design intro

Processes

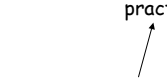
- Requirements ✓
- Design
- Implementation
- Testing

Design

practices, principles, patterns

Design

practices, principles, patterns



e.g. diagrammatic modeling

Design

practices, principles, patterns

e.g. "no forgery"

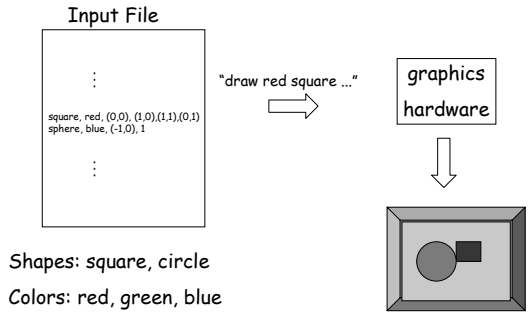
Design

practices, principles, patterns

e.g. singleton

Today

- Intro to design principles:
  - background/history (OO principles)
- Project 2 intro/exercise



Shapes: square, circle

Colors: red, green, blue

rule

- draw red circles
- draw green squares
- draw blue squares
- ignore everything else

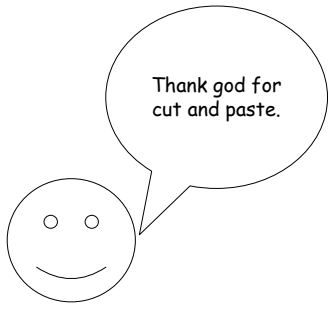
pseudo code

```
Open file
While not at end of file
   read shape, color
        if shape is square then read four vertices
                if color is blue then "draw blue square…"
                if color is green then "draw green square…"

        else if shape is circle then read center and radius
                if color is red then then "draw red circle…"
        else skip this line of input
Close file
```

whoops … I meant

- Shapes: square, circle, triangle
- Colors: red, blue, green, purple
- Rule:
  - Draw blue and purple squares
  - Draw red and green circles
  - Draw every triangle
  - Ignore everything else



hacker joe

pseudo code

```
Open file
While not at end of file
    read shape, color
          if shape is square then read four vertices
                    if color is blue then "draw blue square…"
                    if color is purple then "draw green square…"
          else if shape is circle then read center and radius
                    if color is red then "draw red circle…"
                    if color is green then then "draw red circle…"
          else if shape is triangle then read four vertices
                    if color is blue then "draw blue square…"
                    if color is purple then "draw green square…"
                    if color is red then "draw red circle…"
                    if color is green then then "draw red circle…"
          else skip this line of input


Close file
```

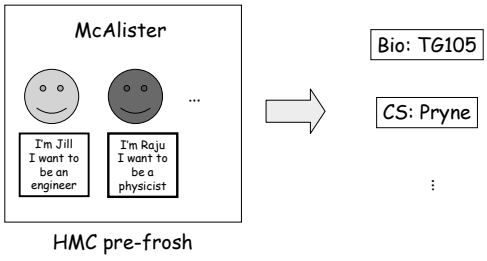
pseudo code

```
Open file
While not at end of file
    read shape, color
          if shape=square then read four vertices
                    if color is blue then "draw blue square…"
                    if color is purple then "draw green square…"
          else if shape=circle then read center and radius
                    if color is red then "draw red circle…"
                    if color is green then then "draw red circle…"
          else if shape=triangle then read four vertices
                    if color is blue then "draw blue square…"
                    if color is purple then "draw green square…"
                    if color is red then "draw red circle…"
                    if color is green then then "draw red circle…"
          else skip this line of input


Close file
```

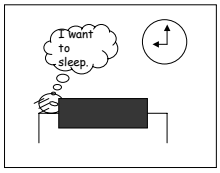
whoops … I meant

new task

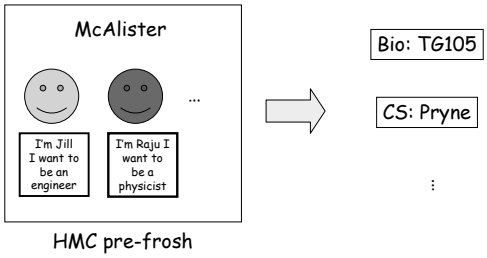


where are the CS pre-frosh?

somewhere in west



new task



procedure

- post classroom assignments
- post campus map
- tell students to
  - refer to the posted notices
  - go where they belong

whoops

CS has been moved to pepsi room

procedure

- post classroom assignments
- post campus map
- tell students to
  - refer to the posted notices
  - go where they belong

strategies

functional ⟺ object oriented

NOTE: this has nothing to do with "functional languages"

strategies

functional
step by step procedure

⟺

object oriented
delegate responsibility

strategies

functional approach
- control is concrete, specific
- rigid, hard to change

⟺

object oriented approach
- control is abstract, general
- adaptable, easy to change

change is inevitable!

