x = new Foo();

When you create an object:

1. Java looks for a "gap" and allocate there.

2. Put at end of the heap.

3. The address of the object is used as the reference.
public class Account {
    private int balance;
    public Account(int openingBalance) {
        balance = openingBalance;
    }
    public void deposit(int dollars) {
        balance = balance + dollars;
    }
    public boolean withdraw(int dollars) {
        if (dollars <= balance) {
            balance = balance - dollars;
            return true;
        } else
            return false;
    }
    public boolean transfer(int dollars, Account destination) {
        if (withdraw(dollars)) {
            destination.deposit(dollars);
            return true;
        } else
            return false;
    }
    public String toString() {
        return ("$" + balance + ".00");
    }
}
Problem: \[ \text{gcd}(m, n) \] greatest common divisor of \( m \) and \( n \)

No formula, so...

Need an ALGORITHM — basic idea for how to solve a problem— series of steps

Solve in General
Start w/ something easier:

\[ n! = n \cdot (n-1) \cdot (n-2) \cdots 1 \]

\[ 0! = 1 \]

```java
int factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n-1);
}
```

Does it work? 

\[ \text{Vote: } \frac{\text{YES}}{\text{NO}} \]

\[ \frac{16}{\sim 20} \]

Let's try some examples...
int factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n-1);
}

factorial(0) ⇒ 1

using substitution model to evaluate

\[
\begin{align*}
\text{factorial}(3) &= 3 \cdot \text{factorial}(2) \\
    &= 3 \cdot 2 \cdot \text{factorial}(1) \\
    &= 3 \cdot 2 \cdot 1 \cdot \text{factorial}(0) \\
    &= 3 \cdot 2 \cdot 1 \cdot 1 = 6
\end{align*}
\]