

## ใบงานการทดลองที่ 2

### เรื่อง ภาษาจาวากับการโปรแกรมเชิงวัตถุ

#### 1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการโปรแกรมเชิงวัตถุด้วยคำสั่งพื้นฐานในภาษาจาวา
- 1.2. รู้และเข้าใจเพื่อศึกษาและทดลองสร้างโปรแกรมเชิงวัตถุโดยใช้ภาษาการโปรแกรมเชิงวัตถุใหม่ๆ

#### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

#### 3. ทฤษฎีการทดลอง

- 3.1. คำสั่งที่ใช้ในการแสดงผลบนแถบ Console คือคำสั่งใด

คำสั่ง `Run`

- 3.2. คำสั่งการคอมเมนต์แบบบรรทัดเดียว และแบบหลายบรรทัด คืออะไร ?

คำสั่ง `//` สำหรับบรรทัดเดียว

คำสั่ง `/* */` สำหรับหลายบรรทัด

- 3.3. ประเภทข้อมูลแบบ Integer คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

Int คือ เลขจำนวนเต็มค่าใดก็ได้ ไม่ทศนิยม จำนวนเต็มบวก จำนวนเต็มลบ และจำนวนเต็ม 0 ซึ่งเลขจำนวนเต็มเหล่านี้สามารถนำไปคำนวณได้  
Ex. `int Tong = 5; System.out.print(Tong);`

- 3.4. ประเภทข้อมูลแบบ Double คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

Double คือ ตัวแปรที่เก็บเลขทศนิยมโดยจะเก็บในรูปแบบ `double x 10e` ให้เขียนในทศนิยม 4 หลัก (มีเครื่องหมาย) `1.7E-308` ถึง `1.7E+308`  
Ex. `Double Tong = 10.1595; System.out.print(Tong);`

- 3.5. ประเภทข้อมูลแบบ Float คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

Float คือ ตัวแปรที่เก็บเลขทศนิยมโดยจะเก็บในรูปแบบ `double x 10e` ให้เขียนในทศนิยม 4 หลัก (มีเครื่องหมาย) `3.4E-37` ถึง `3.4E+37`  
Ex. `Float Tong = 10.1595; System.out.print(Tong);`

- 3.6. ประเภทข้อมูลแบบ Bool คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

bool คือ ตัวแปรที่เก็บค่าใช่หรือไม่ใช่ (True) กับ หรือ False Ex. `boolean TheTong = True; System.out.print(TheTong);`

- 3.7. ประเภทข้อมูลแบบ Char คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

char คือ ตัวแปรที่เก็บอักขระ หรือ สัญลักษณ์ใด ๆ อย่างหนึ่ง เช่น ตัวอักษร 'a' หรือ '0'  
Ex. `String Rai = "0"; System.out.println(Rai);`

- 3.8. ประเภทข้อมูลแบบ String คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบเพื่อแสดงค่าภายในตัวแปร

String คือ ตัวแปรที่เก็บข้อความ หรือ ข้อความใด ๆ อย่างหนึ่ง เช่น ข้อความ "Hello" หรือ "0"  
Ex. `String God = "TheGodIsFatherOfAllGods"; System.out.println(God);`

- 3.9. Widening Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือ การแปลงชนิดข้อมูลที่มีขนาดเล็กไปเป็นชนิดข้อมูลขนาดใหญ่

- 3.10. Narrowing Casting คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือ การแปลงชนิดข้อมูลที่มีขนาดใหญ่ไปเป็นชนิดข้อมูลขนาดเล็ก

3.11. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการทางคณิตศาสตร์ดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
+	บวกกัน ระหว่าง 2 จำนวน	$x = 5 + 5$
-	ลบกัน ระหว่าง 2 จำนวน	$x = 5 - 4$
*	คูณกัน ระหว่าง 2 จำนวน	$x = 3 * 2$
/	หารกัน ระหว่าง 2 จำนวน	$x = 4 / 2$
++	เพิ่มขึ้น 1 จำนวนเดิม 1	$x = x + 1$
--	ลดลง 1 จำนวนเดิม 1	$x = x - 1$
%	หาเศษส่วนแล้วเอาแต่ตัวเลข	$x = 10 \% 2$

