

1. อธิบายและยกตัวอย่าง รับค่าจาก command line มาพอเข้าใจ

C:\> java x 5 ใช้ args[0]

```
class x{
    public static void main(String args[]){
        System.out.println(args[0]);
    }
}
```

- เป็นการรับค่าจาก Argument จาก command line โดยจะเก็บไว้ใน Array ตัวที่ 0 หลังจากนั้นสามารถนำค่าที่ได้ไปประมวลผลต่อไป

2. อธิบายและยกตัวอย่าง รับค่าจาก แป้นพิมพ์ มาพอเข้าใจ

BufferedReader

```
import java.io.*;
class x {
    public static void main(String args[]) throws IOException{
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        int a = Integer.parseInt(stdin.readLine());
        System.out.println(a);
    }
}
```

- ก่อนอื่นเราต้อง import java.io.* เพื่อบอกให้โปรแกรมทราบว่าเราต้องการใช้คลาสนี้
- จากโค้ด เราจะเห็นว่าเราได้สร้างอ็อบเจกต์ขึ้นมาชื่อว่า stdin ซึ่งเป็นอ็อบเจกต์ที่สร้างขึ้นจากคลาส BufferedReader ซึ่งในอ็อบเจกต์นี้ เราก็ได้สร้างอ็อบเจกต์ขึ้นมาอีกอ็อบเจกต์หนึ่ง เป็นอ็อบเจกต์ที่สร้างขึ้นจากคลาส InputStreamReader ซึ่งอ็อบเจกต์นี้ได้รับค่า System.in ซึ่งหมายถึงแป้นพิมพ์นั่นเอง
- เมื่อได้อ็อบเจกต์มาแล้ว เราจะใช้ method readLine(); เพื่อรับค่าเป็นสตริง

3. อธิบายและยกตัวอย่าง การรับค่าเข้า method มาพอเข้าใจ

mycal(5,6) mycal(x,y)

```
class x{
    public static void main(String[] args){
        System.out.println(mycal(5,6));
    }
    private static int mycal(int a,int b){
        return a + b;
    }
}
```

- เป็นการสร้างและเรียก method ในคลาสเดียวกันด้วยการทำงานของมันเป็นคือ มี method ชื่อ mycal ที่รับค่า a และ b เป็น int มาบวกกันแล้วคืนผลลัพธ์ส่วน method หลักก็จะทำการเรียก method mycal แล้วส่งค่าไปคำนวณ

4. อธิบายและยกตัวอย่าง การส่งค่าคืนจาก method มาพอเข้าใจ

int a (int b) { return (b * 2); } System.out.print(a(5));

```
class x {
    static int a(int b) {
        return b * 2;
    }

    public static void main(String[] args) {
        System.out.println(a(5));
    }
}
```

- ทำการสร้าง method a และสร้างตัวแปร b เป็น int ใน method a ทำการ return ค่า ให้ b*2 และใน method main ทำการแสดงผลโดยเรียกใช้ method a แล้วใน ตัวแปร b ที่ค่าเท่ากับ 5

5. อธิบายและยกตัวอย่าง การเรียก method จากต่าง class แบบไม่ extends มาพอเข้าใจ

```
class aa {  
    public static String aaa() {  
        System.out.println("Warayut khanka");  
    }  
}  
  
class bb {  
    public static void main(String[] args) {  
        aa x = new aa();  
        x.aaa();  
    }  
}
```

- สร้าง class ที่ชื่อว่า aa และใน aa มี method ที่ชื่อว่า aaa() และให้ทำการแสดงผลลัพธ์ “Warayut khanka” และทำการสร้างอีก 1 class ที่ชื่อว่า bb ในclass นี้ ทำการสร้างตัวแปล x ให้มีค่าเท่ากับ class aa จึงทำให้ตัวแปล x สามารถเรียกใช้ method ในclass x ได้

