

ข้อที่ 1: ข้อสอบเขียนโปรแกรม

1. เขียนโปรแกรมหาผลรวมของตัวเลขจำนวนเต็มที่ได้รับจากคีย์บอร์ด (17 คะแนน)
 - เขียนโปรแกรมที่รับตัวเลขจากผู้ใช้ และคำนวณผลรวมของตัวเลขเหล่านั้น
 - แสดงผลลัพธ์ออกมาและแก้ไขข้อผิดพลาดในโปรแกรมให้ถูกต้องตามเงื่อนไข
2. แปลงรูปจาก for เป็น while หรือจาก while เป็น for (3 คะแนน)
 - ให้เขียนโค้ดโดยเปลี่ยนประเภทของลูประหว่าง for และ while เพื่อทดสอบความเข้าใจ
3. อธิบายการใช้งานคำสั่งพื้นฐานต่าง ๆ (15 คะแนน)
 - ให้คำอธิบายการใช้งานของคำสั่ง เช่น if/else, for, while, อาร์เรย์, เมธอด, การอ่าน/เขียนไฟล์, การรับค่าจากคีย์บอร์ด, การใช้คอมเมนต์, และหลักการตั้งชื่อตัวแปร
 -

ข้อที่ 1.1: เขียนโปรแกรมหาผลรวมของตัวเลขจำนวนเต็มที่ได้รับจากคีย์บอร์ด


ตัวอย่างนี้รับตัวเลขจากผู้ใช้ไปเรื่อย ๆ จนกว่าผู้ใช้จะป้อนค่า 0 เพื่อหยุดโปรแกรม และคำนวณผลรวมของตัวเลขที่ป้อนเข้ามา

```
java Copy code  
  
import java.util.Scanner;  
  
public class SumNumbers {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int total = 0;  
  
        while (true) {  
            System.out.print("ป้อนตัวเลข (พิมพ์ 0 เพื่อหยุด): ");  
            int number = scanner.nextInt();  
  
            if (number == 0) {  
                break;  
            }  
  
            total += number; // บวกค่าที่ป้อนเข้ามาเข้ากับผลรวม  
        }  
  
        System.out.println("ผลรวมของตัวเลขที่ป้อนคือ: " + total);  
    }  
}
```

ข้อที่ 1.2: แปลงรูปจาก while เป็น for

ในข้อนี้จะเปลี่ยนจากการใช้ while เป็น for เพื่อให้เข้าใจการทำงานของลูปในทั้งสองแบบ ตัวอย่างด้านล่างคือการใช้ for loop เพื่อรับค่าตัวเลขจากผู้ใช้เป็นจำนวน 5 ครั้ง

java

 Copy code

```
import java.util.Scanner;

public class SumNumbersForLoop {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int total = 0;

        for (int i = 0; i < 5; i++) { // รับค่าตัวเลข 5 ครั้ง
            System.out.print("ป้อนตัวเลข: ");
            int number = scanner.nextInt();
            total += number;
        }

        System.out.println("ผลรวมของตัวเลขที่ป้อนคือ: " + total);
    }
}
```

ข้อที่ 1.3: อธิบายการใช้งานคำสั่งพื้นฐานต่าง ๆ

ตัวอย่างการอธิบายคำสั่งพื้นฐานใน Java พร้อมตัวอย่างโค้ด

- **if/else:** ใช้ในการตรวจสอบเงื่อนไข หากเงื่อนไขเป็นจริงจะทำงานในส่วนของ if และหากไม่เป็นจริงจะทำงานในส่วน
ของ else

```
java Copy code  
  
int age = 20;  
if (age >= 18) {  
    System.out.println("คุณบรรลุนิติภาวะแล้ว");  
} else {  
    System.out.println("คุณยังไม่บรรลุนิติภาวะ");  
}
```

for: ใช้ในการวนลูปจำนวนครั้งที่รู้ล่วงหน้า สามารถใช้กับโครงสร้างข้อมูลที่วนลูปได้ เช่น array

```
java Copy code  
  
for (int i = 1; i <= 5; i++) {  
    System.out.println("รอบที่ " + i);  
}
```

while: ใช้ในการวนลูปตรงใดที่เงื่อนไขยังเป็นจริง เหมาะกับกรณีที่ไม่รู้จำนวนครั้งการวนลูปล่วงหน้า

```
java Copy code  
  
int count = 0;  
while (count < 5) {  
    System.out.println("รอบที่ " + (count + 1));  
    count++;  
}
```