3.12. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อกำหนดค่าดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
=	ค่าทางขวา จะถูกใส่ไปทางซ้าย	$x = 10$
+=	นำค่าทางขวาไปบวกกับค่าเดิม แล้วเก็บผลลัพธ์ไว้	$x += 10$ ซึ่งเท่ากับ $x = x + 10$
-=	นำค่าทางขวาไปลบกับค่าเดิม แล้วเก็บผลลัพธ์ไว้	$x -= 10$ ซึ่งเท่ากับ $x = x - 10$
*=	นำค่าทางขวาไปคูณกับค่าเดิม แล้วเก็บผลลัพธ์ไว้	$x *= 10$ ซึ่งเท่ากับ $x = x * 10$
/=	นำค่าทางขวาไปหารกับค่าเดิม แล้วเก็บผลลัพธ์ไว้	$x /= 10$ ซึ่งเท่ากับ $x = x / 10$
%=	นำค่าทางขวาไปหารกับค่าเดิม แล้วเก็บเศษส่วนไว้	$x \% = 10$ ซึ่งเท่ากับ $x = x \% 10$
&=	นำค่าทางขวาไป AND กับค่าเดิม แล้วเก็บผลลัพธ์ไว้	$x \& = 10$ ซึ่งเท่ากับ $x = x \& 10$
=	นำค่าทางขวาไป OR กับค่าเดิม แล้วเก็บผลลัพธ์ไว้	$x  = 10$ ซึ่งเท่ากับ $x = x   10$

3.13. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการเพื่อเปรียบเทียบดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
==	ตรวจสอบว่าค่าทางซ้ายและขวา มีค่าเท่ากันหรือไม่	$2 == 2$
>	ตรวจสอบว่าค่าทางซ้ายมีค่ามากกว่าค่าทางขวามากกว่าหรือไม่	$5 > 1$
>=	ตรวจสอบว่าค่าทางซ้ายมีค่ามากกว่าหรือเท่ากับค่าทางขวามากกว่าหรือไม่	$5 >= 5$
<	ตรวจสอบว่าค่าทางซ้ายมีค่าน้อยกว่าค่าทางขวามากกว่าหรือไม่	$5 < 10$
<=	ตรวจสอบว่าค่าทางซ้ายมีค่าน้อยกว่าหรือเท่ากับค่าทางขวามากกว่าหรือไม่	$5 <= 5$
!=	ตรวจสอบว่าค่าทางซ้ายและขวามีค่าไม่เท่ากันหรือไม่	$5 != 10000$

3.14. จงอธิบายความหมายและยกตัวอย่างการใช้งานตัวดำเนินการตรรกศาสตร์ดังต่อไปนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่างการใช้งาน
&&	จะกลับค่าเป็น true เมื่อทั้ง 2 ค่าเป็น true	$(a \& \& b)$ ถ้าทั้ง 2 เป็น false
	จะกลับค่าเป็น false เมื่อทั้ง 2 ค่าเป็น false	$(a    b)$ ถ้าทั้ง 2 เป็น true
!	นำค่ากลับค่าเป็น true ถ้าเป็น false และกลับค่าเป็น false ถ้าเป็น true	$!a$ ถ้า a เป็น false

3.15. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if พร้อมยกตัวอย่างประกอบ

Syntax: `if (condition) { คำสั่ง ; }`

Ex. `if (10 > 1) {  
    System.out.println("Songkai is the father of All God");  
}`

3.16. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง if else พร้อมยกตัวอย่างประกอบ

Syntax: `if (condition) { คำสั่ง ; } else { คำสั่ง ; }`

Ex. `if (10 > 1) { System.out.println("Songkai is the Original God");  
    else { System.out.println("Songkai is the Devil God"); }`

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

Syntax: `if (condition1) { คำสั่ง ; } else if (condition2) { คำสั่ง ; } else { คำสั่ง ; }`

Ex. `if (10 > 1) { System.out.println("Songkai is the Original God");  
    else if { System.out.println("Songkai is the Devil God");  
    else { System.out.println("Songkai is the Best human in the Universe"); }`

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

Syntax: `Variable = (condition) ? คำสั่ง ; : คำสั่ง ;`

Ex. `int time = 20  
String result = (time < 12) ? "Good day" : "Good evening";`

3.19. ประโยคเงื่อนไข - จงเขียน Syntax การใช้งานคำสั่ง switch พร้อมยกตัวอย่างประกอบ

```
Syntax: switch(expression) {  
    case x: //do this break;  
    case y: //do that break;  
    default: //do this }  
Ex: int Jong = 1;  
switch (Jong) { case 1: System.out.println("God Jong"); break;  
               case 2: System.out.println("deVil Jong"); break; }
```

