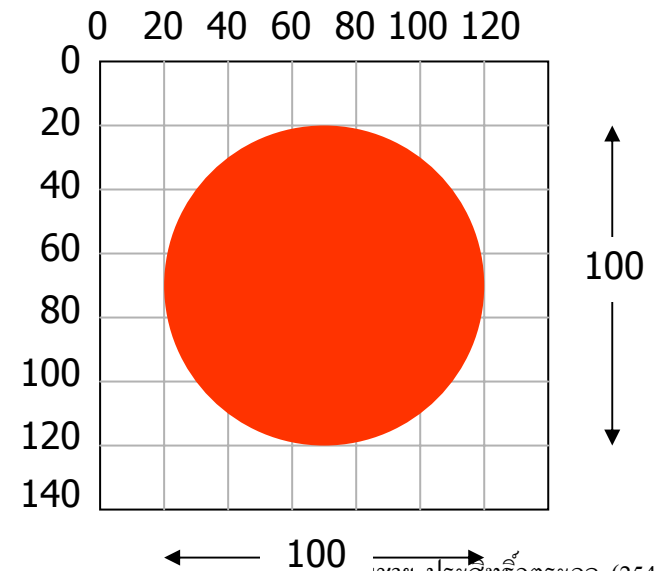


# พิกัดบนจอภาพ

```
public void paint(Graphics g) {  
    g.setColor(Color.red);  
    g.fillOval(20, 20, 100, 100);  
    g.setColor(Color.blue);  
    g.fillOval(40, 40, 60, 60);  
}
```



```
public void paint(Graphics g) {  
    g.setColor(Color.blue);  
    g.fillOval(40, 40, 60, 60);  
    g.setColor(Color.red);  
    g.fillOval(20, 20, 100, 100);  
}
```



# ตัวอย่างโปรแกรมวาดรูป

---

---

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

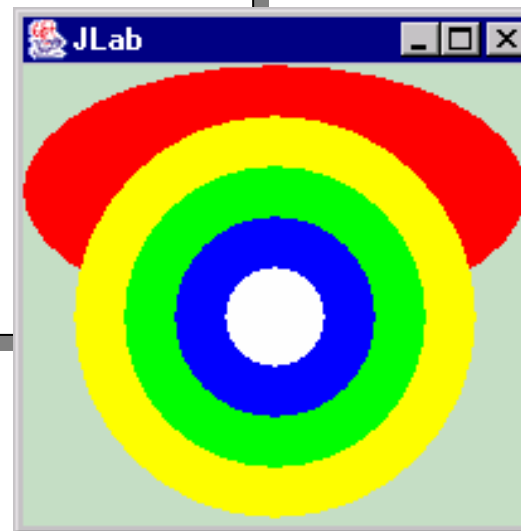
public class Monster extends Applet {

    // main method
    public static void main(String[] args) {
        Frame frame = new Frame("JLab");
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        frame.add(new Monster(), BorderLayout.CENTER);
        frame.setSize(210, 210);
        frame.setVisible(true);
    }
}
```

## ตัวอย่างโปรแกรมวาดรูป (ต่อ)

---

```
public void paint(Graphics g) {  
    g.setColor(Color.red);  
    g.fillOval(0, 0, 200, 100);  
    g.setColor(Color.yellow);  
    g.fillOval(20, 20, 160, 160);  
    g.setColor(Color.green);  
    g.fillOval(40, 40, 120, 120);  
    g.setColor(Color.blue);  
    g.fillOval(60, 60, 80, 80);  
    g.setColor(Color.white);  
    g.fillOval(80, 80, 40, 40);  
}  
  
} // end of Monster class
```



# โปรแกรมหาผลบวก 1 ถึง 10

---

---

```
public class Add1to10 {  
    public static void main(String [] args) {  
        System.out.println(55);  
    }  
}
```

```
public class Add1to10 {  
    public static void main(String [] args) {  
        System.out.println(1+2+3+4+5+6+7+8+9+10);  
    }  
}
```

```
public class Add1to10 {  
    public static void main(String [] args) {  
        double x = 1+2+3+4+5+6+7+8+9+10;  
        System.out.println(x);  
    }  
}
```

# โปรแกรมหาค่าเฉลี่ยของสี่จำนวน

---

---

```
import jlab.JLabIO;

public class Mean {
    public static void main(String [] args) {
        double x = 0.0;
        x += JLabIO.readDouble("data #1 : ");
        x += JLabIO.readDouble("data #2 : ");
        x += JLabIO.readDouble("data #3 : ");
        x += JLabIO.readDouble("data #4 : ");
        System.out.println("mean = " + x / 4.0 );
    }
}
```

