# Unit Testing

# What is debugging?

# How do you currently debug your code?

Slides by Mark Hancock
(adapted from notes by Craig Schock)

# Analysis & Design
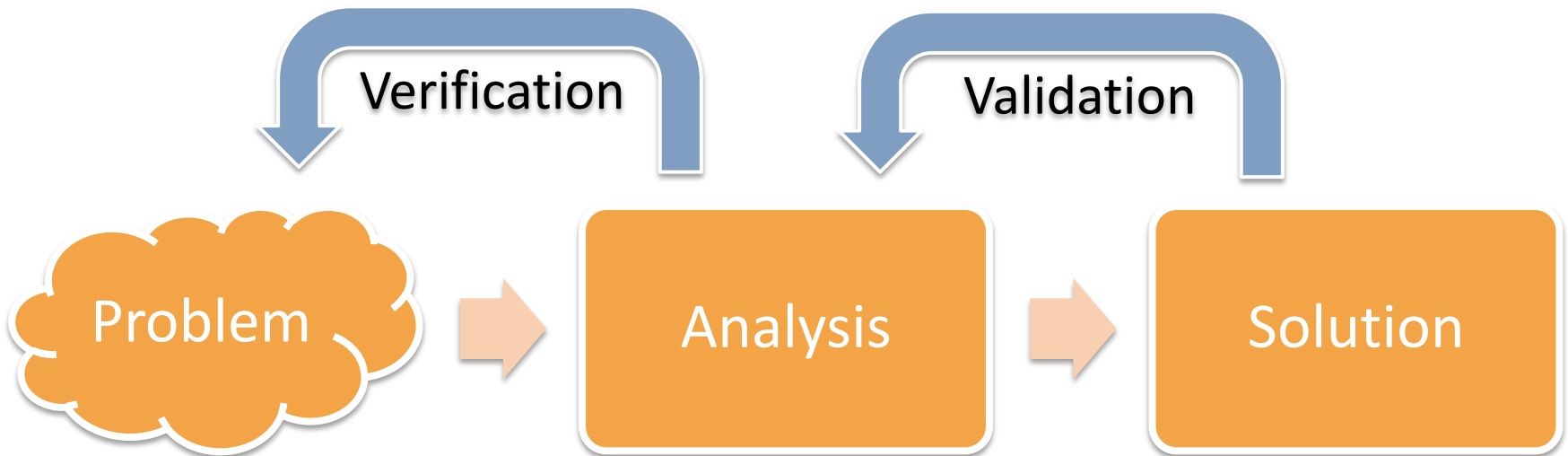
# Why is it important to validate the solution against the analysis?

Slides by Mark Hancock
(adapted from notes by Craig Schock)

In other courses or life experiences, how do you go about testing the validity of something?
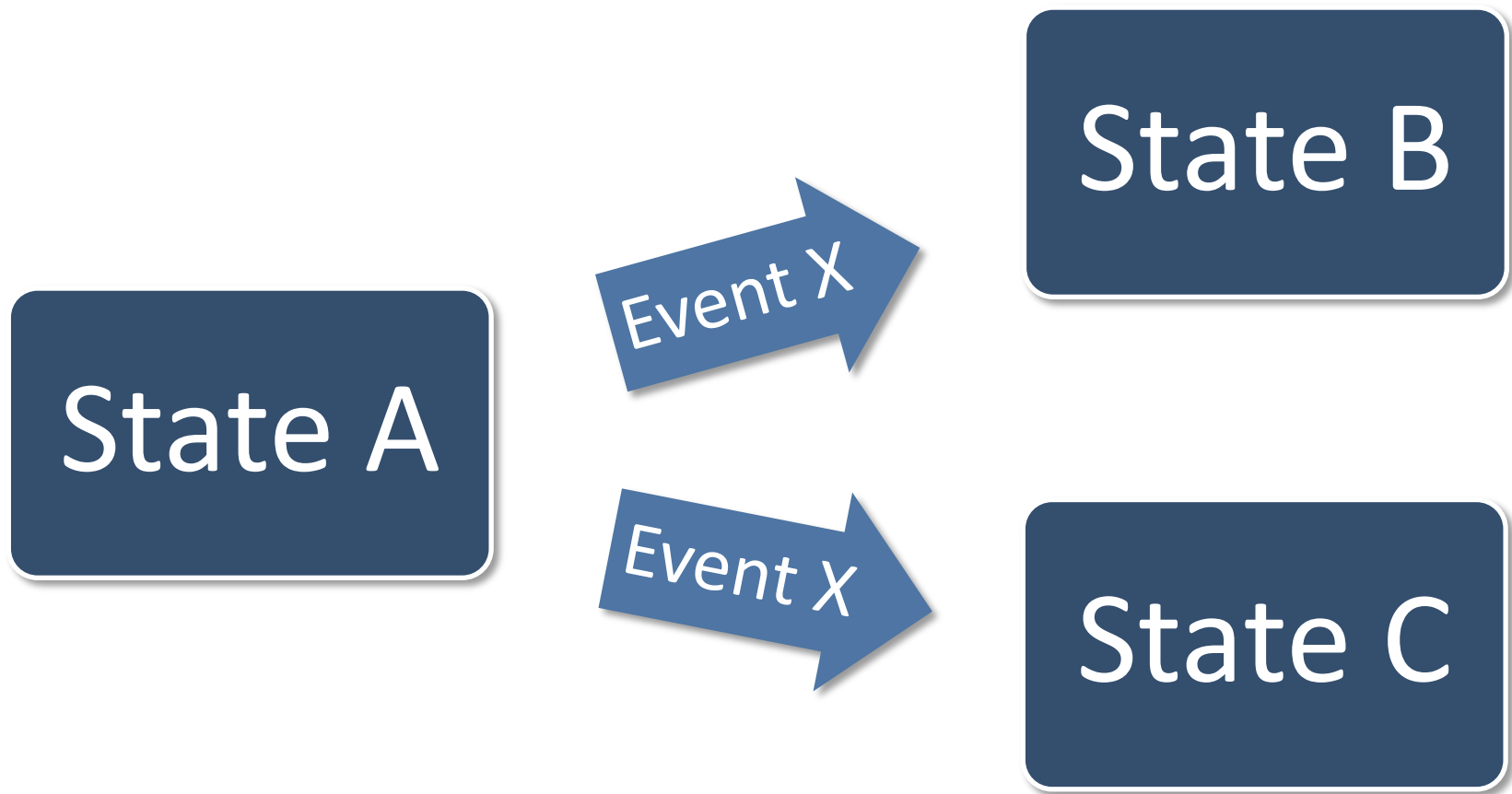
# Hypothesis Testing

