

## 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

- println()

- 3.2. คำสั่งการคอมเมนต์แบบบรรทัดเดียว และแบบหลายบรรทัด คืออะไร ?

- แบบบรรทัดเดียว //(ข้อความ)

- แบบหลายบรรทัด /\*(ข้อความ)\*/

- คือการเขียนข้อความเพื่ออธิบายการทำงานของคำสั่งเพื่อให้ผู้อื่นสามารถเข้าใจได้ง่าย

- 3.3. ประเภทข้อมูลแบบ Integer คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

- ข้อมูลเก็บค่า ตัวเลข ที่เก็บค่าแค่จำนวนเต็ม ไม่สามารถเก็บค่าทศนิยมได้

- มีค่า ระหว่าง - 32768 ถึง 32767 ใช้เนื้อที่ในการเก็บ 2 ไบต์

- เช่น int number = 69 ;

- 3.4. ประเภทข้อมูลแบบ Double คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

- ข้อมูลเก็บค่า ตัวเลข ที่เก็บค่าแค่จำนวนเต็ม และสามารถเก็บค่าทศนิยมได้ และมากกว่า Float

- ใช้พื้นที่ในการเก็บมากกว่า Float 2 เท่า คือมีขนาด 8 ไบต์ มีค่าระหว่าง 1.7E-308 ถึง 1.7E+308

- เช่น int double = 76.3423 ;

- 3.5. ประเภทข้อมูลแบบ Float คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

- ข้อมูลเก็บค่า ตัวเลข ที่เก็บค่าแค่จำนวนเต็ม และสามารถเก็บค่าทศนิยมได้

- ใช้พื้นที่ในการเก็บ 4 ไบต์ มีค่าระหว่าง 3.4E-38 ถึง 3.4E+38 หรือ แสดงเป็น เลขทศนิยมได้ไม่เกิน 6 ตำแหน่ง

- เช่น int float = 99.99 ;

- 3.6. ประเภทข้อมูลแบบ Bool คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

- ข้อมูลที่เก็บค่าความเป็นจริง คือ TRUE กับค่าความเป็นเท็จ คือ FALSE

- เช่น Boolean myBool = true ;

- 3.7. ประเภทข้อมูลแบบ Char คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

- ข้อมูลที่เก็บ ตัวอักษร ในรูปแบบตัวอักษร ASCII ใช้เนื้อที่ในการเก็บ 1 ไบต์

- เช่น char = 'A';

- 3.8. ประเภทข้อมูลแบบ String คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

- ข้อมูลแบบตัวอักษรที่มีความยาวมากกว่า 1 ตัวอักษร มาเรียงต่อกันเป็นข้อความ โดยที่ข้อความนั้นจะต้องถูก

เขียนไว้ในเครื่องหมาย " " (Double Quote)

- เช่น char name[13] = "Good\_Bye";

### 3.9. Widening Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

- คือการ เปลี่ยนขนาดที่ใช้เก็บค่าของตัวแปร จากขนาดเล็กไปขนาดที่ใหญ่กว่า
- byte -> short -> char -> int -> long -> float -> double
- เช่น float ที่เปลี่ยนเป็น double เพราะต้องการเก็บทศนิยมที่เก็บได้มากกว่า float

### 3.10. Narrowing Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

- คือการ เปลี่ยนขนาดที่ใช้เก็บค่าของตัวแปร จากขนาดใหญ่ไปขนาดที่เล็กกว่า
- double -> float -> long -> int -> char -> short -> byte
- เช่น int ที่เปลี่ยนเป็น char เพราะต้องการเปลี่ยนจากการเก็บจำนวนเต็มไปเก็บตัวอักษรแทน

### 3.11. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการทางคณิตศาสตร์ดังต่อไปนี้

| ตัวดำเนินการ | ความหมาย                            | ตัวอย่างการใช้งาน |
|--------------|-------------------------------------|-------------------|
| +            | บวก ค่าระหว่างสองตัวแปร             | $X + Y$           |
| -            | ลบ ค่าระหว่างสองตัวแปร              | $X - Y$           |
| *            | คูณ ค่าระหว่างสองตัวแปร             | $X \times Y$      |
| /            | หาร ค่าระหว่างสองตัวแปร             | $X \div Y$        |
| ++           | เพิ่ม ค่าให้ตัวแปร เพิ่มอีก 1 หน่วย | $++X$             |
| --           | ลด ค่าให้ตัวแปร ลดอีก 1 หน่วย       | $--X$             |
| %            | แสดงเศษของการหารนั้น                | $X \% Y$          |

