

## ใบงานการทดลองที่ 2

### เรื่อง ภาษาจาวากับการโปรแกรมเชิงวัตถุ

#### 1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการโปรแกรมเชิงวัตถุด้วยคำสั่งพื้นฐานในภาษาจาวา
- 1.2. รู้และเข้าใจเพื่อศึกษาและทดลองสร้างโปรแกรมเชิงวัตถุโดยใช้ภาษาการโปรแกรมเชิงวัตถุใหม่ๆ

#### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

#### 3. ทฤษฎีการทดลอง

- 3.1. คำสั่งที่ใช้ในการแสดงผลบนแถบ Console คือคำสั่งใด

เราสามารถใช้งาน console.log() แทนได้ โดยค่าจะถูกแสดงในส่วนของ console

- 3.2. คำสั่งการคอมเมนต์แบบบรรทัดเดียว และแบบหลายบรรทัด คืออะไร ?

// ใช้ในการใส่คำอธิบายแบบบรรทัดเดียว โดยจะมีผลให้ข้อความใดๆ หลังจากเครื่องหมาย // ไปจนสุดบรรทัดนั้นๆ เป็นคำอธิบายทั้งหมด

/\*..\*/ ใช้ในการใส่คำอธิบายแบบหลายบรรทัด โดยจะมีผลให้ข้อความใดๆ ที่อยู่ระหว่าง /\* และ \*/ กลายเป็นคำอธิบาย(อาจจะ เป็น 1 บรรทัดหรือมากกว่าก็ได้)

- 3.3. ประเภทข้อมูลแบบ Integer คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

เลขจำนวนเต็มทั่วไป ไม่ว่าจะเป็นเลขจำนวนเต็มบวก จำนวนเต็มศูนย์และจำนวนเต็มลบ ซึ่งเลขจำนวนเต็มเหล่านี้ สามารถนำไปคำนวณได้ ตัวอย่าง เช่น 100, 56, 0, -20 เป็นต้น

- 3.4. ประเภทข้อมูลแบบ Double คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

เป็นตัวแปรที่เก็บข้อมูลที่เป็นเลขทศนิยมเหมือนกับ float แต่จะ ใช้พื้นที่ในการเก็บมากกว่าเดิม 2 เท่า คือมีขนาด 8 ไบต์ มีค่าระหว่าง 1.7E-308 ถึง 1.7E+308

- 3.5. ประเภทข้อมูลแบบ Float คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

เลขทศนิยมชนิดคงที่ หรืออาจจะเป็นทศนิยม แบบไม่รู้จัก หรืออาจจะเป็นเลขทศนิยมที่เขียนในรูป E (หรือ e) ยกกำลัง ตัวเลขทศนิยมเหล่านี้ สามารถนำมาใช้ในการคำนวณได้ ตัวอย่าง เลขทศนิยมนี้ได้แก่ 20.25, -0.60, 58.96, 5.40e04 เป็นต้น

- 3.6. ประเภทข้อมูลแบบ Bool คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

ข้อมูลที่มีค่าเป็น 0 ถือว่าเป็นเท็จ (False) หรือมีค่าเป็น 1 ถือว่าเป็นจริง (True) หรือจะเป็น yes/no , on/off ก็ได้

- 3.7. ประเภทข้อมูลแบบ Char คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

เป็นตัวอักษร หรือสัญลักษณ์อื่น ๆ ที่มีความยาว เพียง 1 ตัวอักษรเท่านั้น ซึ่งอาจจะเป็นตัวอักษร A-Z, a-z, 0-9 หรือ #, @, \$ และอื่น ๆ เป็นต้น โดยจะเขียนไว้ในเครื่องหมาย ' ' (Single Quote) เช่น 'F', 'M', 'T', '0', '1', '{', '#', '@' ตัวอักษรทั้งหมดนั้นสามารถศึกษาหรือดู รายละเอียดเพิ่มเติมได้จากตารางรหัส ASCII (American Standard Code II)

3.8. ประเภทข้อมูลแบบ String คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

เป็นข้อมูลแบบตัวอักษรที่มีความยาวมากกว่า 1 ตัวอักษร มาเรียงต่อกันเป็นข้อความ โดยที่ข้อความนั้นจะต้องถูกเขียนไว้ในเครื่องหมาย " " (Double Quote) ตัวอย่างเช่น "Chiang mai", "Welcome" เป็นต้น

