

งานมอบหมายที่ 8 - 2%

1. อธิบายและยกตัวอย่าง รับค่าจาก command line มาพอเข้าใจ

C:\> java x 5 ใช้ args[0]

```
public class m1 {  
    public static void main(String args[]) {  
        int s1 = Integer.parseInt(args[0]);  
        int s2 = Integer.parseInt(args[1]);  
        System.out.println("Plus = " + (s1 + s2));  
    }  
}
```

ยกตัวอย่าง รับค่าจาก command line โปรแกรมนี้ มีการรับค่าจาก command line จำนวน 2 ค่า คือ s1 และ s2

แล้วนำมาคำนวณหาผลบวก และแสดงผล

2. อธิบายและยกตัวอย่าง รับค่าจาก แป้นพิมพ์ มาพอเข้าใจ

BufferedReader

```
import java.io.*;  
public class m2 {  
    public static void main(String args[]) throws IOException{  
        BufferedReader str = new BufferedReader(new InputStreamReader(System.in));  
        System.out.print("Insert Name : ");  
        String s = str.readLine(); //อ่านออกมาเป็นสตริง  
        System.out.println("Yourname " + s );  
    }  
}
```

จะเห็นว่าเราได้สร้างอ็อบเจกต์ขึ้นมาชื่อว่า str ซึ่งเป็นอ็อบเจกต์ที่สร้างขึ้นจากคลาส BufferedReader ซึ่งในอ็อบเจกต์นี้ เราก็ได้สร้างอ็อบเจกต์ขึ้นมาอีกอ็อบเจกต์หนึ่ง เป็นอ็อบเจกต์ที่สร้างขึ้นจากคลาส InputStreamReader ซึ่งอ็อบเจกต์นี้ได้รับค่า System.in ซึ่งหมายถึงแป้นพิมพ์นั่นเอง

3. อธิบายและยกตัวอย่าง การรับค่าเข้า method มาพอเข้าใจ

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

```
public class m3 {  
    public void mim(int a,int b)  
    { System.out.println(a+b); } //end method mim()  
    public static void main( String args[] ) {  
        m3 a=new m3();  
        a.mim(10,1); // call method mim()  
    } //end main()  
} //end class
```

Method ที่มีการรับค่าหรือมีการนำค่าเข้าสู่ภายในเมธอด โดยผ่านทาง parameter

ในการเขียนโปรแกรมสิ่งที่จำเป็นต้องพิจารณาคือ Parameter และ Argument โดยที่ ค่าที่ class หรือวัตถุนั้นเก็บเพื่อส่งต่อให้กับ Method นั้นคือ Argument สิ่งที่ Method นั้นเก็บจะเรียกว่า Parameter

จากโปรแกรมจะเห็นได้ว่า Argument คือ 10 และ 1 ส่วน Parameter คือ a และ b

4. อธิบายและยกตัวอย่าง การส่งค่าคืนจาก method มาพอเข้าใจ

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

```
public class m4 {  
  
    public static void main(String[] args) {  
  
        System.out.println("Display Multiply");  
        System.out.println(" "+multiply());  
    } //end main()  
  
    public static String multiply() {  
  
        int b=0;  
        String output= " ";  
        for (int i=2;i<=3 ;i++ ) {  
  
            for(int j=1;j<=12;j++) {  
  
                b = i*j;  
                output += b+ " ";  
            } //end inside for  
            output += " \n ";  
        }  
    }  
}
```

```

    } //end outside for
    return output;
} //end method multiply()
} //end class

```

Method ที่มีการส่งหรือคืนค่ากลับ เป็น method ที่ไม่มีตัวแปร parameter แต่เมื่อสิ้นสุดการทำงานของ method จะทำการ return กลับไปยัง method เมื่อถูกเรียกใช้งาน ข้างหน้าชื่อเมธอดจะไม่มีคำว่า void แต่ชนิดของ dataType ที่ต้องการคืนค่ากลับ และภายในเมธอดจะมีคำว่า return

ตัวอย่างโปรแกรม : การ return ค่าตัวแปรเพื่อแสดงผลสูตรคูณแม่ 2 และ 3

5. อธิบายและยกตัวอย่าง การเรียก method จากต่าง class แบบไม่ extends มาพอเข้าใจ

```

class Demo {
    static void foo() {
        try {
            System.out.println("Inside foo");
            throw new RuntimeException("demo error");
        }
        finally {
            System.out.println("foo's finally");
        }
    }

    static void bar() {
        try {
            System.out.println("Inside bar");
            return;
        }
        finally {
            System.out.println("bar's finally");
        }
    }

    public static void main(String[] args) {
        try {
            foo();
        }
        catch (Exception e) {
            System.out.println("Exception caught");
        }
        finally {
            bar();
        }
    }
}

```

```
}  
}
```

finally keyword (ตัวเล็กเท่านั้น) เป็นคำสงวนใจภาษา Java ซึ่งเราสามารถใช้ได้กับ บล็อก try/catch เพื่อการันตีว่าส่วนของโค้ดนั้นจะถูก execute อย่างแน่นอน แม้จะมีการ thrown exception ก็ตาม ซึ่งส่วนของ finally จะ execute หลังจากส่วนของ try หรือ catch ถูก execute เรียบร้อยแล้ว