### 3.12. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อกำหนดค่าดังต่อไปนี้

| ตัวดำเนินการ | ความหมาย  | ตัวอย่างการใช้งาน              |
|--------------|---|--------------------------------|
| =            | แสดงค่าทั้งสองข้างเท่ากัน   | $X = 5$                        |
| +=           | ค่าฝั่งซ้ายมีค่าเท่ากับตัวมันเองแล้วบวกเพิ่มด้วยฝั่งขวา   | $X += 3, X = X + 3$            |
| -=           | ค่าฝั่งซ้ายมีค่าเท่ากับตัวมันเองแล้วลบเพิ่มด้วยฝั่งขวา  | $X -= 3, X = X - 3$            |
| *=           | ค่าฝั่งซ้ายมีค่าเท่ากับตัวมันเองแล้วคูณเพิ่มด้วยฝั่งขวา   | $X \times = 3, X = X \times 3$ |
| /=           | ค่าฝั่งซ้ายมีค่าเท่ากับตัวมันเองแล้วหารเพิ่มด้วยฝั่งขวา   | $X \div = 3, X = X \div 3$     |
| %=           | ค่าฝั่งซ้ายมีค่าเท่ากับตัวมันเองแล้วหารเพื่อหาเศษเพิ่มด้วยฝั่งขวา   | $X \% = 3, X = X \% 3$         |
| &=           | ค่าฝั่งซ้ายมีค่าเท่ากับตัวมันเองแล้วเปรียบเทียบในแต่ละคู่ของบิต ผลลัพธ์ที่ได้จะเป็น 1 ถ้าหากบิตทั้งสองเป็น 1 จากนั้นให้เก็บค่าผลลัพธ์ของแต่ละบิตไว้ที่ฝั่งซ้าย<br>คำนวณได้ $1 \& 0 = 0, 0 \& 0 = 0, 0 \& 1 = 0$ และ $0 \& 1 = 0$    | $X \& = 3$                     |
| =            | ค่าฝั่งซ้ายมีค่าเท่ากับตัวมันเองแล้วเปรียบเทียบในแต่ละคู่ของบิต ผลลัพธ์ที่ได้จะเป็น 1 ถ้าหากบิตใดบิตหนึ่งเป็น 1 จากนั้นให้เก็บค่าผลลัพธ์ของแต่ละบิตไว้ที่ฝั่งซ้าย<br>คำนวณได้ $1 \& 0 = 0, 0 \& 0 = 0, 0 \& 1 = 0$ และ $0 \& 1 = 0$ | $X  = 3$                       |

3.13. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อเปรียบเทียบดังต่อไปนี้

| ตัวดำเนินการ | ความหมาย                                      | ตัวอย่างการใช้งาน |
|--------------|---|-------------------|
| ==           | เปรียบเทียบค่าทั้งสองว่ามีค่าเท่ากัน          | $X == Y$          |
| >            | เปรียบเทียบค่าทางซ้ายมากกว่าทางขวา            | $X > Y$           |
| >=           | เปรียบเทียบค่าทางซ้ายมากกว่าหรือเท่ากับทางขวา | $X \geq Y$        |
| <            | เปรียบเทียบค่าทางขวามากกว่าทางซ้าย            | $X < Y$           |
| <=           | เปรียบเทียบค่าทางขวามากกว่าหรือเท่ากับทางซ้าย | $X \leq Y$        |
| !=           | เปรียบเทียบค่าทั้งสองว่ามีค่าไม่เท่ากัน       | $X != Y$          |

3.14. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการตรรกศาสตร์ดังต่อไปนี้

| ตัวดำเนินการ | ความหมาย   | ตัวอย่างการใช้งาน       |
|--------------|--|-------------------------|
| &&           | ถ้าค่าทั้งสองเป็นจริง ผลลัพธ์จะเป็นจริง  | $X < 5 \ \&\& \ Y < 10$ |
|              | ถ้าค่าใดค่าหนึ่งเป็นจริง ผลลัพธ์จะเป็นจริง                                     | $X > 5 \    \ Y < 2$    |
| !            | ถ้าค่าใดเป็นจริง จะเปลี่ยนให้เป็นเท็จ<br>ถ้าค่าใดเป็นเท็จ จะเปลี่ยนให้เป็นจริง | $!(X > 5 \    \ Y < 2)$ |