3.20. การวนรอบ - จงเขียน Syntax การใช้งานคำสั่ง for พร้อมยกตัวอย่างประกอบ

```
Syntax: for (statement1; statement2; statements) { //do this }  
Ex. for (int i=0; i<1000000; i++) {  
    System.out.println(" I am the God Jong Rai");  
}
```

3.21. การวนรอบ - จงเขียน Syntax การใช้งานคำสั่ง while พร้อมยกตัวอย่างประกอบ

```
Syntax: while (condition) { //do this }  
Ex. int i=0 while (i<5) {  
    System.out.println(" I am the God Jong Rai");  
    i++;  
}
```

3.22. การวนรอบ - จงเขียน Syntax การใช้งานคำสั่ง do while พร้อมยกตัวอย่างประกอบ

```
Syntax: do { //do this } while (condition);  
Ex. i=0; do { System.out.println(" I am the God Jong Rai");  
            i++;  
        } while (i<5);
```

3.23. อธิบายการทำงานของคำสั่ง break พร้อมยกตัวอย่างประกอบ

คำสั่งนี้ใช้ขี้นอกจากการวนรอบตามเงื่อนไขที่กำหนด

3.24. อธิบายการทำงานของคำสั่ง continue พร้อมยกตัวอย่างประกอบ

คำสั่งนี้ใช้ขี้นอกจากการวนรอบ (skips) ตามเงื่อนไขที่กำหนดไว้ โดยจะไม่ดำเนินการ Break  
และ continue หรือ ไม่ดำเนินการตาม loop แต่ไปเริ่มวนรอบครั้งต่อไปทันที (แต่ถ้า loop จบลง  
จนจบแล้ว)

3.25. อธิบายการทำงานของคำสั่ง Enum

enum เป็น keyword กลุ่มของค่าที่เป็น constant ซึ่งไม่สามารถเปลี่ยนค่าได้ โดยที่ค่าที่กำหนด  
จะเป็นค่าที่คงที่ในโปรแกรม

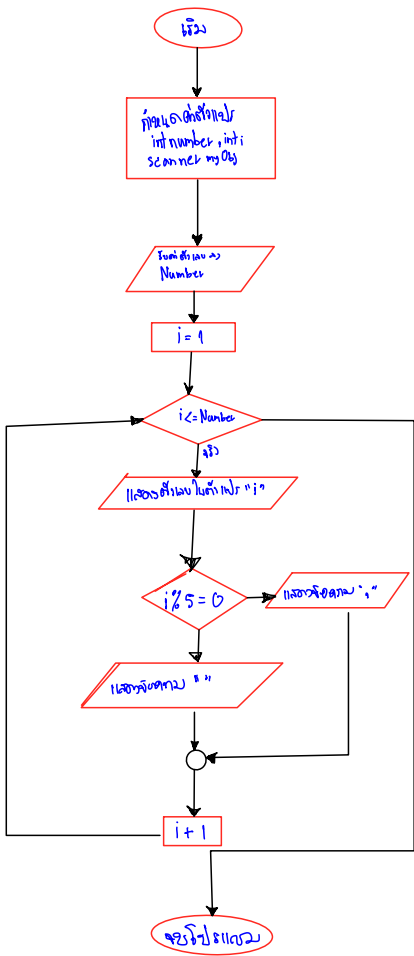
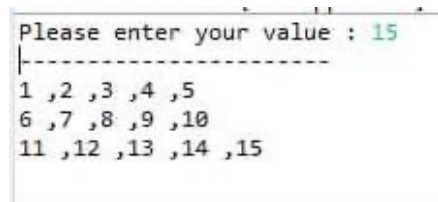
#### 4. ลำดับขั้นการปฏิบัติการ

4.1. จงแก้โจทย์ปัญหาดังต่อไปนี้

4.1.1. จงเขียนโปรแกรมเพื่อแสดงเลข 1 จนถึง N ; โดยที่ N คือค่าที่รับมาจากผู้ใช้ และแสดงคำตอบบรรทัดละ 5 ตัวเลขเท่านั้น

Test case 1	Test case 2
Please enter your value : 8 ----- 1, 2, 3, 4, 5 6, 7, 8	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

