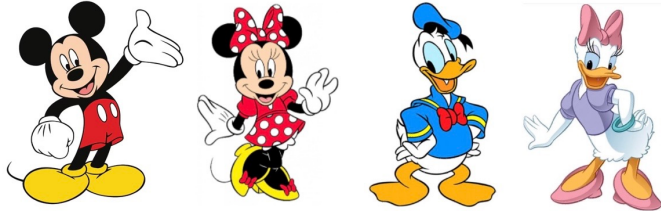


CSE1341 - Lab 5 Assignment

Overview

The Slot Machine is a popular casino game that comes in a lot of shapes and sizes, but all have the same basic functionality: Multiple wheels are spun simultaneously, and then stop randomly. When all wheels have stopped, the prize value (if any) is based on the number of matching wheel values. Each of **three** wheels will contain the **four** characters Mickey, Minnie, Donald, and Daisy.



You will create three incremental versions of the game called Slots1, Slots2 and Slots3.

After you complete Slots1, copy your code into Slots2, then add features to it. You'll do the same by copying your Slots2 code as your starting point for Slots3.

Read the instructions for each program carefully, and make sure the format of your output matches the sample output in the instructions.

Also note that portions of the solution for this lab will be reused in Lab 6.



NOTES:

Each program should include comments that explain what each block of code is doing. Additionally, the programs should compile without errors, and run with the results described in the exercise. The following deductions will be made from each exercise if any of the following is incorrect or missing:

- Proper formatting [5 points]
- Proper names for classes and variables [5 points]
- Comments [5 points per class]
- Program doesn't compile [10 points]
- Source code (java file) missing [10 points]
- Missing array where an array was required [5 points each]
- Missing loop where a loop was required [5 points each]
- Missing class from the design provided [10 points each]
- Missing method from the design provided [5 points each]

This Lab is due Saturday October 29 at 6:00am.

Pre-Lab (5 Points)

Create the shell programs for Slots1.java with the method signatures and comments provided in the instructions. Bring to your lab class for credit. Of course, if you work ahead these can include whatever code you have written so far.

Version 1 (30 Points)

Create a Java solution for the game Slots1. In this incomplete version of the game, player simply spins until choosing to quit. The spun values will be displayed after each spin, as shown in the sample output on the right →

Your solution should have the following two methods:

main method

Create a loop which does the following:

- Prompt user whether to Spin or Quit

- If Q, exit the loop

- if S, call the spinWheel method three times

- Each time spinWheel is called, a String is returned
- Save those returned String values in three String variables
- Print the three spin values

spinWheel method

Create a String array using only within this method that contains the four Strings "Mickey", "Minnie", "Donald" and "Daisy".

Generate a random number which represents the index location of one of the 4 Strings in the array.

Return that String from the array to the main method.

Sample Output:

Slots1

>java Slots1

```
Spin/Quit (S/Q) : s
Mickey - Minnie - Daisy
Spin/Quit (S/Q) : s
Donald - Mickey - Minnie
Spin/Quit (S/Q) : s
Donald - Donald - Minnie
Spin/Quit (S/Q) : s
Mickey - Donald - Mickey
Spin/Quit (S/Q) : s
Minnie - Minnie - Minnie
Spin/Quit (S/Q) : q
```

Hint: see equalsIgnoreCase method in the String class

Version 2 (35 Points)

Create a Java solution for the game Slots2. Start by copying your working code from Slots1 as the starting point for this version.

Revision of Slots1 with the following changes:

Add a method called `checkWinner` which accepts three String values as parameters, and compares them. This method should return a double with the prize amount of the spin.

If all three Strings are the same, print "Won \$1" and return 1.0 to the *main* method.

If two of the Strings match, print "Won 50 cents" and return 0.5 to the *main* method.

Otherwise, print "No prize" and return 0.0 to the *main* method.

In the main method, after the statement that displays the three spun values, add a line that calls the *checkWinner* method, passing the three String values from calling *spinWheel*. The number (double) returned by *checkWinner* will be needed in version 3.

Sample Output: *Slots2*

```
>java Slots2
Spin/Quit (S/Q) : s
Mickey - Daisy - Donald
No prize
Spin/Quit (S/Q) : s
Mickey - Mickey - Donald
Won 50 cents
Spin/Quit (S/Q) : s
Minnie - Daisy - Donald
No prize
Spin/Quit (S/Q) : s
Daisy - Mickey - Donald
No prize
Spin/Quit (S/Q) : s
Donald - Mickey - Minnie
No prize
Spin/Quit (S/Q) : s
Donald - Mickey - Donald
Won 50 cents
Spin/Quit (S/Q) : q
```

Version 3 (30 Points)

Create a Java solution for the game Slots3. Start by copying your working code from Slots2 as the starting point for this version.

Your solution should have three methods: *main*, *checkWinner*, and *spinWheel*

Revision of Slots2 with the following changes:

In the main method:

Add a double variable to store the player's money.

Prompt the user for how much money to insert when starting the game, and store that value in the money variable.

Each time the player chooses to Spin, subtract 25 cents to pay for the spin, then add back any amount the player won using the value returned by the *checkWinner* method.

Add print statements to match the sample output.

Sample Output: *Slots3*

```
>java Slots3
How much money would you like to insert? 100
Spin/Quit (S/Q) : s
Paid 0.25 to spin....
Daisy - Mickey - Mickey
Won 50 cents
Now have $ 100.25
Spin/Quit (S/Q) : s
Paid 0.25 to spin....
Daisy - Minnie - Daisy
Won 50 cents
Now have $ 100.50
Spin/Quit (S/Q) : s
Paid 0.25 to spin....
Minnie - Minnie - Minnie
Won $1
Now have $ 101.25
Spin/Quit (S/Q) : s
Paid 0.25 to spin....
Minnie - Mickey - Donald
No prize
Now have $ 101.00
Spin/Quit (S/Q) : s
Paid 0.25 to spin....
Minnie - Daisy - Mickey
No prize
Now have $ 100.75
Spin/Quit (S/Q) : q
```

