Candies

Suppose you have a dollar in your pocket, and you see a shelf with a row of candies priced at 10¢, 20¢, 30¢, and so forth. You buy one of each candy, starting with the one that costs 10¢, until you can’t afford to buy the next candy on the shelf.

The following program tries to find the number of candies you buy and the amount of change you get. What’s wrong with the program?

```java
public class Candies {
    public static void main(String[] args) {
        double funds = 1.00;
        int itemsBought = 0;
        for (double price = 0.10; funds >= price; price += 0.10) {
            funds -= price;
            itemsBought++;
        }
        System.out.println(itemsBought + " items bought.");
        System.out.println("Change: $" + funds);
    }
}
```

Click here to expand...

Bug: Using double for currency calculations

Solution: Use `BigDecimal` for currency calculations, as it provides arbitrary-precision arithmetic and can accurately represent decimal numbers.

Here's the corrected version of the code using `BigDecimal`:

```java
import java.math.BigDecimal;

public class Candies {
    public static void main(String[] args) {
        BigDecimal funds = new BigDecimal("1.00");
        int itemsBought = 0;
        final BigDecimal increment = new BigDecimal("0.10");

        for (BigDecimal price = new BigDecimal("0.10"); funds.compareTo(price) >= 0; price = price.add(increment)) {
            funds = funds.subtract(price);
            itemsBought++;
        }
        System.out.println(itemsBought + " items bought.");
        System.out.println("Change: $" + funds);
    }
}
```