

1. Given code in Java:

```
public class Demo {  
    public static void main(String[] args) {  
        System.out.print(m1(3));  
    }  
  
    public static int m1(int n) {  
        if (n == 1) return 1;  
        else return n * m1(n-1)  
    }  
}
```

(5 marks) Show how you can rewrite this program to minimize the use of activation record.

2. A modified version of Java allows nested method definitions.

```
public static void main(String[] args){
    int x =5;
    int y =3;
    int z = 2;
    public void method01(int x){
        public void method2(int y){
            x = z+1;
            int m = x - y;
            method3(m);
            System.out.println(x + "," + y + "," + z); //line1
        }

        public void method3(int m){
            int z = y +x+m;
            y = x + z
            System.out.println(x + "," + y + "," + z); //line2
        }

        int m = x + y + z;
        method2(m);
    }
    method1(x+1);
    System.out.println(x + "," + y + "," + z); //line3
}
}
```

You must:

- Count method parameter declaration as a variable declaration.
- For dynamic scope, a variable declaration no longer exists if its method or scope has finished its execution.

- a. (4.5 marks) If this code uses static scope, what will be printed at line1, line 2, and line 3?

- b. (4.5 marks) If this code uses dynamic scope, what will be printed at line1, line 2, and line 3?

3. Given code

```
class Box { //Box containing value
public:
    Box(double l, double w, double h, double v) {
        length = l;
        width = w;
        height = h;
        value = v;
    }
    virtual double volume() { return length*width*height; } //virtual tells compiler that the subclass can override this method
    virtual double containedValue() { return value; } // it allows the subclass method to be called from pointer of the superclass type
protected:
    double length;
    double width;
    double height;
    double value;
};

class Cube : public Box { //subclass of Box
public:
    Cube(double s, double v) : Box(s, s, s, v) { } //use parent's constructor to initialize cube of equal
                                                //sides and its value
    double containedValue() { return value*value*value; }
    void cloneCube() {
        Cube* clone = new Cube(length, value); //line k
    }
};

...

void doSomething () {
    Box* boxPtr1; //line a
    Box* boxPtr2; //line b
    Box box1(1.0, 2.0, 3.0, 4.0); //line c
    Cube cube1(2.0, 3.0); //line d
    double result; //line e
    boxPtr1 = new Cube(3.0, 4.0); //line f
    result = boxPtr1->containedValue(); //line g
    boxPtr2 = &box1; //line h
    boxPtr2 = &cube1; //line i
    cube1.cloneCube(); //line j
}
```

- a. (3 marks) At line h and i, is there a memory leak? Explain (Thai language allowed)

- b. (3 marks) After cloneCube() exits, are there memory spaces not cleared? Is there any memory leak? Explain (Thai language allowed).