```
JLab>javac Mean.java
JLab>java Mean
data #1 : 10
data #2 : 12
data #3 : 14
data #4 : 14
mean = 12.5
JLab>
```

# การทำโปรแกรม

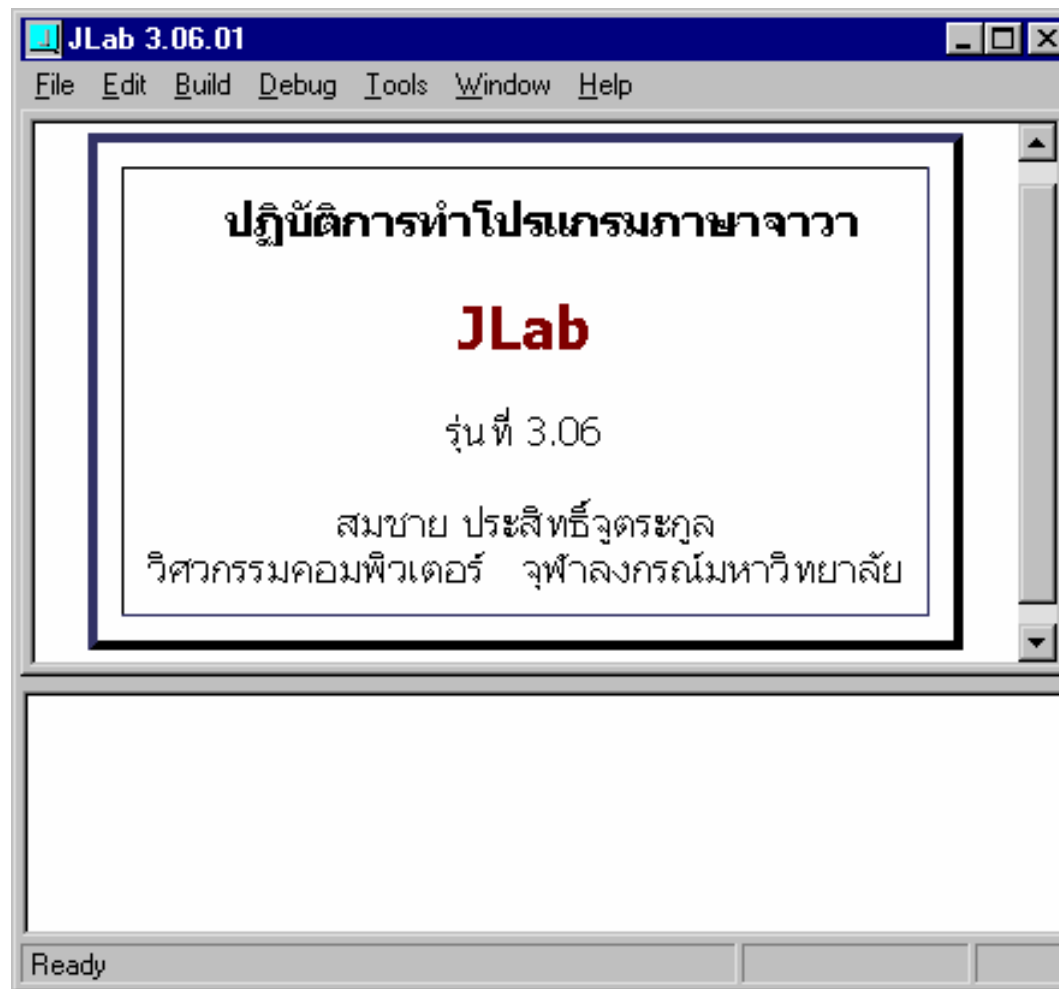
---

---

- edit : เขียนโปรแกรมด้วยภาษาคอมพิวเตอร์
- compile : แปลโปรแกรมที่เขียนให้เป็นรหัสเครื่อง