Array: เป็นโครงสร้างข้อมูลที่ใช้เก็บค่าหลายค่าในตัวแปรเดียว โดยเข้าถึงค่าด้วยดัชนี

```
java Copy code  
  
int[] numbers = {1, 2, 3, 4, 5};  
System.out.println("ค่าใน array ตำแหน่งที่ 0: " + numbers[0]); // แสดงผล 1
```

ข้อที่ 2: ตัวอย่างข้อสอบย่อย (Short Answer)

1. เขียนโค้ดเพื่อหาค่าผลรวมของตัวเลข 1 ถึง 10:

```
java Copy code  
  
int total = 0;  
for (int i = 1; i <= 10; i++) {  
    total += i;  
}  
System.out.println("ผลรวมของ 1 ถึง 10 คือ: " + total);
```

เขียนโปรแกรมเพื่อพิมพ์ "Hello, World!":

```
java Copy code  
  
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

อธิบายเกี่ยวกับ `Array` ใน Java:

- `Array` เป็นโครงสร้างข้อมูลที่สามารถเก็บค่าหลายค่าในตัวแปรเดียว โดยเข้าถึงค่าแต่ละตำแหน่งด้วยดัชนี (index)

```
java Copy code  
  
int[] numbers = {1, 2, 3, 4, 5};  
System.out.println("ค่าใน array: " + numbers[0]); // แสดงผล 1
```

การใช้งาน Method เพื่อหาค่าผลรวมของตัวเลข:

```
java Copy code  
  
public class SumExample {  
    public static void main(String[] args) {  
        int result = sum(5, 10);  
        System.out.println("ผลรวมคือ: " + result);  
    }  
  
    public static int sum(int a, int b) {  
        return a + b;  
    }  
}
```



ให้คำอธิบายการใช้งานของคำสั่ง เช่น if/else, for, while, อาร์เรย์, เมธอด, การอ่าน/เขียนไฟล์, การรับค่าจากคีย์บอร์ด, การใช้คอมเมนต์, และหลักการตั้งชื่อตัวแปร

1. if/else

- คำสั่ง `if/else` ใช้ในการตรวจสอบเงื่อนไข หากเงื่อนไขใน `if` เป็นจริง (`true`) จะทำงานในส่วนของ `if` แต่ถ้าเงื่อนไขเป็นเท็จ (`false`) จะไปทำงานในส่วนของ `else`
- ตัวอย่าง:

```
java Copy code

int age = 20;
if (age >= 18) {
    System.out.println("คุณบรรลุนิติภาวะแล้ว");
} else {
    System.out.println("คุณยังไม่บรรลุนิติภาวะ");
}
```

2. for

- คำสั่ง `for` ใช้สำหรับการวนลูปในจำนวนครั้งที่รู้ล่วงหน้า โดยเราสามารถกำหนดค่าเริ่มต้น เงื่อนไขในการหยุด และการเปลี่ยนแปลงค่าในแต่ละรอบได้
- ตัวอย่าง:

```
java Copy code

for (int i = 0; i < 5; i++) {
    System.out.println("รอบที่ " + i);
}
```

3. while

- คำสั่ง `while` ใช้ในการวนลูปตราบใดที่เงื่อนไขยังเป็นจริง (`true`) และจะหยุดเมื่อเงื่อนไขเป็นเท็จ (`false`) โดยเหมาะกับกรณีที่ไมู้จำนวนครั้งที่การวนลูปล่วงหน้า
- ตัวอย่าง:

```
java Copy code

int count = 0;
while (count < 5) {
    System.out.println("รอบที่ " + count);
    count++;
}
```




4. Array (อาร์เรย์)

- **Array** เป็นโครงสร้างข้อมูลที่ใช้เก็บข้อมูลหลายค่าในตัวแปรเดียว โดยข้อมูลต่างๆ จะถูกเก็บตามลำดับ และสามารถเข้าถึงได้ด้วยดัชนี (index) ที่เริ่มจาก 0

- ตัวอย่าง:

java

 Copy code