3.9. Widening Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เป็นการกำหนดค่าของ Variable ที่มี Data Type ลำดับต่ำกว่า ให้กับ Variable ที่มี Data Type สูงกว่า ตัวอย่าง

```
Int i = 4 ;
```

```
long l = i ; นิพจน์ i จะถูกปรับชนิดข้อมูลจาก int ให้เป็น long โดยอัตโนมัติ
```

3.10. Narrowing Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เป็นการกำหนดค่าของ Variable ที่มี Data Type ลำดับสูงกว่า ให้กับ Variable ที่มี Data Type ต่ำกว่า ตัวอย่าง

```
Double d = 4.97542;
```

```
int i = (int) d.doubleValue(); จาก d ที่เป็น ดับเบิล แต่ จะถูก Cast ให้กลายเป็น int จะได้ i = 4
```

3.11. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการทางคณิตศาสตร์ดังต่อไปนี้

| ตัวดำเนินการ | ความหมาย                                   | ตัวอย่างการใช้งาน |
|--------------|--|-------------------|
| +            | รวมค่าสองค่าเข้าด้วยกัน                    | a + b             |
| -            | ลบค่าหนึ่งออกจากอีกค่าหนึ่ง                | a - b             |
| *            | คูณสองค่า                                  | a * b             |
| /            | หารค่าหนึ่งด้วยค่าหนึ่ง แบบไม่เอาเศษ       | a / b             |
| ++           | เพิ่มค่าของตัวแปรทีละ 1                    | ++a , a++         |
| --           | ลดค่าของตัวแปรลงทีละ 1                     | --a , a--         |
| %            | หารค่าหนึ่งด้วยค่าหนึ่ง แบบเอาเศษจากการหาร | a % b             |

3.12. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อกำหนดค่าดังต่อไปนี้ (ขอไปซ้าย)

| ตัวดำเนินการ | ความหมาย   | ตัวอย่างการใช้งาน  |
|--------------|--|--------------------|
| =            | เท่ากับ  | x = 5              |
| +=           | บวกค่าด้วยตัวแปร   | x += 5 → x = x + 5 |
| -=           | ลดค่าด้วยตัวแปร  | x -= 5 → x = x - 5 |
| *=           | คูณค่าด้วยตัวแปร   | x *= 5 → x = x * 5 |
| /=           | หารค่าด้วยตัวแปร   | x /= 5 → x = x / 5 |
| %=           | หารแบบเอาเศษด้วยตัวแปร   | x %= 5 → x = x % 5 |
| &=           | การเปรียบเทียบในแต่ละคู่ของบิตที่ตรงกัน เนื่องจากตัวดำเนินการ & ผลลัพธ์ที่ได้จะเป็น 1 ถ้าหากบิตทั้งสองเป็น 1 ไม่งั้นนั้นจะเป็น 0           | x &= 5 → x = x & 5 |
| =            | การเปรียบเทียบในแต่ละคู่ของบิตที่ตรงกัน เนื่องจากตัวดำเนินการ   หากบิตใดบิตหนึ่งมีค่าเป็น 1 จะได้ผลลัพธ์เป็น 1 นอกจากนี้จะได้ผลลัพธ์เป็น 0 | x  = 5 → x = x   5 |

## 3.13. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อเปรียบเทียบดังต่อไปนี้

| ตัวดำเนินการ | ความหมาย                                      | ตัวอย่างการใช้งาน |
|--------------|---|-------------------|
| ==           | เท่ากับ? เปรียบเทียบประมาณว่าเท่ากับกันรึป่าว | x == y            |
| >            | มากกว่า                                       | x > y             |
| >=           | มากกว่าหรือเท่ากับ                            | x >= y            |
| <            | น้อยกว่า                                      | x < y             |
| <=           | น้อยกว่าหรือเท่ากับ                           | x <= y            |
| !=           | ไม่เท่ากับ                                    | x != y            |

## 3.14. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการตรรกศาสตร์ดังต่อไปนี้

| ตัวดำเนินการ | ความหมาย   | ตัวอย่างการใช้งาน   |
|--------------|--|---------------------|
| &&           | and ค่าทางตรรกะ จะเป็นจริงก็ต่อเมื่อทั้งสองข้างเป็นจริง  | x < 5 && x < 10     |
|              | or ค่าทางตรรกะ จะเป็นจริงก็ต่อเมื่ออันใดอันหนึ่งเป็นจริง | x < 5    x < 4      |
| !            | กลับค่าทางตรรกะ ถ้าคำตอบเป็นเท็จก็กลับเป็นจริง           | !(x < 5 && x < 10 ) |

