Drawing in Java Using the StdDraw Library: MyHouse.java

CIS 110
```java
public class MyHouse {
    public static void main(String[] args) {
        // set the size of the window to 500 pixels by 500 pixels
        StdDraw.setCanvasSize(500, 500);
        StdDraw.clear(StdDraw.BLUE); // draw a blue sky
    }
}
```
Set Window Size

```java
public class MyHouse {
    public static void main(String[] args) {
        // set the size of the window to 500 pixels by 500 pixels
        StdDraw.setCanvasSize(500, 500);
        StdDraw.clear(StdDraw.BLUE); // draw a blue sky
    }
}
```
Color the entire window blue

```java
10  public class MyHouse {
11     public static void main(String[] args) {
12         // set the size of the window to 500 pixels by 500 pixels
13         StdDraw.setCanvasSize(500, 500);
14
15         StdDraw.clear(StdDraw.BLUE); // draw a blue sky
```
public class MyHouse {
    public static void main(String[] args) {
        // set the size of the window to 500 pixels by 500 pixels
        StdDraw.setCanvasSize(500, 500);
        StdDraw.clear(StdDraw.BLUE); // draw a blue sky
Set the color to grass green

```java
// draw a green field
StdDraw.setPenColor(0, 170, 0);
StdDraw.filledRectangle(0.5, 0.25, 0.6, 0.3);
```
Colors

Composed of three elements:

1. Red
2. Green
3. Blue

Values from 0 .. 255
Why 0 ... 255?

Each color is represented by 32 bits:

```
0000 0000 0000 0000 0000 0000 1111 1111
```

Notice there are 8 bits per color component.

The maximum value (all 1's) that can be represented in 8 bits is 255 in decimal.

Therefore, the range for each color component is 0 (off) ... 255 (full).
Set the color to grass green

```java
//draw a green field
StdDraw.setPenColor(0, 170, 0);
StdDraw.filledRectangle(0.5, 0.25, 0.6, 0.3);
```
Solid rectangle

```
// draw green field
StdDraw.setPenColor(0, 170, 0);
StdDraw.filledRectangle(0.5, 0.25, 0.6, 0.3);
```
//draw a green field
StdDraw.setPenColor(0, 170, 0);
StdDraw.filledRectangle(0.5, 0.25, 0.6, 0.3);
Coordinate System

(0, 0)  1.0  +x

+1.0  +y
Lists of 3 x- and y-coordinates

double[] x = {0.255, 0.745, 0.49};
double[] y = {0.70, 0.70, 0.90};
StdDraw.filledPolygon(x, y);
Draw a solid triangle with corners at (0.255, 0.7), (0.745, 0.7), (0.49, 0.9)

double[] x = {0.255, 0.745, 0.49};
double[] y = {0.70, 0.70, 0.90};
StdDraw.filledPolygon(x, y);
Set line thickness (default is 0.002)

```
StdDraw.setPenRadius(0.005); // thicken the pen for outline drawing
```
Draw a rectangle outline

```java
StdDraw.rectangle(250 / 500.0, 260 / 500.0, 120 / 500.0, 90 / 500.0);
```
Keep repeating the instructions in this block forever

```java
// draw a circular cloud at the mouse location as long
// as the mouse is within bounds

while (true) {
    double cloudX = StdDraw.mouseX();
    double cloudY = StdDraw.mouseY();
    StdDraw.setPenColor(StdDraw.WHITE);
    if (cloudY > 0.55) {
        StdDraw.filledCircle(cloudX, cloudY, 0.005);
    }
    StdDraw.show(30);
}
```
Check the mouse's x- and y-coordinates. Call them `cloudX` and `cloudY`.

```java
// draw a circular cloud at the mouse location as long
// as the mouse is within bounds
while (true) {
    double cloudX = StdDraw.mouseX();
    double cloudY = StdDraw mouseY();
    StdDraw.setPenColor(StdDraw.WHITE);
    if (cloudY > 0.55) {
        StdDraw.filledCircle(cloudX, cloudY, 0.005);
    }
    StdDraw.show(30);
}
```
Draw a circle centered at the cursor with radius 0.005, only if the y-coordinate is greater than 0.55!

```java
// draw a circular cloud at the mouse location as long
// as the mouse is within bounds
while (true) {
    double cloudX = StdDraw.mouseX();
    double cloudY = StdDraw mouseY();
    StdDraw setPenColor(StdDraw WHITE);
    if (cloudY > 0.55) {
        StdDraw.filledCircle(cloudX, cloudY, 0.005);
    }
} StdDraw.show(30);
```
Show the changes we just made; Wait to show any further changes until we encounter StdDraw.show() again and at least 30 milliseconds have past.

```java
// draw a circular cloud at the mouse location as long // as the mouse is within bounds
while (true) {
    double cloudX = StdDraw.mouseX();
    double cloudY = StdDraw.mouseY();
    StdDraw.setPenColor(StdDraw.WHITE);
    if (cloudY > 0.55) {
        StdDraw.filledCircle(cloudX, cloudY, 0.005);
    }
    StdDraw.show(30);
}
```

**StdDraw.show() controls the animation speed, or "frame rate."**