- execute : สั่งงานรหัสเครื่องที่ได้
- เรามักใช้ Integrated programming environment software เป็นเครื่องมือในการพัฒนาซอฟต์แวร์แบบเบ็ดเสร็จสำหรับวงจรการ "edit-compile-execute-debug"
- สำหรับวิชานี้เราจะใช้ JLab เป็นซอฟต์แวร์ในการช่วยพัฒนาโปรแกรม

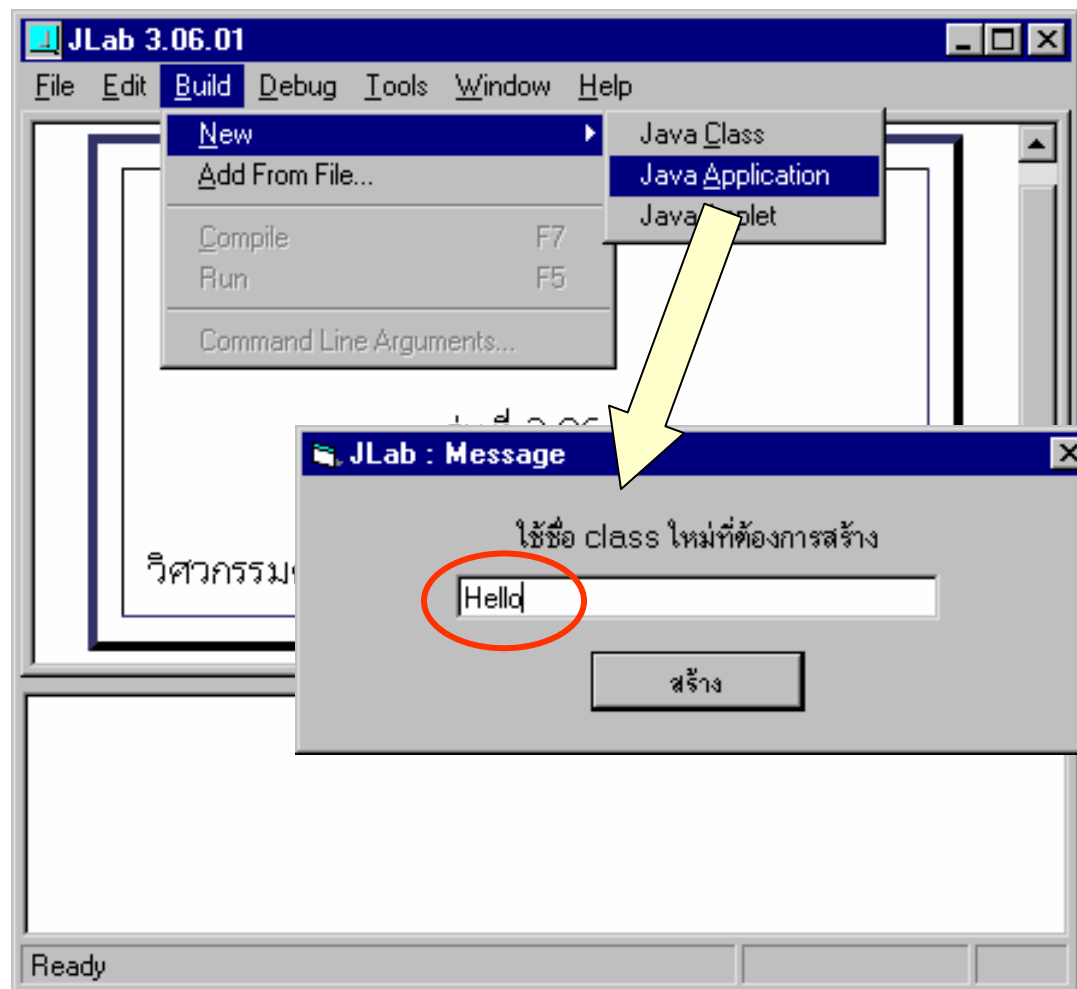
# JLab

---