History: functional → OO

- modularization

modularize

```
Open file
While not at end of file
   read shape, color
        if shape is square then read vertices
                testAndDrawSquare(color, vertices)
        if shape is circle then read center and radius
                testAndDrawCircle(color, center, radius)
Close file
```

functional → OO

- modularization
- user-defined data types

user define data types

```
Open file
While not at end of file
   read shape, color
        if shape=square then read vertices and create theSquare
                testAndDrawSquare(theSquare)
        if shape=circle then read center, radius and create theCircle
                testAndDrawCircle(theCircle)
Close file
```

functional → OO

- modularization
- user-defined data types
- union (abstract data types)

abstract data types

```
Open file
While not at end of file
   theShape = readNextShape()
   testAndDrawShape(theShape)
Close file
```

functional → OO

- modularization
- user-defined data types
- union (abstract data types)
- encapsulation

encapsulation

```
Open file
While not at end of file
   theShape = readNextShape()
   theShape.testAndDraw()
Close file
```

Some OO principles

- favor composition over inheritance
- think like an object
- etc.

Triangle World

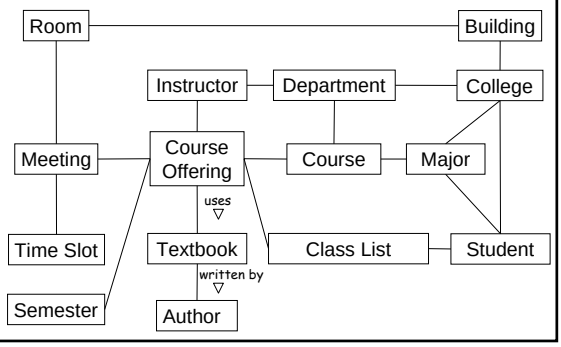
I have a sealed room in which some triangles are magically suspended in the air. Their position and orientation is random. There is also a ball in the room. It has some position and velocity at time t. (It does not intersect any triangles at time t.)

Describe the room and the ball's movement between time t and t+Δt . (In other words, write the dt-timestep use case.)

Triangle World

Build a domain model for triangle world.

Domain Model Example



Example

Which did you have?

- Ball
- Sphere
- Triangle
- Polygon
- Thing
- Shape
- Moving shape
- Moving object
- Velocity
- Walls
- Floor
- World
- Force
- Gravity
- Collision
- Position
- Item

Ball vs. Sphere vs. Shape

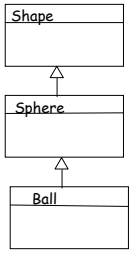
What is the relationship?

design heuristic

think like an object

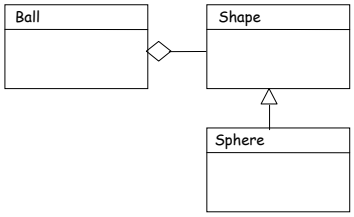
Ball vs. Sphere vs. Shape

What is the relationship?



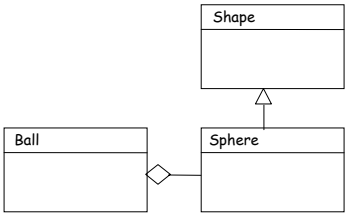
Ball vs. Sphere vs. Shape

What is the relationship?



Ball vs. Sphere vs. Shape

What is the relationship?

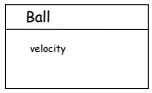


Ball vs. Moving

What is the relationship?

Shape vs. Moving

What is the relationship?

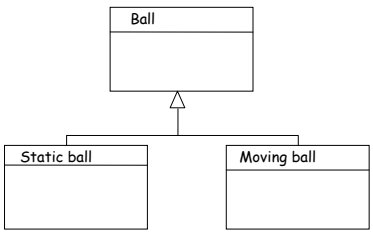


ball doesn't move?

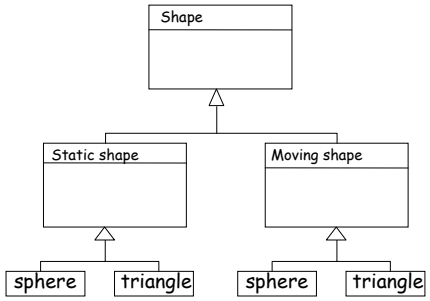
set velocity to 0

Shape vs. Moving

What is the relationship?

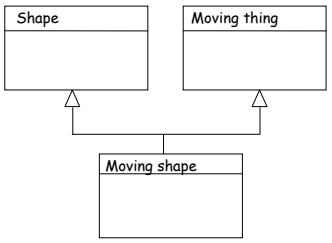


Shape vs. Moving



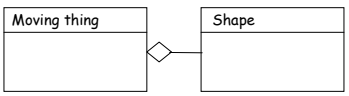
Shape vs. Moving

What is the relationship?

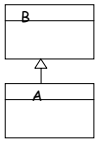


Shape vs. Moving

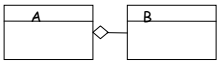
What is the relationship?



Inheritance vs. Composition



white-box reuse



black-box reuse

design heuristic

favor composition over inheritance

next time: more design principles