Packages Have Arrived

Here is a video walkthrough of the solutions.

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**?

```
1  package P1;                                          Write output here:
2  class C1 {
3      private int a = 1;                               ____________________________
4      protected int b = 2;
5      int c = 3;                                       ____________________________
6
7      public static int d() {                          ____________________________
8          return 13;
9      }
10     public void setA(int v) { a = v; }
11     public void setB(int v) { b = v; }
12     public void setC(int v) { c = v; }
13     public int getA() { return a; }
14     public int getB() { return b; }
15     public int getC() { return c; }
16
17     public String toString() {
18         return a + " " + getB() + " " + getC() + " " + d();
19     }
20 }
21 ____________________________________________
22
23 package P1;
24 class C2 extends C1 {
25     public C2() {}
26     public C2(int a, int b, int c) {
27         this.a = a;
28         this.b = b;
29         this.c = c;
30     }
31     public static int d() {
32         return 14;
33     }
34     public C1 gen() {
35         return new C3();
36     }
37 }
```

```
_____________________________________________

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);
    }
}
```

Solution:

Here is a video walkthrough of the solution.

```
package P1;                                    Write output here:

public class C1 {

    private int a = 1;                         ____ 1 2 3 13 _________
    protected int b = 2;
```

```
    int c = 3;                                          ____ 20 60 40 13 ______

    public static int d() {                             ____ 15 2 3 14 ________
        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;

public class C2 extends C1 {

    public C2() {}
    public C2(int a, int b, int c) {

        setA(a);

        this.b = b;
        this.c = c;
    }
    public static int d() {
        return 14;
    }
    public C1 gen() {
        return new C3();
    }
}
```

```
package P1;

public class C3 extends C2 {

    private int a = 15;
    public String toString() {
        return a + " " + getB() + " " + getC() + " " + d();
    }
}
```

```
package P2;

class C4 extends P1.C2 {

    public int getB() {
        return 2 * b;
    }
    public C4(int a, int b, int c) {

        setA(a);

        this.b = b;

        setC(c);

    }
    public C4(int v) {

        super(v,v,v);

    }
}
```

```
package P2;
class C5 {
    public static void main(String... args) {

        P1.C1 x = new P1.C1();

        P1.C2 y = new C4(20, 30, 40);

        P1.C3 z = (P1.C3) y.gen();

        System.out.println(x);
        System.out.println((P1.C2) y);
        System.out.println(z);
    }
}
```

Fixes:
The following lines need to be fixed:
Line 2: In order to access `C1` in another package `P2`, it needs to be a `public class`.
Line 23: Similar logic to line 2–in order to access `C2` in another package, it needs to be a `public class`.
Line 27: `a` is a private variable, so subclasses like `C2` cannot access it directly–instead, it must use the `setA` method.
Line 41: Similar to line 2 and 23.
Line 50: Since we have not imported `P1`, we must use preface the class we want to use by its full package name.
Line 55: Similar to Line 27, `a` is a private variable.
Line 57: A variable with no declared access modifier is package private, which means it can only be accessed within the same package. `C4` is in package `P2`, so it

must use the public method `setC` instead of directly accessing `c`.

Line 60: Again, `C4` cannot directly access `a` or `c`, but it can call its parent's constructor with `super`, which achieves the desired effect.

Line 68, 69: Similar to Line 50; without importing `P1`, the full package name must be used.

Line 70: `y` has static type `C2`, and `C2.gen` has return type `C1`. However, we know that the method actually returns an object of type `C3`, so casting allows the assignment to compile.

Print output:

Line 72: `C1`'s `toString` method simply prints the `a, b, c`, and `d()` values.

Line 73: `C4` has `a, b, c` as `20, 30, 40` respectively. It inherits the `toString` method from `C1`. However, its `getB` method overrides the `getB` method in `C1`, so `getB()` will return 60. Note that the `d()` method is static, so the method that is run is decided at compile time (there is no overriding for static methods). Thus, at compile time, the compiler finds `getString()` inside of `C1` and also uses `C1`'s `d()` method. The final values printed are `20, 60, 40, 13`.

Line 74: `y.gen()` calls the no-argument `C3` constructor. Note that `C3` has its own `private a` with a value of 15. However, it still inherits `b` and `c` from `C1`, and its `d()` method from `C2`. This gives the final output `15 2 3 14`.