# JLab : การสร้างแฟ้มโปรแกรม

---

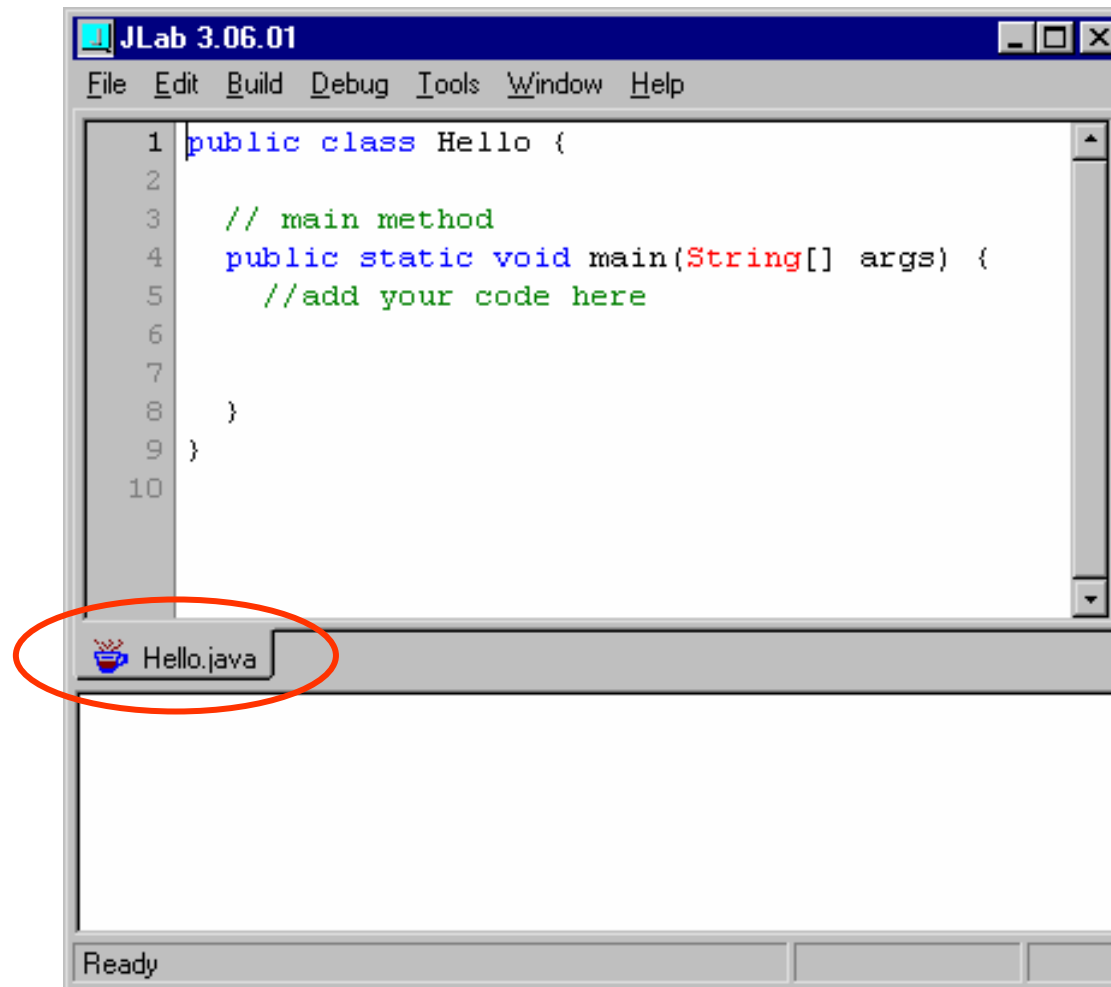




# JLab : การสร้างแฟ้มโปรแกรม

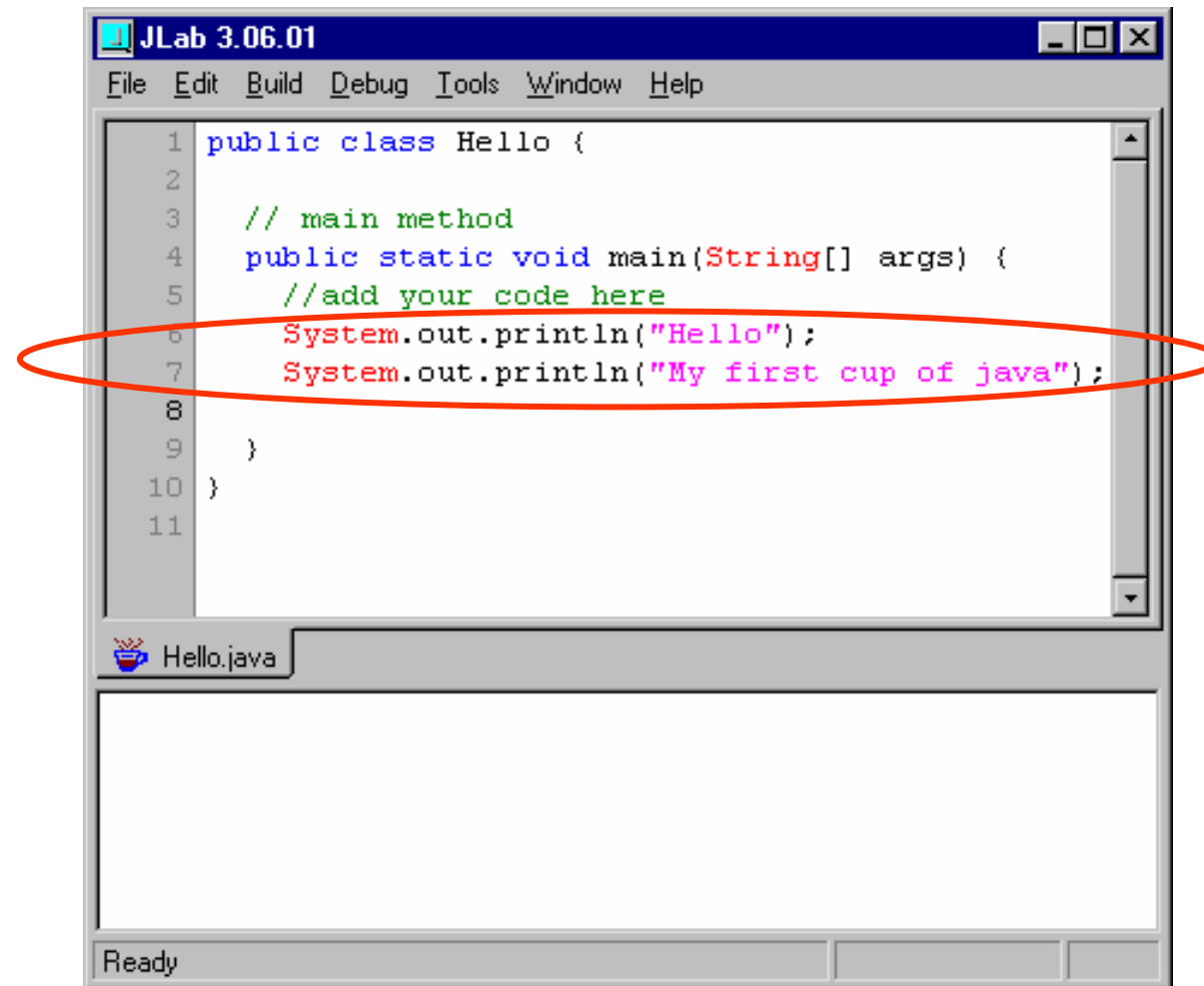
---

---



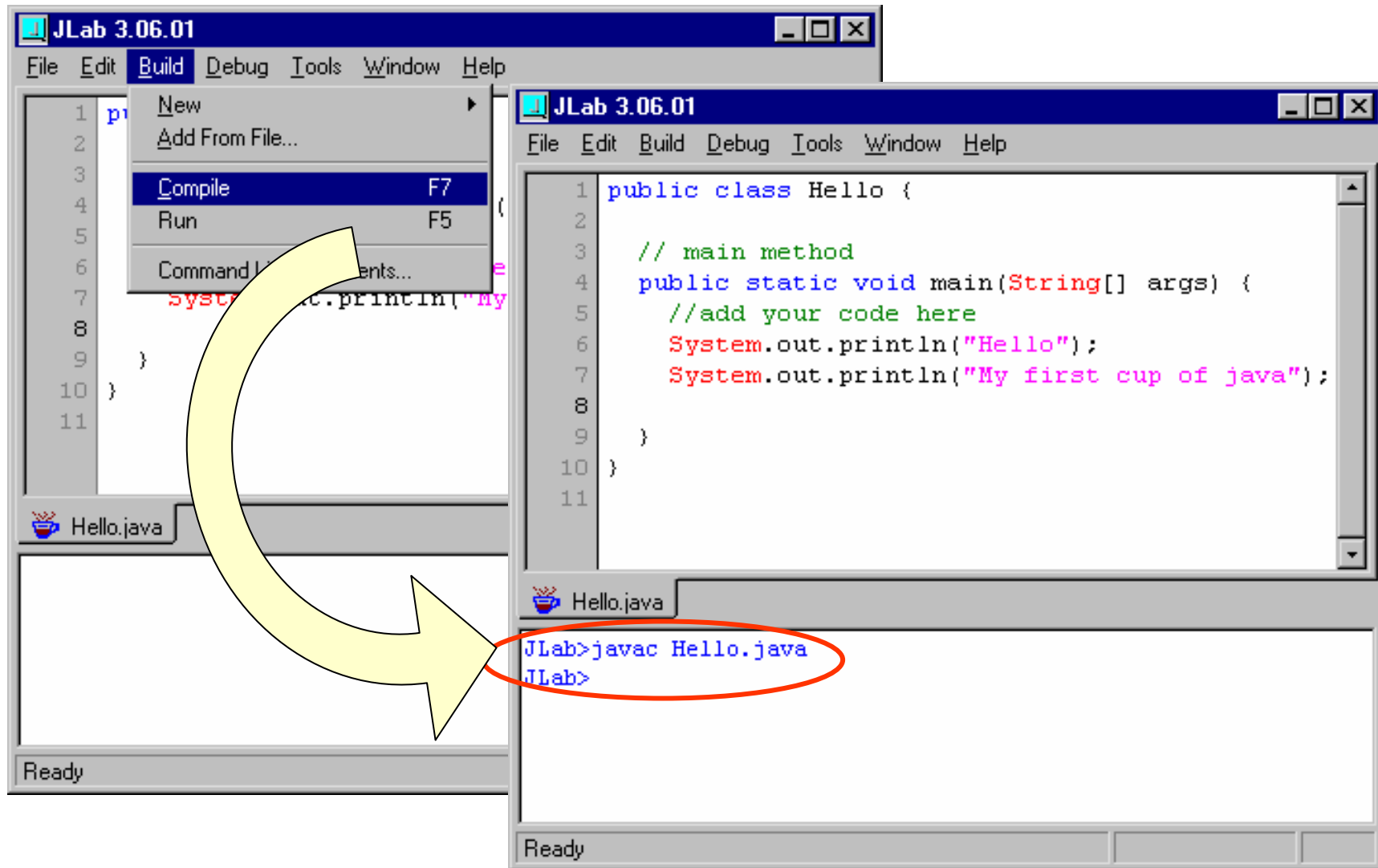
# JLab : การเขียนโปรแกรม

---

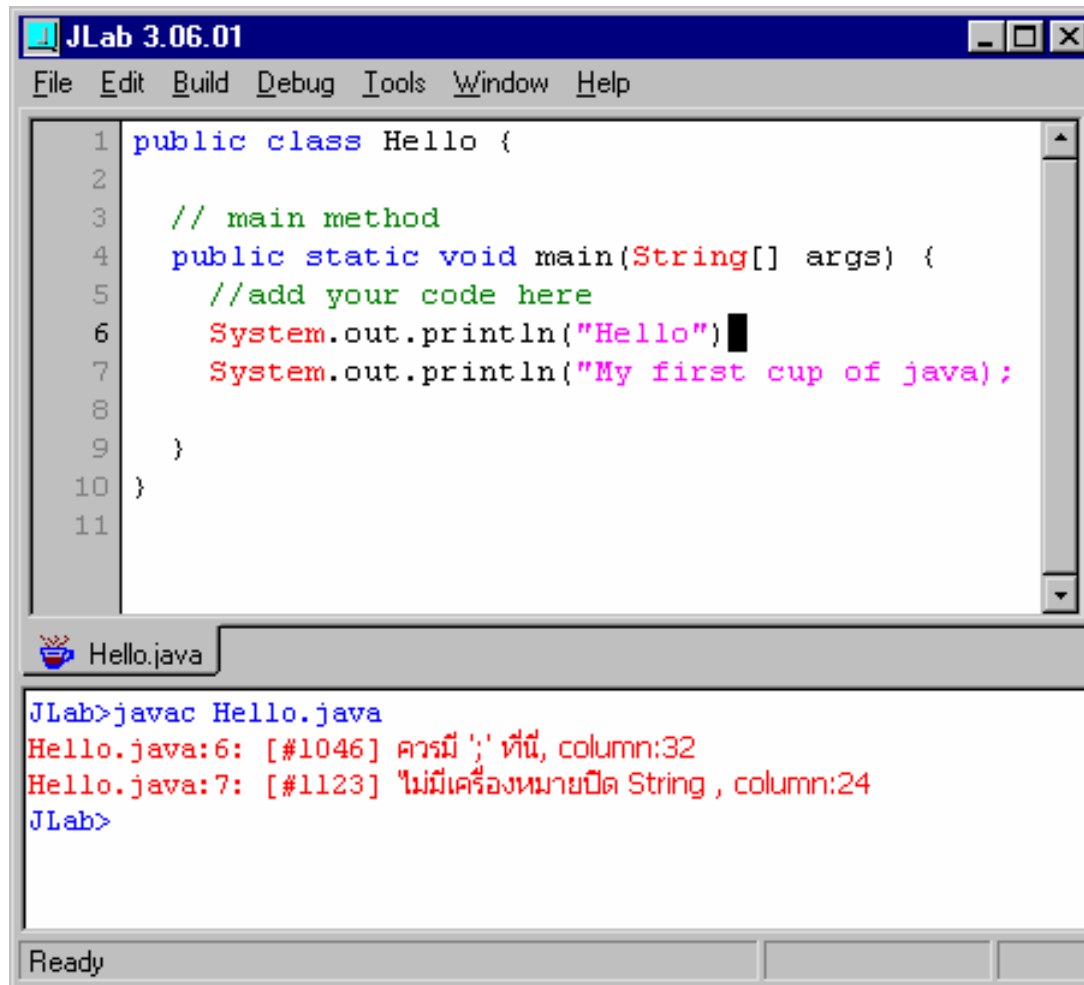