3.15. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if พร้อมยกตัวอย่างประกอบ

```

if (condition) {
    statement1
}
if (a > 0) {
    result = 'positive';
}

```

3.16. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if else พร้อมยกตัวอย่างประกอบ

```

if (condition) {
    statement1
}else
    statement2
if (hour < 18) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}

```

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

```
if (condition1) {  
    statement1  
} else if (condition2) {  
    Statement2  
} else {  
    Statement3  
}  
  
if (time < 10) {  
    greeting = "Good morning";  
} else if (time < 20) {  
    greeting = "Good day";  
} else {  
    greeting = "Good evening";  
}
```

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

```
variable = (condition) ? expressionTrue : expressionFalse;  
  
String message = isLogin ? "Welcome" : "Please login";
```

3.19. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง switch พร้อมยกตัวอย่างประกอบ

```
switch(expression) {  
    case x:  
        break;  
    case y:  
        break;  
    default:  
        break;  
}  
  
int day = 4;  
  
switch (day) {  
    case 1:  
        System.out.println("Monday");  
        break;  
    case 2:  
        System.out.println("Tuesday");  
        break;  
    case 3:  
        System.out.println("Wednesday");  
        break;  
}
```

```
}
```

3.20. การวนรอบ – จงเขียน Syntax การใช้งานคำสั่ง for พร้อมยกตัวอย่างประกอบ

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}  
  
for (int i = 0; i < 5; i++) {  
    System.out.println(i);  
}
```

3.21. การวนรอบ – จงเขียน Syntax การใช้งานคำสั่ง while พร้อมยกตัวอย่างประกอบ

```
while (condition) {  
    // code block to be executed  
}  
  
int i = 0;  
while (i < 5) {  
    System.out.println(i);  
    i++;  
}
```

3.22. การวนรอบ – จงเขียน Syntax การใช้งานคำสั่ง do while พร้อมยกตัวอย่างประกอบ

```
do {  
    // code block to be executed  
}  
  
while (condition);  
  
int i = 0;  
do {  
    System.out.println(i);  
    i++;  
}  
  
while (i < 5);
```

3.23. อธิบายการทำงานของคำสั่ง break พร้อมยกตัวอย่างประกอบ

- คำสั่ง break ใช้เพื่อออกจาก Loop

```
for (int i = 0; i < 10; i++) {  
    if (i == 4) {  
        break;  
    }  
    System.out.println(i);  
}
```

3.24. อธิบายการทำงานของคำสั่ง continue พร้อมยกตัวอย่างประกอบ

- ใช้สำหรับการข้ามการทำงาน (Skips) ที่ต้องการ