6. อธิบายและยกตัวอย่าง การเรียก method จากต่าง class แบบ extends มาพอเข้าใจ

MemberProfile.java

```
public class MemberProfile {  
  
    public static String address = "Nan Thailand";  
  
    public static void displayFullname() {  
  
        System.out.println("Mim");  
  
    }  
  
}
```

MyClass.java

```
public class MyClass extends MemberProfile {  
  
    public static void main(String[] args) {  
  
        displayFullname();  
  
        System.out.println(address);  
  
    }  
  
}
```

จะเรียก MemberProfile ว่า Superclass และ MyClass เรียกว่า Subclass นั้นหมายถึง MyClass สามารถเรียกใช้งาน Method รวมทั้งพวกตัวแปร ต่าง ๆ ที่อยู่ภายใน Class ของ MemberProfile ได้ทันที

สำหรับการ extends Class (Inheritant) ในภาษา Java จะใช้ในกรณีที่ต้องการขยายขอบเขต การทำงานของ Class ปัจจุบันไปยัง Class ปลายทางที่ถูก extends

7. อธิบายและยกตัวอย่าง for แกรับค่า ส่วน for ที่สองประมวลผลมาพอเข้าใจ

```
import java.io.*;
public class one {
    public static void main(String args[]) throws IOException {
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        int i[] = new int[5];
        int plus = 0, Minus = 0;
        for (int j=0; j<5; j++) {
            i[j] = Integer.parseInt(stdin.readLine());
        }
        for (int j=0; j<5; j++) {
            plus = i[0] + i[1] + i[2] + i[3] + i[4];
            Minus = i[0] - i[1] - i[2] - i[3] - i[4];
        }
        System.out.println("plus = " + plus);
        System.out.println("Minus = " + Minus);
    }
}
```

ยกตัวอย่าง for แกรับค่า ส่วน for ที่สองประมวลผล

โปรแกรมนี้ รับข้อมูลจากแป้นพิมพ์ เก็บลงอาร์เรย์ตัวหนึ่ง จำนวน 5 ค่า แล้วสลับกัน ส่วน for ที่สองก็ก็นำมาคำนวณหา บวก และลบ

8. อธิบายและยกตัวอย่าง ตัวแปรแบบ instance และแบบ local variable มาพอเข้าใจ

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

ตัวแปรแบบ Local นั้นจะ ไม่มีค่า Default ดังนั้นเราจึงควรที่จะใส่ค่าให้กับตัวแปร Local ในตอนที่ เราได้ทำการประกาศตัวแปร

ตัวอย่าง การประกาศตัวแปรแบบ Local - ประกาศตัวแปรแบบ Local ไว้ใน method

```
public class Test {  
    public static void main(String[] args) {  
    }  
    // Method ที่ทำการประกาศตัวแปร  
    public static void declareVariableInMethod(){  
        int i = 0; // ตัวแปรที่ถูกประกาศใน method  
        System.out.println(i);  
        // ตัวแปรจะถูกทำลายหลังจากจบ method  
    }  
}
```

ตัวแปรแบบ Instance นั้นจะถูกประกาศเอาไว้ในคลาส แต่จะอยู่ภายนอกของ method, constructor และ block ต่างๆ ซึ่งตัวแปรแบบ Instance นั้นจะถูกสร้างขึ้นก็ต่อเมื่อมีการสร้าง Object ขึ้นมา การเรียกใช้ตัวแปรแบบ Instance นั้นสามารถเรียกใช้ภายใน method, constructor หรือใน block ได้

หากเราไม่กำหนดค่าให้ตัวแปร Instance ตัวแปร Instance จะมีค่า Default ให้เองโดยอัตโนมัติ ซึ่งขึ้นอยู่กับชนิดของตัวแปรข้อมูลที่เราได้ทำการประกาศไว้ เช่น ประกาศตัวแปรเอาไว้เป็น int i; ค่า default ของ i จะเท่ากับ 0 โดยอัตโนมัติ

ตัวอย่าง การประกาศตัวแปรแบบ Instance

```
public class Student {  
    // ตัวแปรแบบ Instance โดยใช้ Access Modifier เป็น private เพื่อให้ตัวแปรเหล่านี้ถูกเห็นได้แค่ภายในคลาสนี้ เท่านั้น  
    private String instanceVariable; // ค่า Default ของ String จะเป็นค่า null  
    private int i = 2; // เราสามารถทำการกำหนดค่าให้กับตัวแปรได้เลย  
    // การเรียกใช้ตัวแปรใน Constructor  
    public Student(){  
        instanceVariable = "Test Instance Variable";  
        i = 3;  
    }  
}
```

9. อธิบายและยกตัวอย่าง การทำ casting มาพอเข้าใจ

การเปลี่ยนแปลงชนิดของข้อมูล (casting)

ใช้แปลงชนิดข้อมูล จากชนิดหนึ่งเป็นอีกชนิดได้ โดยไม่สนใจขนาด แต่ถ้า cast ข้อมูลที่มีขนาดใหญ่ไปสู่ข้อมูลที่มีขนาดเล็ก จะทำให้สูญเสียค่าความเป็นจริงของข้อมูลนั้นได้ ดังนั้นการประกาศค่าตัวแปรและcast ต้องทำด้วยความระมัดระวัง

การแปลงค่าระหว่าง String และ int

รูปแบบ

ตัวแปรแบบ String =Integer.toString(ตัวแปรแบบint)

ตัวอย่าง

```
int num1=33;
```

```
String x;
```

```
x=Integer.toString(num1) // x มีค่าเท่ากับ “33”
```
