1 Packages Have Arrived

In the following classes, cross out the lines that will result in an error (either during compilation or execution). Next to each crossed-out line write a replacement for the line that correctly carries out the evident intent of the erroneous line.

Each replacement must be a single statement. Change as few lines as possible.

After your corrections, what is printed from running `java P2.C5`?

```java
package P1;

class C1 {
    private int a = 1;
    protected int b = 2;
    int c = 3;

    public static int d() {
        return 13;
    }

    public void setA(int v) { a = v; }
    public void setB(int v) { b = v; }
    public void setC(int v) { c = v; }

    public int getA() { return a; }
    public int getB() { return b; }
    public int getC() { return c; }

    public String toString() {
        return a + " " + getB() + " " + getC() + " " + d();
    }
}

package P1;

class C2 extends C1 {
    public C2() {}
    public C2(int a, int b, int c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    public static int d() {
        return 14;
    }
}
```
public C1 gen() {
    return new C3();
}

package P1;
class C3 extends C2 {
    private int a = 15;
    public String toString() {
        return a + " " + getB() + " " + getC() + " " + d();
    }
}

package P2;
class C4 extends C2 {
    public int getB() {
        return 2 * b;
    }

    public C4(int a, int b, int c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    public C4(int v) {
        this.a = this.b = this.c = v;
    }
}

package P2;
class C5 {
    public static void main(String... args) {
        C1 x = new C1();
        C2 y = new C4(20, 30, 40);
        C3 z = y.gen();
        System.out.println(x);
        System.out.println((P1.C2) y);
        System.out.println(z);
    }
}
2 Bit Operations

In the following questions, use bit manipulation operations to achieve the intended functionality and fill out the function details -

(a) Implement a function `isPalindrome` which checks if the binary representation of a given number is a palindrome. The function returns true if and only if the binary representation of `num` is a palindrome. Assume `num` is 32 bits.

For example, the function should return true for `isPalindrome(0xDEADDAED)` since binary representation of 9 is `1001` which is a palindrome.

```java
/**
 * Returns true if binary representation of num is a palindrome
 */
public static boolean isPalindrome(int num) {
    // Code goes here
}
```
(b) Implement a function \texttt{swap} which for a given integer, swaps two bits at given positions. The function returns the resulting integer after bit swap operation.

For example, when the function is called with inputs \texttt{swap(31, 3, 7)}, it should reverse the 3rd and 7th bits from the right and return 91 since \texttt{31} (00011111) would become \texttt{91} (01011011).

```java
/**
 * Function to swap bits at position a and b (from right) in integer num
 */
public static int swap(int num, int a, int b) {
    // IMPLEMENTATION
    return num;
}
```

### 3 Bits Runtime

Determine the best and worst case runtime of \texttt{tricky}.

```java
public void tricky(int n) {
    if (n > 0) {
        tricky(n & (n - 1));
    }
}
```

Best Case: $\Theta( \quad )$, Worst Case: $\Theta( \quad )$