

## 1. อธิบายและยกตัวอย่าง รับค่าจาก command line มาพอเข้าใจ

- เป็นการรับ Arguments จาก command line ใส่โดยจะเริ่มนับที่ช่องที่ 0 มายัดค่าเข้ากับตัวแปรเพื่อใช้ในการคำนวณต่อไป

```
public class Test {  
    public static void main(String[] args) {  
        double a, b, c;  
        a = Double.parseDouble(args[0]); //ทำการแปลงค่าจาก arg[0] ให้เป็น double  
        b = Double.parseDouble(args[1]); //ทำการแปลงค่าจาก arg[1] ให้เป็น double  
        c = Double.parseDouble(args[2]); //ทำการแปลงค่าจาก arg[2] ให้เป็น double  
        System.out.println(a+"\n"+b+"\n"+c);  
    }  
}
```

## 2. อธิบายและยกตัวอย่าง รับค่าจาก แป้นพิมพ์ มาพอเข้าใจ

- การรับค่าจากแป้นพิมพ์ข้อนี้จะใช้ BufferedReader โดยการทำงานของมันเป็นจะทำการสร้าง method ของ BufferedReader แล้วอ่านตัวอักษรที่เรากดในแป้นพิมพ์ผ่าน command line เพื่อมาเก็บค่าในตัวแปร i

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

### 3. อธิบายและยกตัวอย่าง การรับค่าเข้า method มาพอเข้าใจ

- เป็นการสร้างและเรียก method ในคลาสเดียวกันด้วยการทำงานของมันคือ มี method ชื่อ sum ที่รับค่า int 2 ค่ามาบวกกันแล้วคืนค่าผลลัพธ์ส่วน method หลักก็จะทำการเรียก method sum แล้วส่งค่าไปคำนวณ

```
public class Test {  
    public static void main(String args[]){  
        System.out.println("10 + 20 = " + sum(10, 20) );  
    private static int sum (int a, int b) {  
        return a + b;  
    }  
}  
}
```

### 4. อธิบายและยกตัวอย่าง การส่งค่าคืนจาก method มาพอเข้าใจ

- เป็นการสร้างและเรียก method ในคลาสเดียวกันด้วยการทำงานของมันคือ มี method ชื่อ sum ที่รับค่า int 2 ค่ามาบวกกันแล้วคืนค่าผลลัพธ์ส่วน method หลักก็จะทำการเรียก method sum แล้วส่งค่าไปคำนวณ

```
public class Test {  
    public static void main(String args[]){  
        System.out.println("10 + 20 = " + sum(10, 20) );  
    private static int sum (int a, int b) {  
        return a + b;  
    }  
}  
}
```

## 5. อธิบายและยกตัวอย่าง การเรียก method จากต่าง class แบบไม่ extends มาพอเข้าใจ

- การเรียก method จากต่าง class แบบไม่ extends นั้น class จะต้องเป็น public ก่อนจึงจะสามารถถึง class นั้นมาใช้ได้โดยที่ myObj ก็จะเป็นการสืบทอด class MyClass จึงสามารถถึง method มาใช้ได้ด้วย

```
public class MyClass {
    public static String x(){
        System.out.println( " Hello World ")
    }
}
class OtherClass {
    public static void main(String[] args) {
        MyClass myObj = new MyClass();
        myObj.x();
    }
}
```

## 6. อธิบายและยกตัวอย่าง การเรียก method จากต่าง class แบบ extends มาพอเข้าใจ

- เป็นการเรียก method จากต่าง class แบบ extends คือ สามารถใช้ method ทุกอย่างของ class ที่ถูก extend มา

```
public class MyClass {
    public static String x(){
        System.out.println( " Hello World ")
    }
}
class OtherClass extends MyClass {
    public static void main(String[] args) {x();} }
}
```

## 7. อธิบายและยกตัวอย่าง for แรกรับค่า ส่วน for ที่สองประมวลผลมาพอเข้าใจ

- จะเป็นการรับค่าจากแป้นพิมพ์โดยใช้ BufferedReader รับค่ามาใส่ในตัวแปรอาร์เรย์ที่มีความจุ 5 โดยลูป for แรกจะใช้รับค่า และ ลูปที่ 2 จะใช้คำนวณค่าที่ได้จากลูปแรก

```
import java.io.*;

class x {

    public static void main(String args[]) throws IOException {

        BufferedReader stdin = new BufferedReader(new InputStreamReader(
System.in));

        int i[] = {0,0,0,0,0};
        String buf1 ;
        for(int j = 0;j<5;j++){
            buf1 = stdin.readLine();
            i[j] = Integer.parseInt(buf1);
        }
        for(int j = 0;j<5;j++){
            int sum = 0;
            sum = i[j] + i[j];
            System.out.println(sum);
        }
    }
}
```

## 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
- ตัวแปรแบบ **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

- casting ใช้แปลงชนิดข้อมูล จากชนิดหนึ่งเป็นอีกชนิดได้ โดยไม่สนใจขนาด

ตัวอย่าง การแปลงค่าระหว่าง int และ String

```
int num1;  
String x="10";  
num1=Integer.parseInt(x) // num1มีค่าเท่ากับ 10
```