

隨堂練習卷

建議作答 40 分鐘 · 總分 100

§ 1 一、選擇題 (每題 6 分, 共 30 分)

1. `** (B) **` — 宣告括號內是參數 (parameter)，呼叫時實際傳入的值是引數 (argument)。
2. `** (B) **` — ``static`` 屬類別本身，用 ``類名.方法(...)`` 呼叫、不需建物件 (如 ``Math.sqrt``)。
3. `** (B) **` — 呼叫堆疊後進先出 (LIFO)：最後呼叫的方法最先返回。
4. `** (B) **` — `enum` 取值用 `型別.常數` (``Status.WON``)，比較用 ``==``；(A) 不是字串、(C) 不必 ``new``、(D) `enum` 常數是 ``static final`` 不可重新賦值。
5. `** (B) **` — Java 一律值傳遞；物件傳的是參照值的拷貝 (可改物件內容、不可換物件本身)。

§ 2 二、呼叫堆疊與值傳遞 (共 16 分)

6. 呼叫堆疊變化：

```
main 開始 → push main 框 (result = ?)
呼叫 square(5) → push square 框 (x = 5) 到頂端
square 執行 → return x * x = 5 * 5 = 25
return 後 → pop square 框, 25 交回 main
           → result = 25, 再 pop main 框, 堆疊清空
```

`push` : **square** 框 (x = **5**) ; `return` 後 pop，回傳值 **25** 交回 `main`。 `result` = **25** (後進先出：最後 push 的 square 最先 pop。)`

7. 輸出：

5

``addOne`` 改的是參數 ``n`` (值的拷貝)，基本型別值傳遞，動不到呼叫端的 ``a``，故仍是 5。

§ 3 三、改錯 (每題 6 分, 共 18 分)

8. ``nextInt(6)`` 得 ``0~5``，不是 ``1~6``，少了 ``+1`` 平移。要 `1~6` 得加 1：

```
int face = 1 + rand.nextInt(6); // 0~5 平移成 1~6
```

9. 兩個方法參數列相同 (都是 `int s`)，只差回傳型別 → **不能多載**，編譯錯誤。要讓參數列不同，例如把圓面積版改收 `double`：

```
static int area(int s){ return s*s; }
static double area(double r){ return Math.PI * r * r; }
```

10. `main` 是 `static`，不能直接呼叫實例方法 `hello()`。要先建物件再透過物件呼叫，或把 `hello` 改成 `static`：

```
public static void main(String[] a){ new T().hello(); } // 先 new
// 或: static void hello(){ ... } 並用 hello();
```

§ 4 四、程式設計 (共 36 分)

11. (18 分)

```
import java.security.SecureRandom;

public class RollDice {
    private static SecureRandom rand = new SecureRandom();
    public static int roll() {
        return 1 + rand.nextInt(6); // nextInt(6) 得 0~5, +1 平移成 1~6
    }
    public static void main(String[] args) {
        for (int i = 0; i < 10; i++) {
            System.out.print(roll() + " "); // 每次點數為 1~6 (亂數)
        }
        System.out.println();
    }
}
```

(評分：`import` + `SecureRandom` 欄位 4 分、`roll()` 縮放 + 平移 8 分、`main` 迴圈擲 10 次並印出 6 分。輸出每次不同，但每個點數都在 1~6。)

12. (18 分)

```
public class MaxOverload {
    public static int maxOf(int a, int b) {
        return (a > b) ? a : b;
    }
    public static int maxOf(int a, int b, int c) {
        return maxOf(maxOf(a, b), c); // 重用兩數版
    }
    public static void main(String[] args) {
        System.out.println(maxOf(3, 9)); // 9
        System.out.println(maxOf(4, 8, 2)); // 8
    }
}
```

(評分：兩數版 6 分、三數版 8 分、`main` 印兩結果 4 分。三數版直接比較三者亦可。)