#### 4.1.2. ผังงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผังงาน)

ผังงาน	โค้ดโปรแกรม
 <pre> graph TD     Start([เริ่ม]) --&gt; Init[กำหนดค่าตัวแปร int number, int i; Scanner myObj]     Init --&gt; Input[/รับค่าตัวเลข Number/]     Input --&gt; i1[i = 1]     i1 --&gt; Cond1{i &lt;= Number}     Cond1 -- ใช่ --&gt; Output1[/แสดงตัวเลขในค่า i/]     Output1 --&gt; Cond2{i % 5 == 0}     Cond2 -- ใช่ --&gt; Output2[/แสดงข้อความ : '/]     Output2 --&gt; Output3[/แสดงข้อความ '/]     Output3 --&gt; iplus1[i + 1]     iplus1 --&gt; Cond1     Cond1 -- ไม่ใช่ --&gt; iplus1     Cond2 -- ไม่ใช่ --&gt; iplus1     iplus1 --&gt; End([จบโปรแกรม])   </pre>	<pre> Lab201.java x Lab202.java Test.java 1 import java.util.Scanner; // Import the Scanner class 2 3 4 public class Lab201 { 5     public static void main(String[] args) { 6         int Number ; 7 8         Scanner myObj = new Scanner(System.in); // Create a Scanner 9         System.out.print("Please enter your value : "); 10 11         Number = myObj.nextInt(); // Read user input 12         System.out.println("-----"); 13 14 15         for(int i=1; i&lt;=Number; i++) { 16             System.out.print(i); 17 18             if(i%5==0) { 19                 System.out.println(""); 20             } else { 21                 System.out.print(" ,"); 22             } 23         } 24     } 25 } 26 27 28   </pre> 

#### 4.2. จงแก้ไขข้อผิดพลาดดังต่อไปนี้

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

Test case 1	Test case 2
Please enter your value : 8 ----- 2 = Prime Number 3 = Prime Number 4 = Not Prime : Divide by 2 5 = Prime Number 6 = Not Prime : Divide by 2, 3 7 = Prime Number	Please enter your value : 16 ----- 2 = Prime Number 3 = Prime Number 4 = Not Prime : Divide by 2 5 = Prime Number 6 = Not Prime : Divide by 2, 3 7 = Prime Number

