SUMMARY
In this assignment you will write a program to design stage lighting.

DEADLINE
This assignment is due on Wednesday, April 30 at 11:00 pm.

SPECIFICATION
You will create two classes to simulate a stage backdrop with colored lighting fixtures. The program will allow a user to see the beams from the light fixtures and the color-mixing that occurs. Create a package called stagedesigner with the following two public classes:

Light
The class Light will represent a single lighting fixture.

 Constructors:
• public Light(int x, int y, double direction, double angle, Color color)
  • x ... the x coordinate of the light (origin of the beam)
  • y ... the y coordinate of the light (origin of the beam)
  • direction ... the direction of the beam in radians
  • angle ... the width of the beam in radians
  • color ... the color of the beam

Methods:
• public Color getColor()
  • Return the color of the light fixture

• public boolean inBeam(int px, int py)
  • Return true if the coordinates (px, py) are in the beam, false otherwise.
**StageDesigner**

The class `StageDesigner` will model the stage.

**Constructors:**

- `public StageDesigner(int width, int height)`
  - `width` ... the width of the stage view (in pixels)
  - `height` ... the height of the stage view (in pixels)

**Methods:**

- `public void addLight(Light light)`
  - Add `light` to the model
  - Throws an exception if the light is out of the stage bounds

- `public Picture getDisplay()`
  - Return a `Picture` object of the stage view.

**NOTES**

**Lights**

Since the most common direction for lighting is pointing down, let a direction of 0 rad be down (south). The direction increases in a clockwise direction, so pointing right (east) would be $\pi/2$ rad, up (north) would be $\pi$ rad, and west would be $3\pi/2$ rad (or $-\pi/2$ rad).

**Colors**

When light beams overlap, the colors mix by summing the component values. e.g., red and blue overlapping make purple, and red, blue, and green all overlapping make white. See the examples below.
EXAMPLES

```java
int width = 1000;
int height = 600;
StageDesigner sd = new StageDesigner(width, height);
sd.addLight(new Light(100, 10, 0.5, 0.5, Color.RED));
sd.addLight(new Light(500, 10, 0.0, 0.5, Color.GREEN));
sd.addLight(new Light(900, 10, -0.5, 0.5, Color.BLUE));
sd.getDisplay().show();
```

```java
int width = 1000;
int height = 600;
StageDesigner sd = new StageDesigner(width, height);
sd.addLight(new Light(100, 10, 0.5, 0.5, Color.RED));
sd.addLight(new Light(500, 400, Math.PI, 0.5, Color.GREEN));
sd.addLight(new Light(900, 10, -0.5, 0.5, Color.BLUE));
sd.getDisplay().show();
```

```java
int width = 600;
int height = 600;
StageDesigner sd = new StageDesigner(width, height);
sd.addLight(new Light(300, 300, 0.0 * Math.PI, 1.0, Color.RED));
sd.addLight(new Light(300, 300, 0.5 * Math.PI, 1.0, Color.GREEN));
sd.addLight(new Light(300, 300, 1.0 * Math.PI, 1.0, Color.BLUE));
sd.addLight(new Light(300, 300, 1.5 * Math.PI, 1.0, Color.YELLOW));
sd.getDisplay().show();
```

```java
int width = 600;
int height = 600;
StageDesigner sd = new StageDesigner(width, height);
Color[] colors = {Color.RED, Color.GREEN, Color.BLUE, Color.YELLOW, Color.MAGENTA, Color.CYAN};
int y, c = 0;
double direction;
double angle = 0.2;
for(int x = 50; x < width; x += 60) {
    y = (int) Math.pow((width/2 - x) / 30.0, 2);
    direction = ((double)width/2 - x) / ((double)width/2) / 2.0;
    sd.addLight(new Light(x, y, direction, angle, colors[c]));
    c = (c + 1) % colors.length;
}
sd.getDisplay().show();
```
SUBMISSION

Submit the files

  Light.java and StageDesigner.java

with your method added by the deadline given above. Place all files for submission in a directory in your cs account. Submit the entire directory with the following command:

  submit102 assign3 MY_DIRECTORY

For example, if you named your directory assignment-03, and your current working directory contained the assignment-03 directory, you would submit with the following command:

  submit102 assign3 assignment-03