- (Normal) scientific approach:
  - Problem
  - Hypothesis
  - Hypothesis Test

What are some properties that distinguish computer programs from events in the physical world (e.g., the weather, people's actions)?

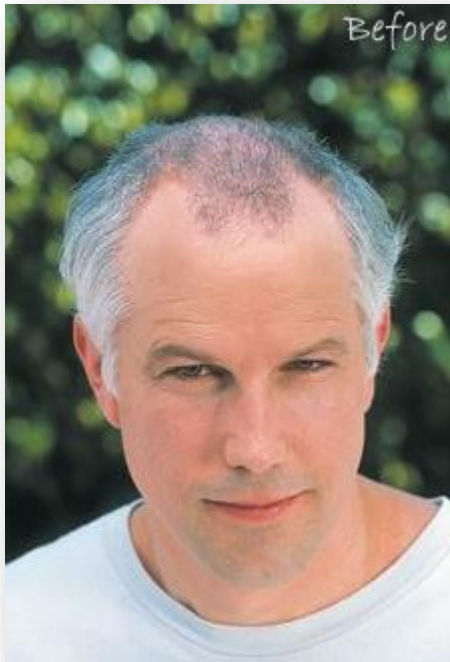Slides by Mark Hancock
(adapted from notes by Craig Schock)

# Determinism

State A → Event X → State B

Slides by Mark Hancock
(adapted from notes by Craig Schock)

# Non-Determinism

State B

Event X

State A

Event X

State C

# Example: Hair Restoration

**Before**                                    **After**



Source: http://www.hairclub.com
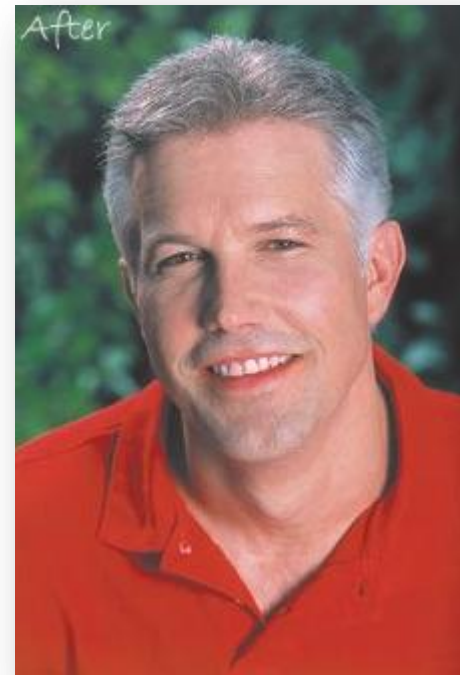
# Example: Photoshop

**Before**

**After**



Source: http://www.hemmy.net

Slides by Mark Hancock
(adapted from notes by Craig Schock)

How could you formulate a test using a before state and an after state?

# Unit Testing

- Base unit of implementation: objects
- Process:
  - setup known state of object(s)
  - send messages to object(s)
  - test resulting object(s)'s state

Slides by Mark Hancock
(adapted from notes by Craig Schock)

# What kind of messages can we send to objects?

Slides by Mark Hancock
(adapted from notes by Craig Schock)

# Types of Messages

- Reveal State

- Change State
  - create objects
  - delete objects
  - create associations
  - delete associations
  - change internal state

# Exercise

- Create a unit test for the "passHighestCard" method in the HonestPlayer class
    - describe the initial known state
    - describe the method you would call
    - describe the expected resulting state

# "Do Assignment" Algorithm

```
while (!assignment.isDone())
{
    // write some code
    while (!assignment.compiles())
    {
        // fix syntax errors
    }

    while (!assignment.behavesCorrectly())
    {
        // fix semantic errors
    }

    if (caffeineSupply.isDepleted())
    {
        sleep(28800000);
    }
}
```

# What's are the disadvantages of this algorithm?

# Disadvantages

- tedious

- perform one test at a time

- test things we know will likely work

# Automated Testing

- Write code that automatically tests the state

- Write many tests that can be run at once

- As you write new code, if any of them fail, you know you've introduced a bug.

It tends to be hard to convince programmers to use unit tests. Why do you think that might be?

Slides by Mark Hancock
(adapted from notes by Craig Schock)

# Unit Testing Summary

- A process of validation

- Makes use of determinism

- Send messages to objects in known state and test the result

- Can create many tests on all of the units in your code

Slides by Mark Hancock
(adapted from notes by Craig Schock)

# Next Class

- JUnit (Unit Testing in Java)
- Midterm Review (in Java)

Slides by Mark Hancock
(adapted from notes by Craig Schock)