## 3.15. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if พร้อมยกตัวอย่างประกอบ

| การใช้งาน  | Example   |
|--|---|
| <pre>if (condition) { // block of code to be executed if the condition is true }</pre> | <pre>int x = 20; int y = 18; if (x &gt; y) { System.out.println("x is greater than y"); }</pre> |

## 3.16. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if else พร้อมยกตัวอย่างประกอบ

| การใช้งาน   | Example   |
|---|---|
| <pre>if (condition) { // block of code to be executed if the condition is true } else { // block of code to be executed if the condition is false }</pre> | <pre>int time = 25 ; if (time &lt; 31 ) { System.out.println("Good day."); } else { System.out.println("Good evening."); } // Outputs "Good day."</pre> |

## 3.17. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if , else if และ else พร้อมยกตัวอย่างประกอบ

| การใช้งาน   | Example   |
|---|---|
| <pre>if (condition1) { // block of code to be executed if condition1 is true } else if (condition2) { // block of code to be executed if the condition1 is false and condition2 is true } else { // block of code to be executed if the condition1 is false and condition2 is false }</pre> | <pre>int time = 17 ; if (time &lt; 10) { System.out.println("Good morning."); } else if (time &lt; 20) { System.out.println("Good day."); } else { System.out.println("Good evening."); } // Outputs " Good day."</pre> |

## 3.18. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if แบบ Short hand พร้อมยกตัวอย่างประกอบ

| การใช้งาน  | Example  |
|--|--|
| <pre>variable = (condition) ? expressionTrue : expressionFalse ;</pre> | <pre>int time = 25 ; string result = ( time &lt; 18 ) ? "Good boy" : "Good girl" ; System.out.println(result);</pre> |

## 3.19. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง switch พร้อมยกตัวอย่างประกอบ

| การใช้งาน   | Example  |
|---|--|
| <pre>switch(expression) { case x : // code block break; case y: // code block break; default: // code block }</pre> | <pre>int gender = 2 ; switch (gender) { case 1 : System.out.println("Male") ; Break ; Case 2 : System.out.println("Female") ; Break ; } //output "Female" (gender 2)</pre> |

## 3.20. การวนรอบ – จงเขียน Syntax การใช้งานคำสั่ง for พร้อมยกตัวอย่างประกอบ

| การใช้งาน   | Example  |
|---|--|
| <pre>for (statement 1; statement 2; statement 3) { // code block to be executed }</pre> | <pre>for (int i = 0; i &lt; 5; i++) { System.out.println(i); }</pre> |

## 3.21. การวนรอบ – จงเขียน Syntax การใช้งานคำสั่ง while พร้อมยกตัวอย่างประกอบ

| การใช้งาน   | Example   |
|---|---|
| <pre>while (condition) {     // code block to be executed }</pre> | <pre>int i = 0; while (i &lt; 5) {     System.out.println(j) ;     i++; }</pre> |

## 3.22. การวนรอบ – จงเขียน Syntax การใช้งานคำสั่ง do while พร้อมยกตัวอย่างประกอบ

| การใช้งาน  | Example  |
|--|--|
| <pre>do {     // code block to be executed } while (condition) ;</pre> | <pre>int i = 0 do {     System.out.println(i) ;     i++ ; } while (i &lt; 5) ;</pre> |

## 3.23. อธิบายการทำงานของคำสั่ง break พร้อมยกตัวอย่างประกอบ

| การทำงาน  | Example   |
|---|---|
| คำสั่ง break เป็นคำสั่งที่ใช้เพื่อควบคุมลูปให้จบการทำงานในทันที โดยที่ไม่สนใจว่าเงื่อนไขจะยังคงเป็นจริงอยู่หรือไม่ โดยจะกลับไปทำงานต่อจากคำสั่งที่อยู่ต่อจากคำสั่ง switch() | <pre>for (int i = 0; i &lt; 10; i++) {     if (i == 3) {         break;     }     System.out.println(i) ; }</pre> |

