

# CS209 \_ LAB2

## Key Content

- Review Class define, abstract class, inheritance and polymorphisms.
- Review StdDraw.
- Comprehensive use above skills and make some interesting animation.

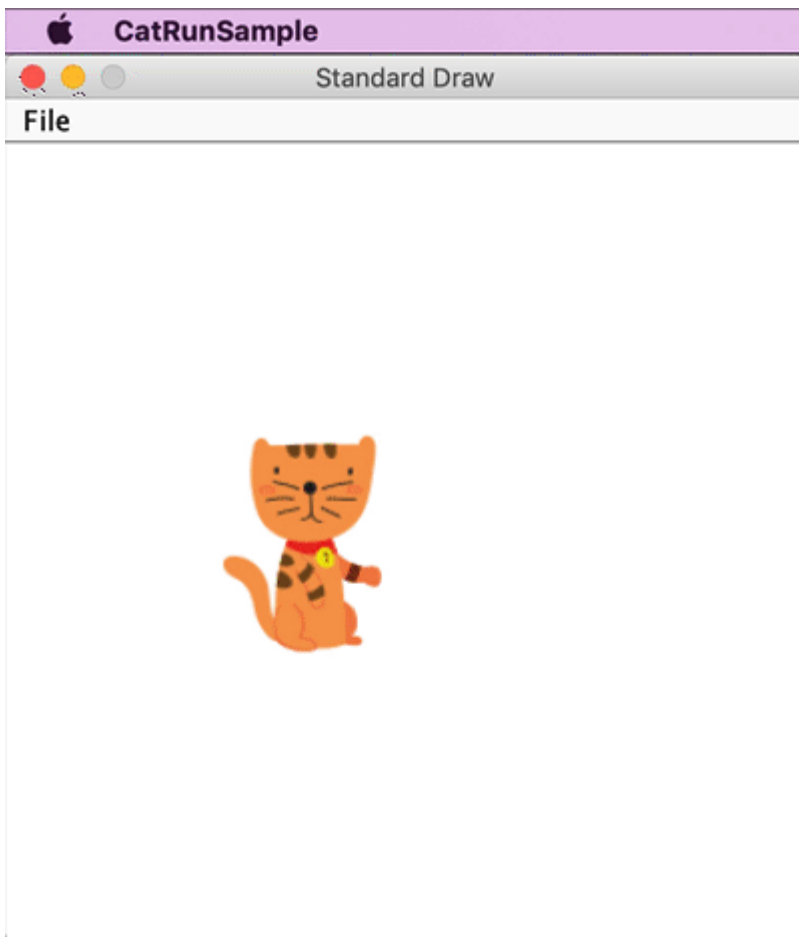
## Before exercise

### 1. Study the slides from Prof. He

### 2. Cat running animation sample code

```

1. class CFG {
2.     public final static int DEFAULT_CANVAS_WIDTH = 400;
3.     public final static int DEFAULT_CANVAS_HEIGHT = 400;
4.     public final static int DEFAULT_SCALE_MIN = -1000;
5.     public final static int DEFAULT_SCALE_MAX = 1000;
6.
7. }
8.
9. public class CatRunSample {
10.
11. public static void main(String[] args) {
12.     StdDraw.setCanvasSize(CFG.DEFAULT_CANVAS_WIDTH, CFG.DEFAULT_CANVAS_HEIGHT);
13.     StdDraw.setScale(CFG.DEFAULT_SCALE_MIN, CFG.DEFAULT_SCALE_MAX);
14.
15.     String imageFileName = "cat.png";
16.     Picture picture = new Picture(imageFileName);
17.     // int scaledHeight = picture.height();
18.     // int scaledWidth = picture.width();
19.
20.     int scaledHeight = CFG.DEFAULT_SCALE_MAX;
21.     int scaledWidth = picture.width() * CFG.DEFAULT_SCALE_MAX / picture.height();
22.
23.     int speed = 80;
24.     StdDraw.enableDoubleBuffering();
25.     double y = 0;
26.     double x = CFG.DEFAULT_SCALE_MIN + 0.5 * scaledHeight;
27.     for (; x < CFG.DEFAULT_SCALE_MAX - 0.5 * scaledHeight; ) {
28.         x += speed;
29.         StdDraw.clear();
30.         StdDraw.picture(x, y, imageFileName, scaledWidth, scaledHeight);
31.         StdDraw.show();
32.         StdDraw.pause(100);
33.
34.     }
35. }
36.
37. }
```



