Last question:

Next permutation:

\[ X \begin{array}{cccc} 2 & 4 & 3 & 1 \end{array} \]

3 4 2 1

3 1 2 4
Continuing example from last time.

In some other method in some other class:

```java
myClass x = new myClass(7, -12);

System.out.println(x.happy); // Print 7
System.out.println(x.getHappy()); // Print 7
```

Diagram:
```
x
```
```java
    myClass Object
       |
       v
    7  happy
       |
       v
   -12  sad
```
re-write constructor using the 'this' reference.

```cpp
myClass (int happy, int sad) {
    this->happy = happy;
    this->sad = sad;
}
```

Ex. Person 4
One fact about inheritance:
Descendant classes possess all members (variables & methods) of their ancestor classes.
Note: Object has 11 public methods. We are concerned with 2 of them both instance methods:

- `toString()`
- `equals()`

`toString()` : used to create a string representation of an object. To do this, must redefine `toString()`, i.e. override
Ex. Persons

To override an inherited method must use same header as in the ancestors. In doing this we may not narrow access privileges.

Access modifiers:

- Private
- (Package)
- Protected
- Public
- Default
- Narrow
- Widest
1. **Note difference:**

   * **Override an inherited method:**
     * give a defn with same signature
     * may not narrow access

   * **Overload a method:**
     * give a defn with a different signature