## 3.24. อธิบายการทำงานของคำสั่ง continue พร้อมยกตัวอย่างประกอบ

| การทำงาน  | Example  |
|---|--|
| คำสั่ง continue ใช้สำหรับควบคุมเพื่อข้ามการทำงานของลูปในรอบปัจจุบันไปยังรอบใหม่ในทันที โดยที่โปรแกรมจะข้ามคำสั่งที่ปรากฏหลังจากคำสั่ง continue ทั้งหมด; ไม่เหมือนกับคำสั่ง break คำสั่ง continue ยังคงทำงานในลูปในขณะที่เงื่อนไขเป็นจริงตามปกติ | <pre>for (int i = 0; i &lt; 10; i++) {     if (i == 4) {         continue;     }     System.out.println(i) ; }</pre> |

## 3.25. อธิบายการทำงานของคำสั่ง Enum

Enum เป็นประเภทข้อมูลที่เป็นชุดของตัวเลขจำนวนเต็ม มันใช้กำหนดกลุ่มของค่าคงที่ที่มีความหมายและเข้าใจได้ง่ายขึ้นในการเขียนโปรแกรม โดยคำที่ใช้ในการประกาศ enum นั้นจะเป็น String literal ใดๆ โดยค่าเหล่านี้ถือว่าเป็นค่าคงที่ของจำนวนเต็ม โดยคำสั่ง enum นั้นจะใช้สำหรับการสร้าง enum และ Name เป็นชื่อของ enum ที่สร้างขึ้น และภายในวงเล็บ {} นั้นจะเป็นชุดค่าของ enum โดยเราสามารถเป็น literal ใดๆ อนุญาตให้ทำได้ ค่าของ enum สามารถมีได้ตั้งแต่หนึ่งหรือหลายค่า โดยแต่ละค่านั้นจะคั่นด้วยเครื่องหมาย Commas

## 4. ลำดับขั้นการปฏิบัติการ

## 4.1. จงแก้ไขปัญหาดังต่อไปนี้

4.1.1. จงเขียนโปรแกรมเพื่อแสดงเลข\ 1 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้และแสดงคำตอบบรรทัดละ 5 ตัวเลขเท่านั้น \

| Test case 1  | Test case 2  |
|--|--|
| Please enter your value : 8<br>-----<br>1, 2, 3, 4, 5<br>6, 7, 8 | Please enter your value : 21<br>-----<br>1, 2, 3, 4, 5<br>6, 7, 8, 9, 10<br>11, 12, 13, 14, 15<br>16, 17, 18, 19, 20<br>21 |

4.1.2. ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)

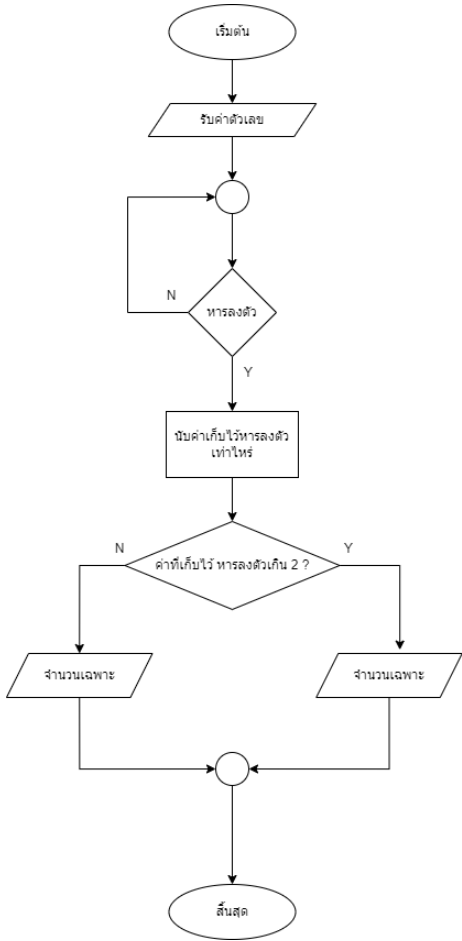
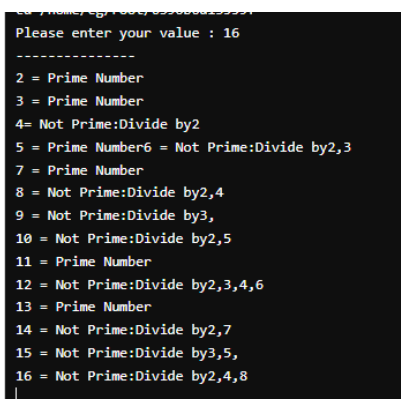
| ผังงาน  | โค้ดโปรแกรม  |
|---|--|
| <pre> graph TD     Start([เริ่มต้น]) --&gt; Input[/ใส่เลข/]     Input --&gt; LoopStart(( ))     LoopStart --&gt; Cond1{i &lt;= เลข?}     Cond1 -- N --&gt; End([สิ้นสุด])     Cond1 -- Y --&gt; Cond2{i%5==0?}     Cond2 -- Y --&gt; Print1[print i]     Print1 --&gt; LoopStart     Cond2 -- N --&gt; Print2[print i]     Print2 --&gt; Inc[i++]     Inc --&gt; LoopStart     </pre> | <pre> 1 package LAB00P; 2 3 import java.util.Scanner; // Import the Scanner class 4 5 public class LAB2_1 { 6 7     public static void main(String[] args) { 8 9         Scanner myObj = new Scanner(System.in); 10        int num; 11 12        System.out.print("Please enter your value : "); 13        num = myObj.nextInt(); 14        System.out.println("-----"); 15 16        for (int i = 1; i &lt;= num; i++) { 17            if (i%5 == 0) { 18                System.out.println(i); 19                continue; 20            } 21            if (i &gt;= num){ 22                System.out.println(i); 23                break; 24            } 25            System.out.print(i); 26            System.out.print(','); 27        } 28    } 29 } </pre> <pre> Please enter your value : 21 ----- 1,2,3,4,5 6,7,8,9,10 11,12,13,14,15 16,17,18,19,20 21 </pre> |