```
int[] numbers = {1, 2, 3, 4, 5};
System.out.println("ค่าใน array ตำแหน่งที่ 0: " + numbers[0]);
```

5. Method (เมธอด)

- **Method** เป็นฟังก์ชันหรือกระบวนการที่ทำงานเฉพาะอย่างในโปรแกรม สามารถส่งค่าเข้ามาทำงานในเมธอดได้ และเมธอดสามารถส่งค่าผลลัพธ์กลับไปได้เช่นกัน

- ตัวอย่าง:

java

 Copy code

```
public class Example {
    public static void main(String[] args) {
        int result = add(5, 10);
        System.out.println("ผลรวมคือ: " + result);
    }

    public static int add(int a, int b) {
        return a + b;
    }
}
```

7. การรับค่าจากคีย์บอร์ด

- การรับค่าจากคีย์บอร์ดใน Java สามารถทำได้โดยใช้ `Scanner` เพื่ออ่านข้อมูลจากผู้ใช้ เช่น อ่านค่าจำนวนเต็ม หรือสตริง
- ตัวอย่าง:


```
java Copy code  
  
import java.util.Scanner;  
  
public class InputExample {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("ป้อนชื่อของคุณ: ");  
        String name = scanner.nextLine();  
        System.out.println("สวัสดี, " + name + "!");  
    }  
}
```

8. การใช้คอมเมนต์

- คอมเมนต์ใน Java ใช้สำหรับใส่คำอธิบายหรือหมายเหตุในโค้ดเพื่อให้ผู้อ่านเข้าใจได้ง่ายขึ้น และไม่ส่งผลกระทบต่อการทำงานของโปรแกรม
- มีคอมเมนต์ 2 แบบ:
 - คอมเมนต์บรรทัดเดียว ใช้ `//` นำหน้า

```
java Copy code  
  
// นี่คือนคอมเมนต์บรรทัดเดียว
```

- คอมเมนต์หลายบรรทัด ใช้ `/* ... */`

```
java Copy code  
  
/*  
    นี่คือนคอมเมนต์หลายบรรทัด  
    สามารถใช้สำหรับใส่คำอธิบายยาวๆ  
*/  
  

```

9. หลักการตั้งชื่อตัวแปร

- หลักการตั้งชื่อตัวแปรที่ดีทำให้โค้ดอ่านง่ายและเข้าใจได้ชัดเจน คำแนะนำในการตั้งชื่อตัวแปร:
 - ใช้คำที่สื่อถึงหน้าที่หรือความหมาย เช่น `totalAmount`, `isFinished`, `studentName`
 - ใช้ camelCase สำหรับตัวแปร เช่น `firstName`, `totalScore`
 - หลีกเลี่ยงการใช้ชื่อย่อหรือคำที่ไม่ชัดเจน

ตัวดำเนินการ หรือ เครื่องหมายดำเนินการ (operator)

ตัวดำเนินการอาจถูกแบ่งได้เป็นหลายประเภท ดังนี้-

1. ตัวดำเนินการตัวเลข (Numeric Operators)

ชื่อ	ความหมาย	ตัวอย่าง	ผลลัพธ์
+	การบวก	34 + 1	35
-	การลบ	34.0 - 0.1	33.9
*	การคูณ	300 * 30	9000
/	การหาร	1.0 / 2.0	0.5
%	การหารหาเศษ	20 % 3	2

มาดูตัวอย่างการคูณตัวเลขสองตัวแรกที่เราได้เขียนดูกัน การคูณของจำนวนเต็มสองจำนวน

```
1 public class MultiplyExample {
2     public static void main(String[] args) {
3         int a;
4         int b;
5         int c;
6         a = 5;
7         b = 58;
8         c = a*b; //integer number to keep the result of multiplication
9         System.out.println("5*58 = " + c);
10    }
11 }
```

ผลลัพธ์คือ:

5*58 = 290

ในความเป็นจริง คุณสามารถคูณจำนวนเต็มสองจำนวนอย่างชัดเจนโดยไม่ต้องกำหนดค่าให้กับตัวแปร และแสดงผลของการกระทำบนหน้าจอ หรือคุณจำนวนด้วยตัวแปร: ตัวอย่างที่ 2. การคูณ จำนวน