6. อธิบายและยกตัวอย่าง การเรียก method จากต่าง class แบบ extends มาพอเข้าใจ

```
class aa {  
    public void aaa() {  
        System.out.println("Warayut khanka");  
    }  
}  
  
class bb extends aa {  
    public static void main(String[] args) {  
        bb x = new bb();  
        x.aaa();  
    }  
}
```

- สร้าง class ที่ชื่อว่า aa และใน aa มี method ที่ชื่อว่า aaa() และให้ทำการแสดงผลลัพธ์ “Warayut khanka” และทำการสร้างอีก 1 class ที่ชื่อว่า bb โดยclass bb extends มีจาก class aa ใน class นี้ ทำการสร้างตัวแปล x ให้มีค่าเท่ากับ class bb จึงทำให้ตัวแปล x สามารถเรียกใช้ method ในclass x ได้เพราะว่า class bb extends มาจาก class aa

7. อธิบายและยกตัวอย่าง for แรกรับค่า ส่วน for ที่สองประมวลผลมาพอเข้าใจ

```
import java.io.*;
class x {
    public static void main(String args[]) throws IOException{
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        int a = Integer.parseInt(stdin.readLine());
        for(int i=0; i<=a; i++){
            for(int j=1; j<=a; j++){
                System.out.println(j);
            }
        }
    }
}
```

- ทำการเรียกใช้ BufferedReader เพื่อรับค่าจากแป้นพิมพ์ โดยสร้างตัวแปร a เพื่อเก็บค่าที่รับเข้ามา หลังจากนั้น ตัวแปร a มาประมวลผล โดยใช้ for for แรกทำหน้าที่รับค่าเข้ามาโดยการทำงานหรือรอบของการทำงานจะขึ้นอยู่กับตัวแปร a เช่น ถ้า a เป็น 5 for ก็ทำงาน 5 รอบ หลังจากนั้น for ที่ 2 ให้ทำการประมวลผลโดยใช้ตัวแปร j มาแสดงผล โดยที่ จะทำการแสดงผลตั้งแต่เลข 1 ไปจนถึงค่า a

8. อธิบายและยกตัวอย่าง ตัวแปรแบบ instance และแบบ local variable มาพอเข้าใจ

- **ตัวแปรแบบ Instance** นั้นจะถูกประกาศเอาไว้ในคลาส แต่จะอยู่ภายนอกของ method, constructor และ block ต่างๆ ซึ่งตัวแปรแบบ Instance นั้นจะถูกสร้างขึ้นก็ต่อเมื่อมีการสร้าง Object ขึ้นมา การเรียกใช้ตัวแปรแบบ Instance นั้นสามารถเรียกใช้ภายใน method, constructor หรือใน block ได้
- **Local variables** เป็นตัวแปรที่ถูกประกาศใน method, constructors หรือ blocks โดยจะถูกสร้างขึ้นเมื่อ method, constructors หรือ block ถูกเรียกใช้ และจะถูกทำลายเมื่อสิ้นสุด method, constructors หรือ block โดยตัวแปรแบบ Local นั้นจะถูกมองเห็นเฉพาะขอบเขตที่ถูกประกาศเอาไว้เท่านั้น เช่น ถ้าประกาศตัวแปรเอาไว้ใน method ตัวแปรนั้นก็จะสามารถใช้งานได้ภายใน method นั้นเท่านั้น

9. อธิบายและยกตัวอย่าง การทำ casting มาพอเข้าใจ

```
class x {  
    public static void main(String[] args) {  
        char a = 'A';  
        int b = (int) a; // Manual casting: char to int  
  
        System.out.println(a);    // Outputs A  
        System.out.println(b);    // Outputs 65  
    }  
}
```

- สร้างตัวแปรที่ ชื่อว่า a มีชนิดเป็น char และมีค่าเท่ากับ A และสร้างตัวแปรที่ชื่อว่า b ตัวแปร b จะทำการ casting จาก char ไปเป็น int และหลังจากนั้นให้ทำการแสดงผลโดย A จะมีค่าคือ 65