4.2. จงแก้ไขปัญหาดังต่อไปนี้

## 4.2.1. จงเขียนโปรแกรมเพื่อแสดงจำนวนเฉพาะที่อยู่ระหว่าง เลข 2 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้

| Test case 1   | Test case 2   |
|---|---|
| <p>Please enter your value : <b>8</b></p> <p>-----</p> <p>2 = Prime Number</p> <p>3 = Prime Number</p> <p>4 = Not Prime : Divide by 2</p> <p>5 = Prime Number</p> <p>6 = Not Prime : Divide by 2, 3</p> <p>7 = Prime Number</p> <p>8 = Not Prime : Divide by 2, 4</p> | <p>Please enter your value : <b>16</b></p> <p>-----</p> <p>2 = Prime Number</p> <p>3 = Prime Number</p> <p>4 = Not Prime : Divide by 2</p> <p>5 = Prime Number</p> <p>6 = Not Prime : Divide by 2, 3</p> <p>7 = Prime Number</p> <p>8 = Not Prime : Divide by 2, 4</p> <p>9 = Not Prime : Divide by 3</p> <p>10 = Not Prime : Divide by 2, 5</p> <p>11 = Prime Number</p> <p>12 = Not Prime : Divide by 2, 3, 4, 6</p> <p>13 = Prime Number</p> <p>14 = Not Prime : Divide by 2, 7</p> <p>15 = Not Prime : Divide by 2, 3, 5</p> <p>16 = Not Prime :Divide by 2, 4, 8</p> |

## 4.2.2. ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)

| ผังงาน  | โค้ดโปรแกรม   |
|---|---|
|  <pre> graph TD     Start([เริ่มต้น]) --&gt; Input[/รับค่าตัวเลข/]     Input --&gt; Loop(( ))     Loop --&gt; Div{หารลงตัว}     Div -- N --&gt; Prime[/จำนวนเฉพาะ/]     Div -- Y --&gt; Divisor[/ป้อนค่าเก็บไว้หารลงตัวเท่าไร?/]     Divisor --&gt; Div2{ค่าที่เก็บไว้หารลงตัวเกิน 2?}     Div2 -- N --&gt; Prime     Div2 -- Y --&gt; Divisor     Prime --&gt; Join(( ))     Divisor --&gt; Join     Join --&gt; End([สิ้นสุด])   </pre> | <pre> 1 package LAB00P; 2 3 import java.util.Scanner; 4 5 public class LAB2_2 { 6 7     public static void main(String[] args) { 8         int count; 9         int num; 10        Scanner myObj = new Scanner(System.in); 11 12        System.out.print("Please enter your value : "); 13        num = myObj.nextInt(); 14        System.out.println("-----"); 15        for ( int i = 2; i &lt;= num; i++ ) { 16            System.out.print(i); 17            System.out.print(" = "); 18            count = 0; 19            for ( int j = 2; j &lt;= i; j++ ) { 20                if( i % j == 0 ) { 21                    count++; 22                } 23            } 24            if( count == 1 ) { 25                System.out.print("Prime Number"); 26            }else if ( count &gt; 1 ) { 27                System.out.print("Not Prime:Divide by"); 28                for ( int j = 2; j &lt;= i/2; j++ ) { 29                    if( i % j == 0 ) { 30                        System.out.print(j); 31                        if( j &gt;= i/2 ) { 32                            continue; 33                        } 34                        System.out.print(","); 35                    } 36                } 37            } 38            System.out.println(); 39        } 40    } 41 }   </pre>  <pre> Please enter your value : 16 ----- 2 = Prime Number 3 = Prime Number 4 = Not Prime:Divide by2 5 = Prime Number6 = Not Prime:Divide by2,3 7 = Prime Number 8 = Not Prime:Divide by2,4 9 = Not Prime:Divide by3, 10 = Not Prime:Divide by2,5 11 = Prime Number 12 = Not Prime:Divide by2,3,4,6 13 = Prime Number 14 = Not Prime:Divide by2,7 15 = Not Prime:Divide by3,5, 16 = Not Prime:Divide by2,4,8   </pre> |