```
1 public class MultiplyExample {
2     public static void main(String[] args) {
3         int a;
4         a = 5;
5         System.out.println("7*7 = " + 7*7);
6         System.out.println("a*5 = " + a*5);
7     }
8 }
```

และนี่คือผลลัพธ์:

7*7 = 49 a*5 = 25


```

1 public class MultiplyExample {
2     public static void main(String[] args) {
3         int a;
4         a = 5;
5         System.out.println("7*7 = " + 7*7);
6         System.out.println("a*5 = " + a*5);
7     }
8 }

```

และนี่คือผลลัพธ์:

7*7 = 49 a*5 = 25

คุณยังสามารถคูณเลขเศษส่วนด้วยเลขเศษส่วนอื่นๆ หรือเลขเศษส่วนด้วยจำนวนเต็ม โปรดทราบว่าผลลัพธ์ของการคูณเศษส่วนด้วยจำนวนเต็มจะเป็นประเภทเศษส่วน ในการคูณแบบนี้ Java จะแปลงประเภทจำนวนเต็มดั้งเดิม เช่น int ให้กับประเภทของจำนวนเศษส่วนที่ใช้คุณ (เช่น สองเท่า) และผลลัพธ์ที่ได้จะเป็นสองเท่าด้วย

```

1 public class MultiplyExample2 {
2     public static void main(String[] args) {
3         double x = 15.7;
4         double y = 2.1;
5         int a = 3;
6         double z = x*y;
7         double b = a*x;
8         //if you try something like int s = a*x; your program won't run, it's a mistake.
9         System.out.println(x + "*" + y + " = " + z);
10        System.out.println(a + "*" + x + " = " + b);
11    }
12 }

```

การคูณเป็นการดำเนินการที่ง่ายมาก แต่ต้องใช้ด้วยความระมัดระวัง ตัวอย่างเช่น คุณสามารถกำหนดประเภทผลลัพธ์การคูณที่สั้นเกินไป และผลลัพธ์ก็อาจไม่พอดีกับตัวแปรนี้ สมมติว่าตัวแปรสั้นเท่ากับ 32767 นี่คือการขีดจำกัดบนสำหรับตัวแปรประเภทนี้ (ตัวเลข 32768 ไม่สามารถสั้นได้อีกต่อไป เนื่องจากไม่พอดีกับ 2 ไบต์ที่กำหนดให้กับประเภทข้อมูลนี้) ลองพิจารณาตัวอย่าง:

```

1 public class MultiplyExample3 {
2     public static void main(String[] args) {
3         short myShort1 = 32767;
4         short myShort2 = 2;
5         short myShort3 = myShort1*myShort2;
6     }
7 }

```

IDE สมัยใหม่ เช่น IDEA จะขีดเส้นใต้บรรทัดที่กำหนดตัวแปร `myShort3` เป็นสีแดง แต่ถ้าเราเรียกใช้โปรแกรม เราได้รับข้อความแสดงข้อผิดพลาดต่อไปนี้:

ข้อผิดพลาด: (5, 34) java: ประเภทที่เข้ากันไม่ได้: การแปลงแบบสูญเสียที่เป็นไปได้จาก int เป็น short

ดังนั้นเมื่อคุณเขียนโปรแกรม ให้ลองคิดว่าข้อมูลชนิดนั้นเพียงพอสำหรับคุณหรือไม่ ในกรณีของตัวอย่างข้างต้น int เหมาะสม คุณยังสามารถเขียนโปรแกรมเอนกประสงค์โดยผู้ใช้ป้อนตัวเลขสองตัว:

```
1 import java.util.Scanner;
2
3 public class MultiplyExample3 {
4
5     public static void main(String[] args) {
6
7         Scanner scanner = new Scanner(System.in);
8         System.out.print("Enter first number = ");
9         double myDouble1 = scanner.nextDouble();
10        System.out.print("Enter second number = ");
11        double myDouble2 = scanner.nextDouble();
12        scanner.close();
13
14        double result = myDouble1*myDouble2;
15
16        // Displaying the multiplication result
17        System.out.println(myDouble1 + "*" + myDouble2 + " = " + result);
18    }
19 }
```

นี่คือผลลัพธ์:

ใส่ตัวเลขแรก = 5 ใส่ตัวเลขที่สอง = 12 5.0*12.0 = 60.0