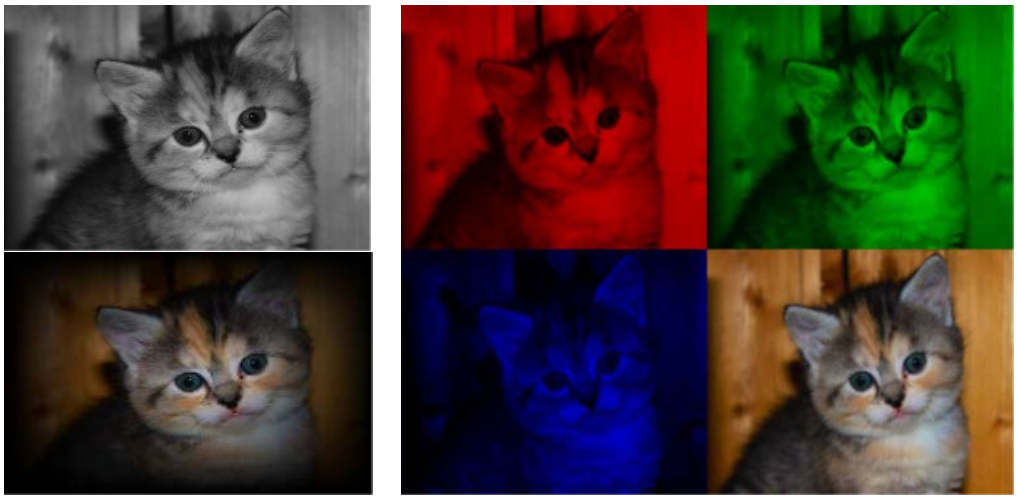
ITSE 2417- Fall 2016
Program Set #4

See Java Grading /Program Guide Sheet for directions and grading/submission information.

1. Write a Java FX GUI application that creates a US Interstate shield. Use the appropriate colors and needed fonts. Choose a legal interstate number and add an appropriate state name. Set the title to `US Interstate`, and center the stage on the screen. Do not use an image. Place all classes into one file.

Name the program: `GUIInterShieldXX.java`, where XX are your initials.

2. Write a Java FX GUI application that transforms a photo of an adorable kitten. The application will give the user the choice to do the following:
   - Change the picture to grayscale. - Gray colors are generated by creating pixels where the red, green, and blue values are equal.
   - Apply an Instagram-like fade effect to the picture. - A fade happens when pixels that are closer to the edges of the image are darkened.
   - Change the color to a particular RGB image. Change the image to either a Red, Green or Blue image. The user will choose which of the three colors to choose.

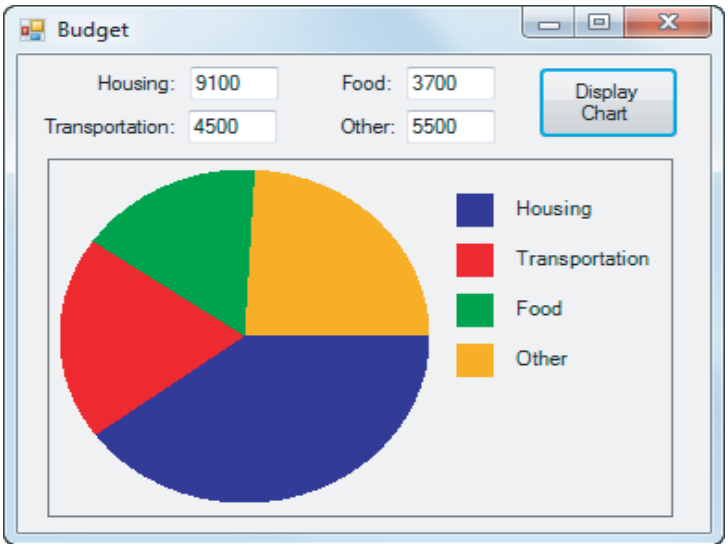


The original picture size will not be changed (`300 x 400`). Each of the three bullets above involve formulas which can be researched off the Web. Use appropriate controls as needed. Set the title to `Cute Kitten`, and center the stage on the screen. Place all classes into one file. Kitten image will be provided by the professor. Do not use an image in the results. Output should be user friendly.

Name the program: `GUIKittenPicXX.java`, where XX are your initials.

3. Write a Java FX GUI application that allows the user to display a budget as a pie chart. After the user enters the values in and clicks on the button, the pie chart should be displayed. Set the title to `Budget`, and center the stage on the screen. Place all classes into one file. Output should look similar to below.

Sample Run:



Name the program: `GUIPieChartXX`, where XX are your initials.

4 (**). Write a Java FX GUI program to emulate Keno games played at casinos. All games offered by casinos count on the fact that you will lose "in the long run". To play Keno, pay some money for each game, typically $1, and then select from 1 to 10 numbers from 1 to 80. The user then selects 20 numbers from the pool of 80. You are paid based on two factors:

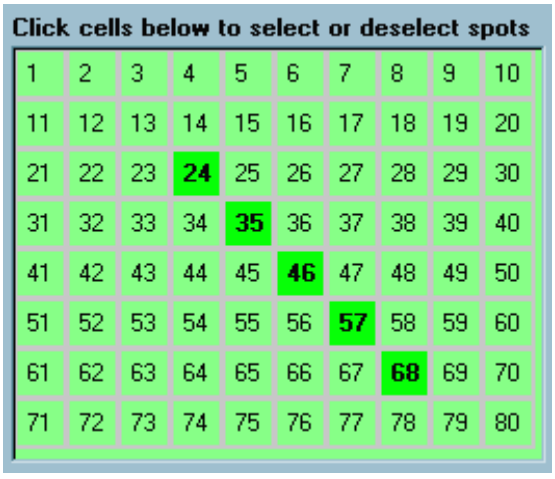
- the number of numbers you selected
- the number that matched the 20 numbers drawn.

The program allows you to play 1 to 10 spot games. You may select the spots on your own or let the program select numbers randomly. Buttons allow you to play a single game, a round of 5 games where the same spots are retained from game to game, 20 5-game rounds to simulate 100 games at a time. The result of each individual game are listed including the payout and total payout for all games since the last time statistics were cleared. A payout table specifies how much is returned to you for each outcome. Display "house percentages" and allow users to define custom payout tables. Some terms used in Keno:

- Spots are the numbers chosen for a game.
- The pool is set of numbers from which spots are selected.
- The draw is the numbers drawn for matching against the spots.
- Catches are the spots which match numbers in the draw.
- The payout or rate table specifies how much money you receive for each winning outcome. If it's not in the table, your payout is zero.
- The "House percentage" identifies how much of the money taken in is paid back as winnings which may range from 5% to 30%.

Set the title to Keno, and center the stage on the screen. Use appropriate controls as needed. Place all classes into one file. The sample run below is a text/GUI-based example. You can use it help test your solution-graphically.

Sample Run: (partial)




```
Game #1
You selected: 24 35 46 57 68

Drawn numbers:
65 53 35 34 29 71 22 23 9 79
7 41 52 66 37 43 77 54 20 15

1 hit(s): 35
Payout: $0.41
Total payout: $0.41
```

Name the program: GUIKenoXX.java, where XX are your initials.