```
1 public class Hello {
2
3 // main method
4 public static void main(String[] args) {
5 //add your code here
6 System.out.println("Hello");
7 System.out.println("My first cup of java");
8
9 }
10 }
11
```

# JLab : การแปลโปรแกรม



# JLab : โปรแกรมที่ผิดกฎ



```
JLab 3.06.01
File Edit Build Debug Tools Window Help

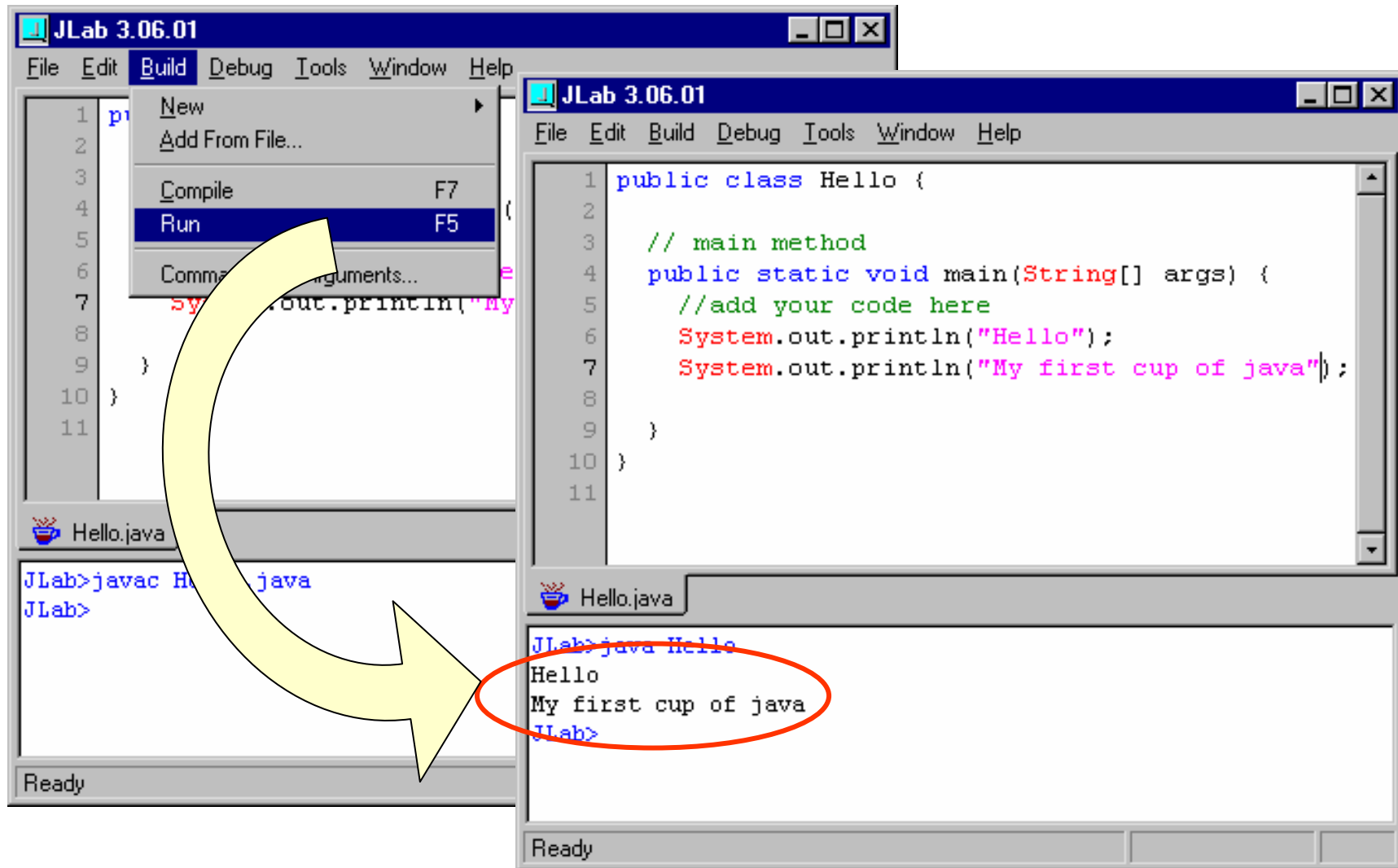
1 public class Hello {
2
3 // main method
4 public static void main(String[] args) {
5 //add your code here
6 System.out.println("Hello");
7 System.out.println("My first cup of java);
8
9 }
10 }
11

Hello.java

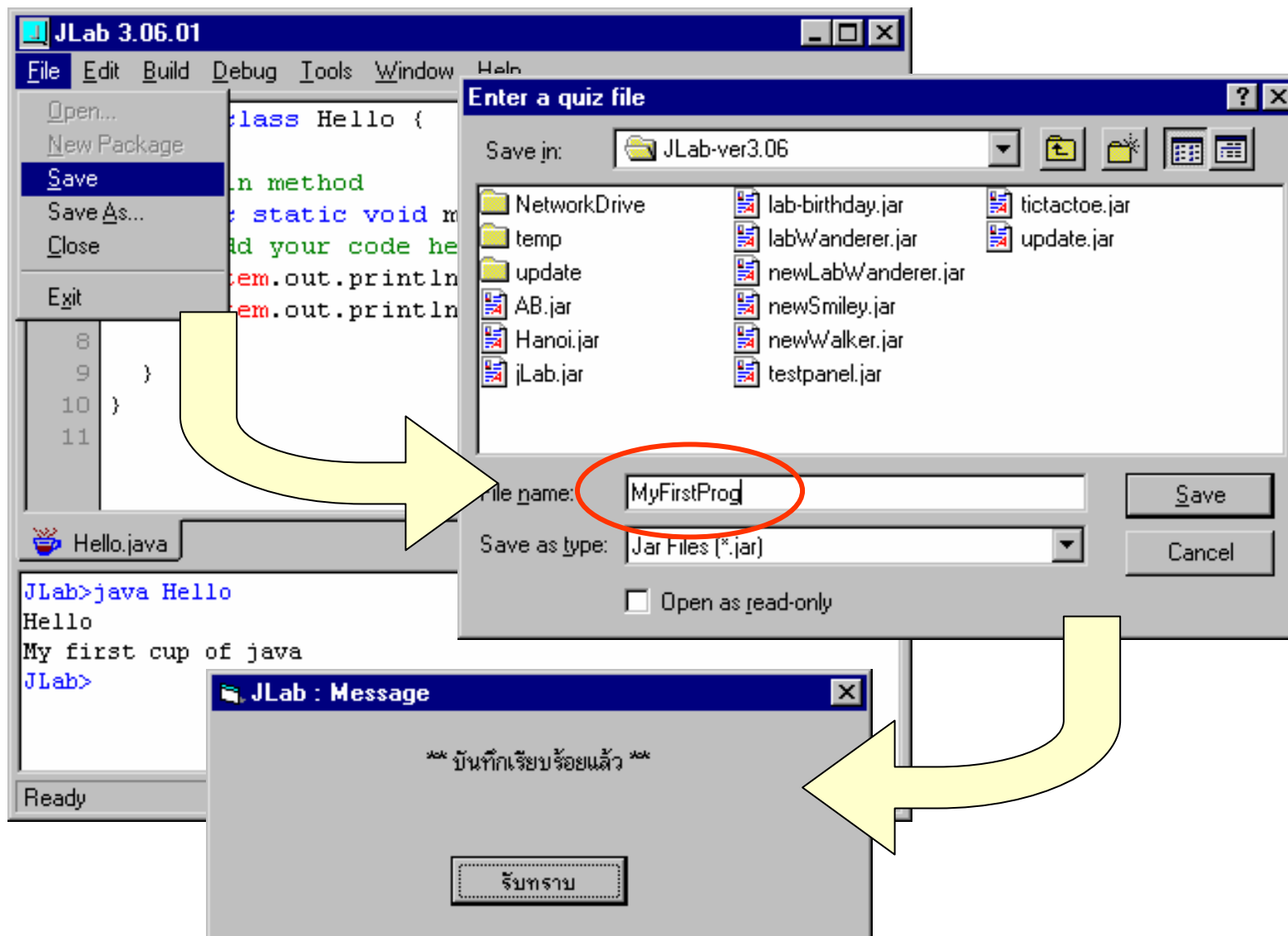
JLab>javac Hello.java
Hello.java:6: [#1046] ควรมี ';' ที่นี่, column:32
Hello.java:7: [#1123] ไม่มีเครื่องหมายปิด String , column:24
JLab>