## 5. สรุปผลการปฏิบัติการ

ได้เรียนรู้คำสั่งพื้นฐานในภาษา java ได้ดีมากขึ้น และโจทย์ปัญหาที่ให้อีกก็ช่วยพัฒนาในการออกแบบให้คุ้นเคยกับหลักการและการใช้งานในโปรแกรมมากขึ้นด้วย

## 6. คำถามท้ายการทดลอง

6.1. คำสั่งต่างๆ ระหว่างภาษา C และภาษา Java มีความเหมือนหรือแตกต่างกันอย่างไร ?

ภาษาซี มีเครื่องมืออำนวยความสะดวกสำหรับการเขียนโปรแกรมเชิงโครงสร้างและอนุญาตให้มีขอบข่ายตัวแปร (scope) และการเรียกซ้ำ (recursion) ในขณะที่ระบบชนิดตัวแปรพลวัตก็ช่วยป้องกันการดำเนินการที่ไม่ตั้งใจหลายอย่าง เหมือนกับภาษาโปรแกรมเชิงคำสั่งส่วนใหญ่ในแบบแผนของภาษาอัลกอล

ภาษาจาวาเป็นภาษาโปรแกรมเชิงวัตถุ ซึ่งภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส (C++) โดยรูปแบบที่เพิ่มเติมขึ้น คล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C)

โดยที่ทั้งสองจะมีคำสั่งที่คล้ายกันแต่อาจจะแตกต่างกันบ้าง แต่ก็คล้ายกันอยู่ มีการใช้คำสั่งต่างๆ หรือว่า syntax คล้ายๆ กัน

6.2. จงระบุข้อควรระวังในการใช้คำสั่ง Switch Case

1. switch ไม่สามารถตรวจสอบนิพจน์ชนิดเลขจำนวนจริง ที่มีจุดทศนิยม
2. switch นำมาใช้ตรวจสอบชนิดข้อมูลที่เป็ int หรือ char เท่านั้น
3. การตรวจสอบเงื่อนไขภายใน case ของ switch ในแต่ละกรณี จะไม่สามารถนำตัวแปรมาใช้ได้ จะใช้ได้แต่เพียงค่าคงที่เท่านั้น
4. switch ไม่สามารถตรวจสอบเงื่อนไขหลายๆ ตัวภายในนิพจน์เดียวได้
5. ในการใช้คำสั่ง switch....case มักจะใช้งานร่วมกับคำสั่ง break ถ้าผู้อ่านไม่ใส่คำสั่ง break เมื่อ จบ case ใดๆ คอมไพเลอร์ จะเข้าไปตรวจสอบเงื่อนไขใน case ถัดไป

6.3. การเลือกใช้คำสั่ง For เหมาะกับสถานการณ์ใดมากที่สุด

เหมาะกับการใช้การวนซ้ำหลายๆ ครั้งในจำนวนรอบที่แน่นอน เนื่องจากเราสามารถกำหนดค่าเริ่มต้น เงื่อนไขและการเปลี่ยนแปลงค่าไว้ที่เดียวกัน

6.4. คำสั่ง Casting ควรถูกนำมาใช้งานในสถานการณ์ใดที่สุด

เหมาะกับการใช้เก็บข้อมูลที่มีขนาดที่เกินความจำเป็นหรือว่ามีความจำเป็นที่เราจะต้องเพิ่มขนาดที่ใช้เก็บข้อมูล