# Python Lucky 7

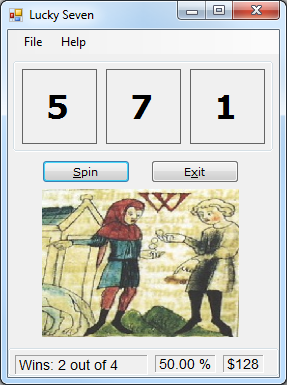
Time required: 90 minutes

Comment each line of code as shown in the tutorials and other code examples.

Follow all directions carefully and accurately. Think of the directions as a minimum requirement.

Create a Python Tkinter application that simulates a one-armed Vegas bandit.

## Create the Form

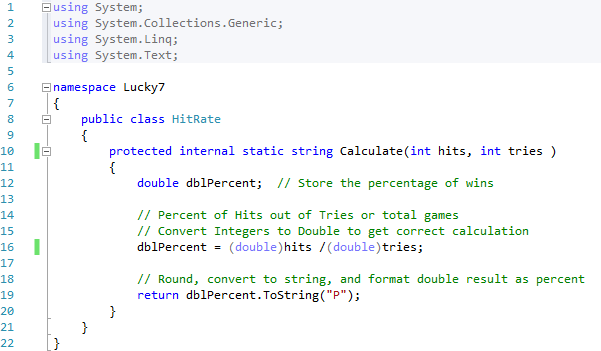
1. Create a new **GroupBox**. We are going to make the groupbox look like a status bar.
2. Set the GroupBox **Dock** property to **Bottom**, delete the **Text** property, and resize it like the sample application design.
3. Click the **Label** control in the Toolbox, and then create a new rectangular label on the GroupBox at the bottom of the form.
4. Set the properties shown in the following table for the new label.

|  |  |
| --- | --- |
| Name | lblWins |
| AutoSize | False |
| BorderStyle | Fixed3D |
| Font | Arial, 10-point |
| Text | Wins: 0 |
| TextAlign | MiddleLeft |

1. Add the other 2 status labels for % and winnings.
2. Use labels with large fonts to display the numbers.
3. Create the rest of the project as shown.
4. Press **F5** to debug and test the program.

## Add a class

1. In the **Project** menu, select the **Add Class** command.
2. Name the module **HitRate.cs.** Click **Add**.
3. In Solution Explorer, Double Click the **HitRate.cs** class and add the following code.



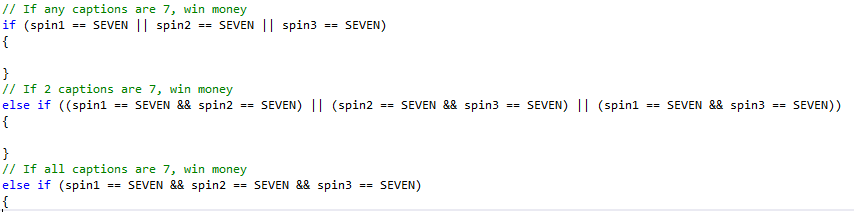
The HitRate function determines the percentage of wins by dividing the hits argument by the tries argument and then adjusts the appearance of the result by using the Format function. The HitRate function is declared as a string because the Format function returns a string value. The Hits and the Tries arguments are placeholders for the two Integer variables that will be passed to the function during the function call. The HitRate function is general-purpose enough to be used with any integer numbers or variables, not only with Wins and Spins. You can re-use it in later programs.

Use the following code to call this class when you need it in the main program.



## Finish the coding

1. Set the players starting money at $20. Set the players bet at $3 per roll.
2. Set the winning amount at $20 when you roll a single 7.
3. Set the winning amount at $100 when you roll two 7’s.
4. Set the winning amount at $200 when you roll three 7’s.
5. Keep track of the money that you have won or lost.
6. Create a single method to spin the dice and call it from the Spin button and the Spin menu item.
7. Create a method for any repeated code.
8. Create a series of If Else If statements like the example using && (and) and || (or) that determines if the player has rolled the winning combinations and add the appropriate amount of money to the Player Money variable.



## Add a Menu and About form

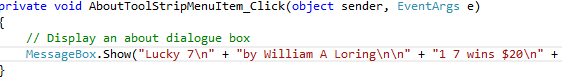
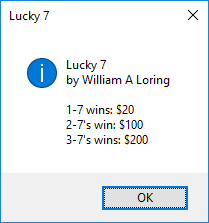
Add a menu to the program with the following elements. Set **CTRL-S** as a shortcut key for the **Spin** command. The About menu item should open a message box with the program name and your name.

File **H**elp

Spin CTRL-S About

Exit

1. Set the **Spin** menu command to the Spin method.
2. Add an **About Box**. Add a MessageBox.Show like the following example. Add the game rules like the example.

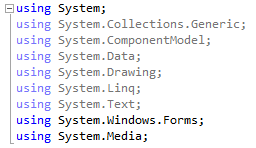
  


1. Disable the Spin button while the Spin code is executing, and re enable it at the end of the spin.
2. Add the following code to add a pause before each number appearing, to simulate the real spinning of a gambling machine.  
   // Global variable  
   int intTimeOut = 250; // Delay in milliseconds, there are 1,000 milliseconds in a second



## Play Sounds

1. At the top of the form code, add **using System.Media;** this allows us to create a SoundPlayer object to play sounds.



1. Right click on the Project name in Solution Explorer and Select **Properties.**
2. Select **Resources** from the left-hand side of the page which is then displayed.
3. Select **Audio** from the first of the horizontal tabs (it may be showing its default choice of ‘Image’.)
4. Click on the **Add Resource** button, which is located next to the tab you have just used.
5. From the File Browser dialog, navigate to the **spin.wav** and **jackpot.wav** file and click OK. (These files are attached to the assignment in Blackboard.)
6. Add the following code to play spin. Play the spin sound when the numbers are rolling.



1. Play the jackpot sounds when the player wins.



## Run the Lucky Seven program

1. Press **F5** to start debugging the program. In a moment the program should run.
2. Click the **Spin** button.
3. Comment each line of code.
4. Test your finished project. Make corrections as necessary.
5. Zip up the project folder and submit it to Blackboard.

## Extra Credit

You may want to add some extras to this game.

* Add a loop when the player wins to blink the image on and off 10 times.
* Replace the labels with picture boxes with images of numbers.
* Replace the image of the 2 men with the coins with another image.
* Make the program look more like a Las Vegas slot machine.