```
int i = 0;
while (i < 10) {
    if (i == 4) {
        i++;
        continue;
    }
    System.out.println(i);
    i++;
}
```

3.25. อธิบายการทำงานของคำสั่ง Enum

- Enum (Enumerated type) อินัม คือ ข้อมูลบางชนิดที่มีค่าเป็นไปได้เพียงไม่กี่ค่า เช่น ข้อมูล เพศ, สี, ระดับคะแนน หรือ ข้อมูลเดือนต่าง ๆ ในปัจจุบัน
- Enum ใช้สำหรับจัดเก็บค่าที่มีความเป็นไปได้ไม่กี่ค่า และมีความแน่นอนของค่าข้อมูล

#### 4. ลำดับขั้นการปฏิบัติการ

##### 4.1. จงแก้โจทย์ปัญหาดังต่อไปนี้

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>-----<br>1, 2, 3, 4, 5<br>6, 7, 8 | Please enter your value : 21<br><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(( )) --&gt; Input[/input num/]     Input --&gt; LoopStart(( ))     LoopStart --&gt; Cond1{i &lt;= num ?}     Cond1 -- No --&gt; End(( ))     Cond1 -- Yes --&gt; Cond2{i % 5 == 0 ?}     Cond2 -- Yes --&gt; Print1[print ค่า i]     Print1 --&gt; LoopStart     Cond2 -- No --&gt; Print2[print ค่า i ตามด้วย ,]     Print2 --&gt; Inc[i++]     Inc --&gt; LoopStart     Cond1 --&gt; Print3[print ค่า i]     Print3 --&gt; End </pre> | <pre> 1 package myfirstpackage; 2 3 import java.util.Scanner; // Import the Scanner class 4 5 public class Lab02 { 6 7     public static void main(String[] args) { 8 9         Scanner myNum = new Scanner(System.in); 10        int num; 11 12        System.out.print("Please enter your value : "); 13        num = myNum.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 } 30 </pre> |

#### 4.2. จงแก้โจทย์ปัญหาดังต่อไปนี้

4.2.1. จงเขียนโปรแกรมเพื่อแสดงจำนวนเฉพาะที่อยู่ระหว่าง เลข 2 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้

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



#### 4.2.2. ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)

| ผังงาน   | โค้ดโปรแกรม  |
|--|--|
| <pre> graph TD     Start(( )) --&gt; Input[/รับค่าตัวเลข/]     Input --&gt; Connector1(( ))     Connector1 --&gt; Divisible{หารลงตัว}     Divisible -- No --&gt; Connector1     Divisible -- yes --&gt; Count[/นับค่าเก็บไว้ว่าหารลงตัวเท่าไร/]     Count --&gt; Greater2{ค่าที่เก็บไว้ว่า&lt;br/&gt;หารลงตัวเกิน 2 ?}     Greater2 -- No --&gt; Output1[/จำนวนเฉพาะ/]     Greater2 -- yes --&gt; Output2[/ไม่ใช่จำนวนเฉพาะ/]     Output1 --&gt; Connector2(( ))     Output2 --&gt; Connector2     Connector2 --&gt; End(( ))         </pre> | <pre> 1 package OOPPackage; 2 3 import java.util.Scanner; // Import the Scanner class 4 5 public class week2Lab2 { 6     static void ModNum(int i) { 7         int count = 0; 8         for (int j = 2; j &lt;= i; j++) { 9             if (i % j == 0) { 10                 count++; 11             } 12         } 13         CheckPrime(count,i); 14     } 15     static void CheckPrime(int count,int i) { 16         if( count == 1 ) { 17             System.out.println("Prime Number"); 18         } 19         else { 20             System.out.print("Not Prime Number Divide by "); 21             ShowDivide(count,i); 22         } 23     } 24     static void ShowDivide(int count,int i) { 25         for (int j = 2; j &lt;= i; j++) { 26             if( i % j == 0 ) { 27                 System.out.print(j); 28                 count--; 29                 if( count == 1 ) { 30                     System.out.println(); 31                     break; 32                 } 33                 System.out.print(","); 34             } 35         } 36     } 37     public static void main(String[] args) { 38         Scanner myObj = new Scanner(System.in); // Create a Scanner object 39 40         System.out.print("Please enter your value : "); 41         int num = myObj.nextInt(); // Read user input 42         System.out.println("-----"); 43 44         for (int i = 2; i &lt;= num; i++) { 45             System.out.print(i); 46             System.out.print(" "); 47             ModNum(i); 48         } 49     }         </pre> |

## 5. สรุปผลการปฏิบัติการ

ได้รู้และเข้าใจคำสั่งพื้นฐานในภาษาจาวาได้มากขึ้น และการแก้โจทย์ปัญหาจะช่วยพัฒนาในการออกแบบ และพื้นฐานในการใช้งานโปรแกรมมากขึ้น

## 6. คำถามท้ายการทดลอง

6.1. คำสั่งต่างๆ ระหว่างภาษา C และภาษา Java มีความเหมือนหรือแตกต่างกันอย่างไร ?

- โครงสร้างคำสั่งที่ต่างกัน แต่มีความคล้ายกันอยู่
- มี Syntax ที่คล้ายกันแต่ใช้ชื่อเรียกไม่เหมือนกัน

6.2. จงระบุข้อควรระวังในการใช้คำสั่ง Switch Case

- ต้องมีคำสั่ง Break ในทุกเคส

6.3. การเลือกใช้คำสั่ง For เหมาะกับสถานการณ์ใดมากที่สุด

- เมื่อเราสามารถรู้การทำงานซ้ำที่แน่นอนในการทำงาน

6.4. คำสั่ง Casting ควรถูกนำมาใช้งานในสถานการณ์ใดที่สุด

- เมื่อ ตัวแปรที่ใช้เก็บข้อมูลมีขนาดที่เกินความจำเป็น หรือ มีความจำเป็นที่ต้องเพิ่มขนาดที่ใช้เก็บข้อมูล