8 = Not Prime : Divide by 2, 4

8 = Not Prime : Divide by 2, 4

9 = Not Prime : Divide by 3

10 = Not Prime : Divide by 2, 5

11 = Prime Number

12 = Not Prime : Divide by 2, 3, 4, 6

13 = Prime Number

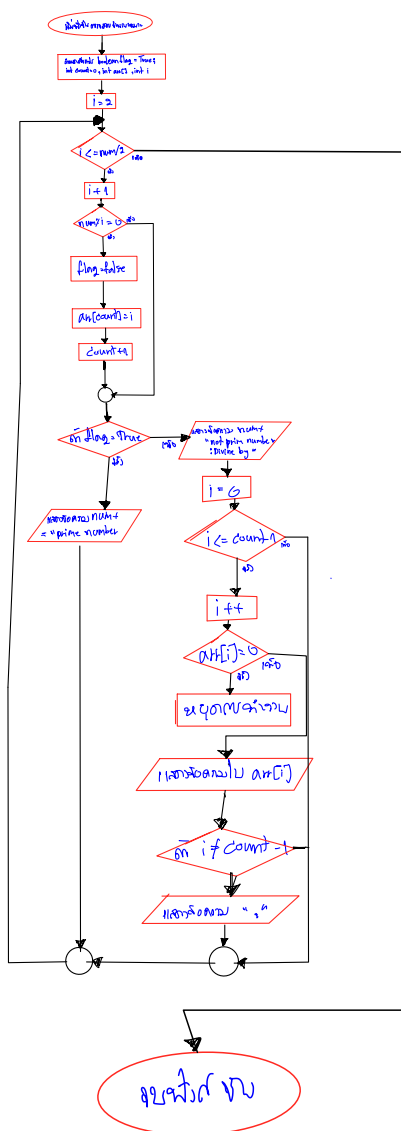
14 = Not Prime : Divide by 2, 7

15 = Not Prime : Divide by 2, 3, 5

16 = Not Prime : Divide by 2, 4, 8

#### 4.2.2. ผลงานแสดงกระบวนการทำงานและโค้ดโปรแกรม(ที่ตรงตามผลงาน)

ผลงาน



โค้ดโปรแกรม

```
1 Lab201.java 2 Lab202.java 3 Test.java
1 import java.util.Scanner;
2
3 public class Lab202 {
4
5     // ตรวจสอบจำนวน
6     public static void check_prime_number(int num) {
7         boolean flag = true;
8         int count = 0;
9         int arr[] = new int[num];
10        // ตรวจสอบจำนวน
11        for (int i = 2; i <= num/2; ++i) {
12            if (num % i == 0) {
13                flag = false;
14                arr[count] = i;
15                count++;
16            }
17        }
18        //
19        if (flag) {
20            System.out.println(num + " = prime number");
21        }
22        else {
23            System.out.print(num + " = not prime number: Divide by ");
24            // ตรวจสอบจำนวน
25            for (int i = 0; i < count-1; ++i) {
26                if (arr[i] != 0) {
27                    break;
28                }
29                System.out.print(arr[i]);
30                // ตรวจสอบจำนวน
31                if (i < count-1) {
32                    System.out.print(", ");
33                }
34            }
35            System.out.print("\n");
36        }
37    }
38    // End check prime number
```

```
40
41
42 public static void lab2_2() {
43     Scanner keyboard = new Scanner(System.in);
44     System.out.print("Please enter your value : ");
45     int n = keyboard.nextInt();
46     for (int i = 2; i <= n; ++i) {
47         check_prime_number(i);
48     }
49     // End lab2_2
50
51 public static void main(String[] args) {
52     lab2_2();
53 }
54 }
```

```
Please enter your value : 15
2 = prime number
3 = prime number
4 = not prime number: Divide by 2
5 = prime number
6 = not prime number: Divide by 2,3
7 = prime number
8 = not prime number: Divide by 2,4
9 = not prime number: Divide by 3
10 = not prime number: Divide by 2,5
11 = prime number
12 = not prime number: Divide by 2,3,4,6
13 = prime number
14 = not prime number: Divide by 2,7
15 = not prime number: Divide by 3,5
```

--	--

## 5. สรุปผลการปฏิบัติการ

การเขียนคำสั่ง เริ่มในทันที เช่น if รหัส และ คำสั่ง loop ที่ มี ส่วนต่อท้าย  
มาไว้แล้ว เช่น while ทิ้งไว้เพื่อใช้กับ ได้หลายรูปแบบ

## 6. คำถามท้ายการทดลอง

6.1. คำสั่งต่างๆ ระหว่างภาษา C และภาษา Java มีความเหมือนหรือแตกต่างกันอย่างไร ?

แตกต่างกับภาษา C ภาษา Java มีคอมไพเลอร์ Procedure ไม่สนับสนุน Thread และ Overloading Function ปรุกับ pointer  
ใช้ malloc บัญชี และจัดการเป็นระบบ Garbage collector สำหรับ Java ไม่สนับสนุน ค่า Value และ reference  
สนับสนุน OOP คอมไพเลอร์ Java (JVM) ไม่สนับสนุน Thread และ Overloading ไม่ปรุกับ pointer ใช้คำสั่ง new ในการ  
บ่งชี้และจัดสรรหน่วยความจำ Garbage collector สามารถเก็บอัตโนมัติ สำหรับ C ต้องจัดการกับ Value

6.2. จงระบุข้อควรระวังในการใช้คำสั่ง Switch Case

ต้องระวัง เมื่อใช้คำสั่ง break เพราะถ้าตกมาไม่ถึงคำสั่ง break จะทำให้โปรแกรมทำงานไม่ถูกต้อง

6.3. การเลือกใช้คำสั่ง For เหมาะกับสถานการณ์ใดมากที่สุด

เหมาะสำหรับใช้กับ Loop ที่ชัดเจน

6.4. คำสั่ง Casting ควรถูกนำมาใช้งานในสถานการณ์ใดที่สุด

เหมาะ กับ สถานการณ์ ที่ เรา ต้อง ทดแปลง ค่า ชนิด ตัวแปรใด ตัวแปร หนึ่ง  
ไปเป็น ชนิด ตัวแปร หนึ่ง ใน Operator ใน กรณี ที่ ชนิด ตัวแปร ไม่ตรงกัน Operator ไม่ใช้ เพื่อแปลง ค่า ชนิด ตัวแปร  
ไม่ตรงกับเงื่อนไข