Ready
```

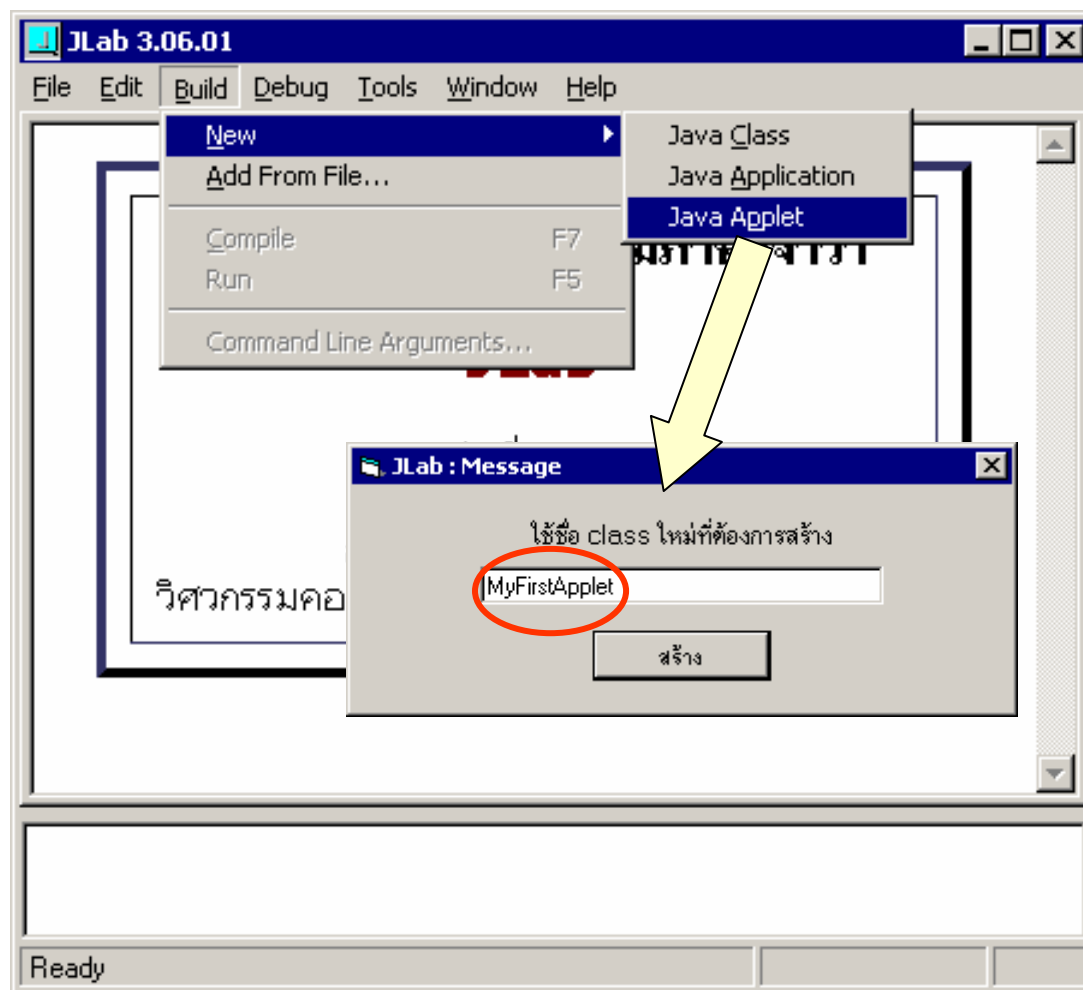
# JLab : การสั่งโปรแกรมให้ทำงาน



# JLab : การบันทึก

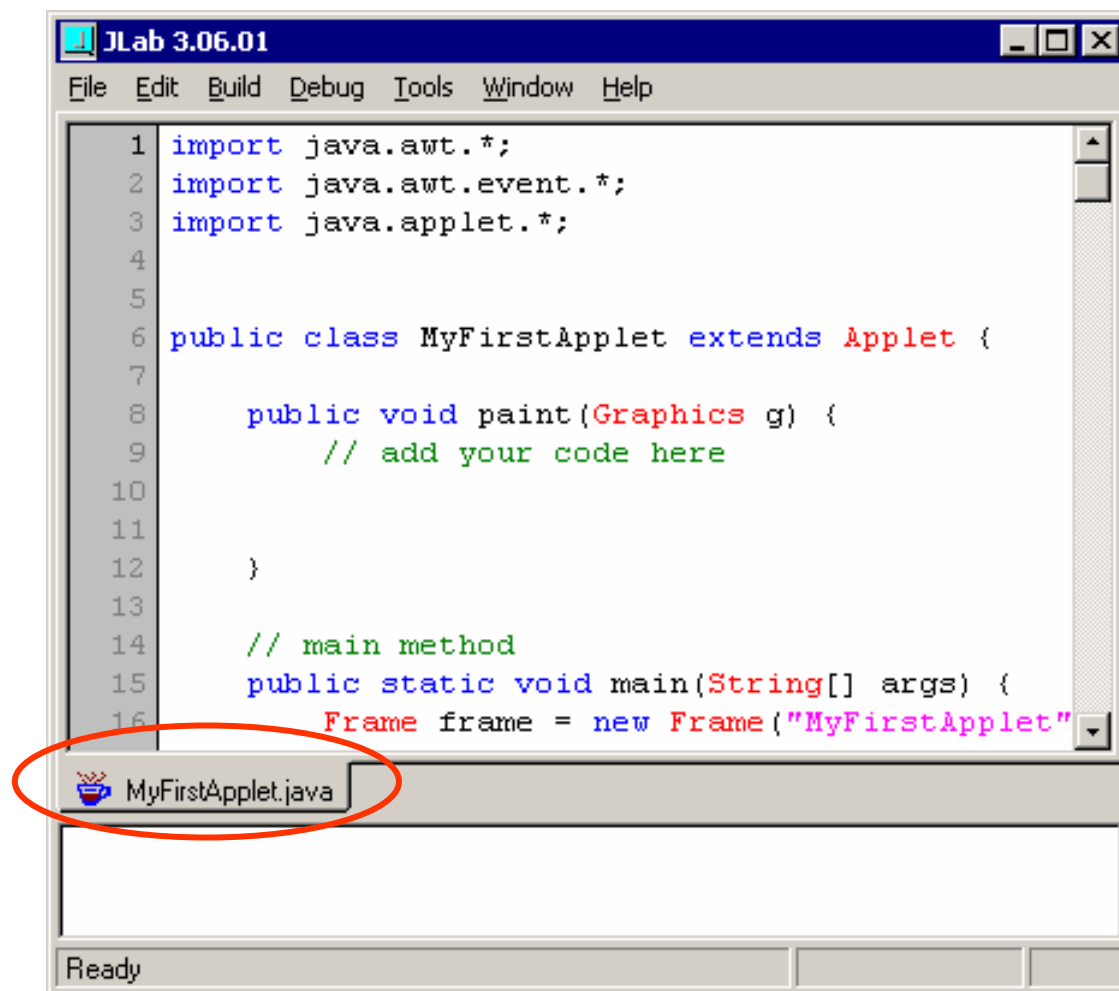


# JLab : การสร้างโปรแกรมแบบกราฟิก



# JLab : การสร้างโปรแกรมแบบกราฟิก

---