## Exercise

1. Create an abstract class named **Animal**, which has two protected fields: **speed** and **imageName**, has one abstract methods:  
`abstract public void raceDraw(int s, int time, int raceNo, int raceHeight);`  
 The **raceDraw** will draw the animal on the right race track according the **raceNo**, and on the right location according the time and the start point **S**.
2. Create three classes: Pig, Rabbit and Turtle, which are the subclasses of Animal.
  - a. Constructor: each subclass has only one constructor with two parameters including **speed** and **imageName**.
  - b. Override abstract method **raceDraw**:

Class	Algorithm	Example
Rabbit	Rabbit will run 2s, then sleep 2s.	If invoke the raceDraw in 0.1s interval, speed is 80, when time is 3.5s, you should draw the rabbit 160 scale units from start point.
Turtle	Turtle will always run.	If invoke the raceDraw in 0.1s interval, speed is 1, when time is 3.5s, you should draw the turtle 3.5 scale units from start point.
Pig	Pig will run 0.5s, then sleep 1s.	If invoke the raceDraw in 0.1s interval, speed is 55, when time is 2.5s, you should draw the pig 55 scale units from start point.

3. Create a class **AnimalRace** for demo.

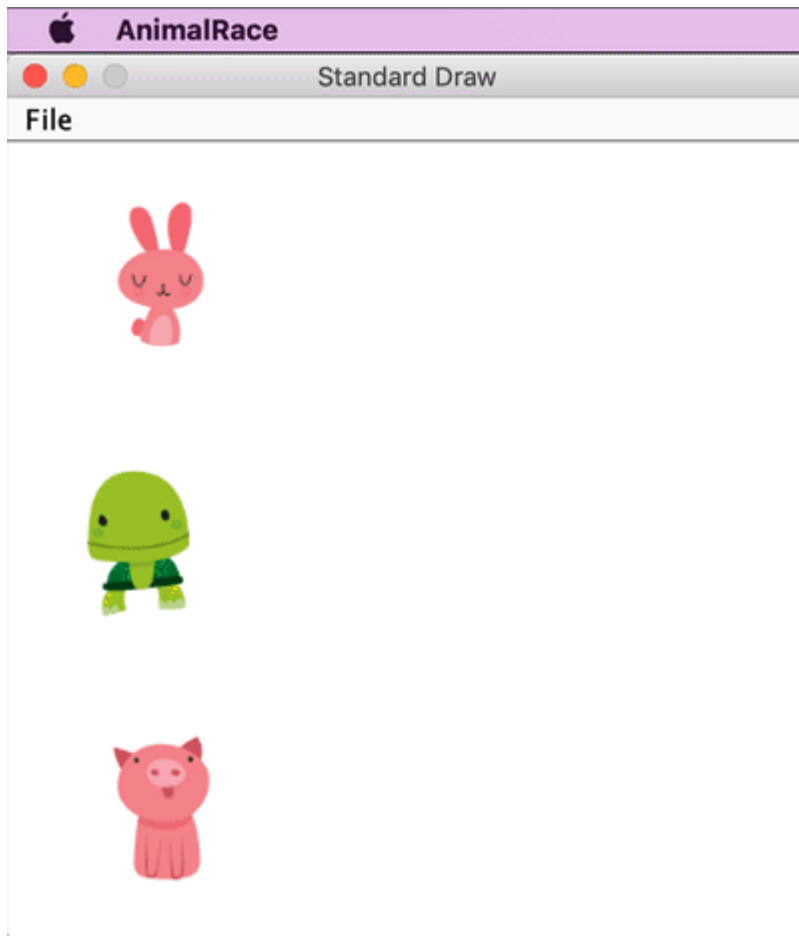
You can use the following value to set your canvas and scale.

```
class CFG {  
    public final static int DEFAULT_CANVAS_WIDTH = 400;  
    public final static int DEFAULT_CANVAS_HEIGHT = 400;  
    public final static int DEFAULT_SCALE_MIN = -1000;  
    public final static int DEFAULT_SCALE_MAX = 1000;  
}  
StdDraw.setCanvasSize(CFG.DEFAULT_CANVAS_WIDTH, CFG.DEFAULT_CANVAS_HEIGHT);  
StdDraw.setScale(CFG.DEFAULT_SCALE_MIN, CFG.DEFAULT_SCALE_MAX);
```

Create an **Animal** ArrayList, add instance of **Rabbit**, **Turtle**, **Pig** one by one.

You need to make an animal race animation. **CatRunSample** is for your reference.

Demo:



4. You can make your creative animation yourself, like a frog jump on the canvas or a bird flying in the sky or a cow walking a curvy path and so on.