```
JLab 3.06.01
File Edit Build Debug Tools Window Help
1 import java.awt.*;
2 import java.awt.event.*;
3 import java.applet.*;
4
5
6 public class MyFirstApplet extends Applet {
7
8     public void paint(Graphics g) {
9         // add your code here
10
11
12     }
13
14     // main method
15     public static void main(String[] args) {
16         Frame frame = new Frame("MyFirstApplet")
```

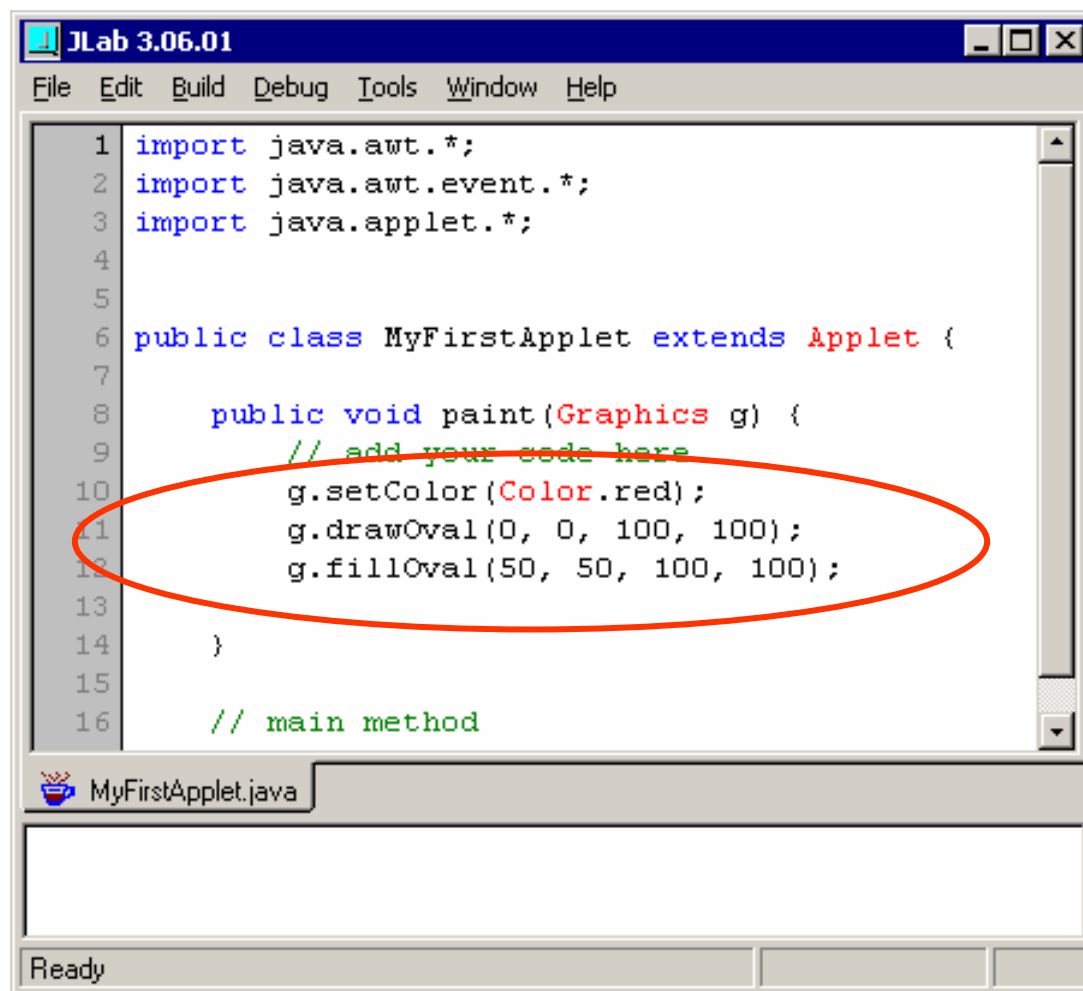
MyFirstApplet.java

Ready



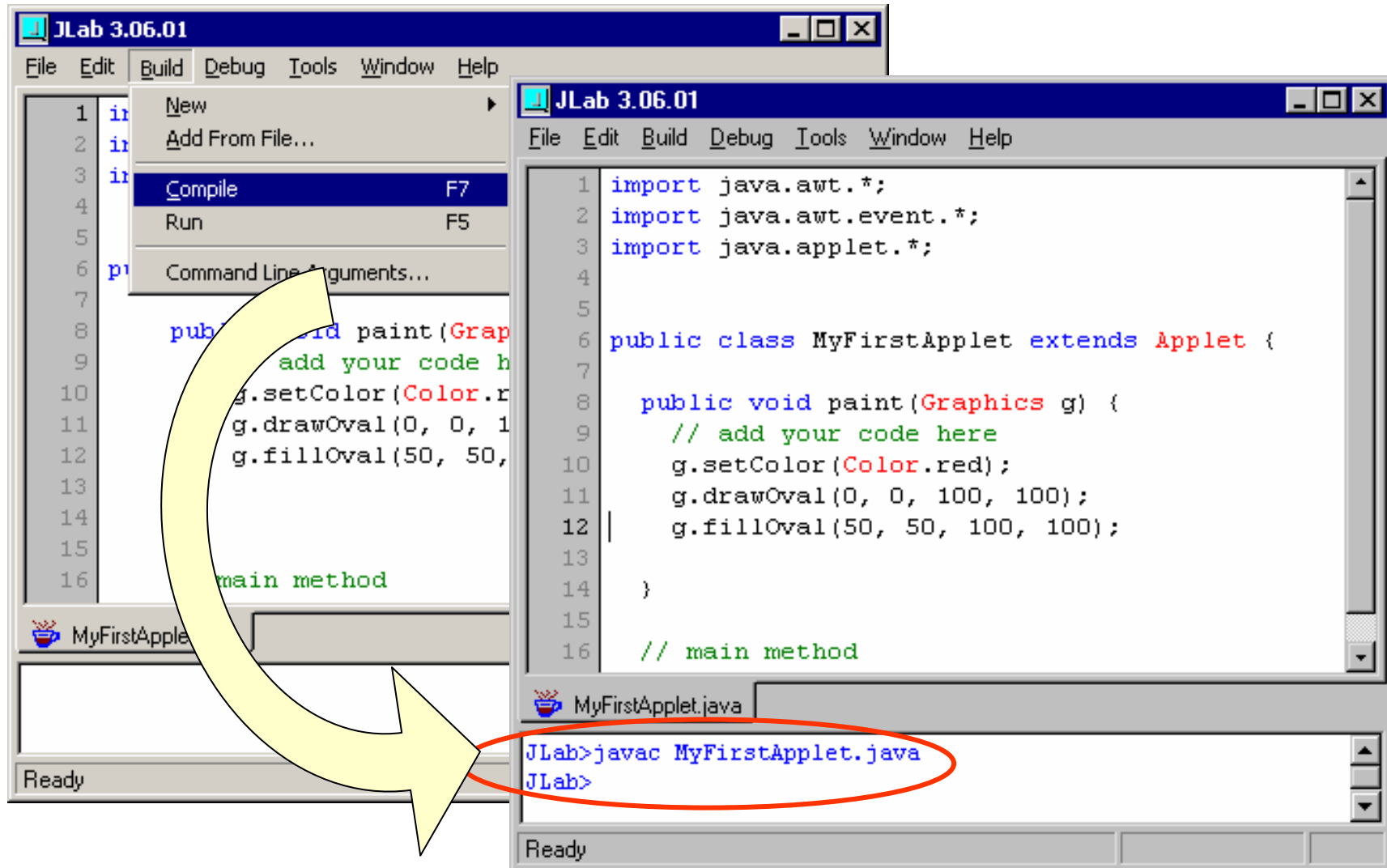
# JLab : การเขียนโปรแกรม

---

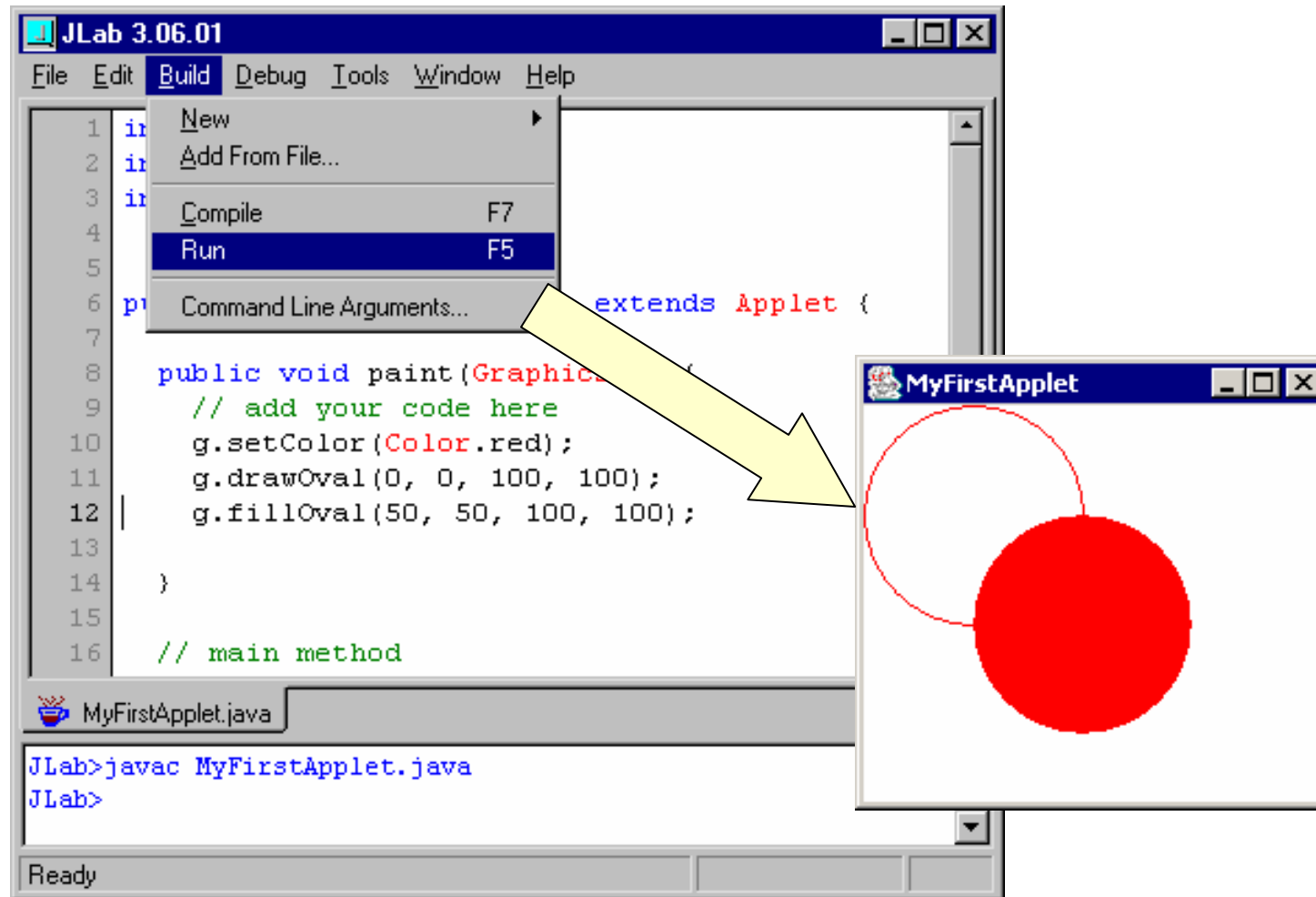


```
1 import java.awt.*;
2 import java.awt.event.*;
3 import java.applet.*;
4
5
6 public class MyFirstApplet extends Applet {
7
8     public void paint(Graphics g) {
9         // add your code here
10        g.setColor(Color.red);
11        g.drawOval(0, 0, 100, 100);
12        g.fillOval(50, 50, 100, 100);
13
14    }
15
16    // main method
```

# Jlab : การแปลโปรแกรม



# JLab : การสั่งโปรแกรมทำงาน



# สรุป

---

---

- โปรแกรม Java ประกอบด้วย class
- การทำงานของ Java application จะเริ่มที่ main
- ควรเขียนโปรแกรมให้อ่านง่ายๆ
- บรรทัดคำสั่งต่างๆ ต้องปิดท้ายด้วย semicolon ;
- การทำงานจะทำทีละคำสั่งตามลำดับจากบนลงล่าง
- การทำโปรแกรมเป็นวงจรการ edit-compile-execute-debug