

CS150 Assignment 8

Powerball!

Date assigned: Friday, November 18, 2011

Date due: Monday, December 5, 2011, 9:15am (45 points)

Powerball® is a combined large jackpot game and a cash game. Every Wednesday and Saturday night at 10:59 p.m. Eastern Time, five white balls are drawn out of a drum with 59 balls and one red ball is drawn out of a drum with 39 red balls.

Each ticket costs \$1 and players win by matching one of the 9 ways to win listed here:

Match	Prize
5 white balls + red ball	Grand Prize (maximum \$25 million)
5 white balls	\$200,000
4 white balls + red ball	\$10,000
4 white balls	\$100
3 white balls + red ball	\$100
3 white balls	\$7
2 white balls + red ball	\$7
1 white ball + red ball	\$4
red ball	\$3

Notes:

- White balls are matched in any order. For example, if the user selects 34, 6, 15, 43, 2 and the draw is 2, 19, 34, 9, 15, then they have matched three white balls (34, 15, and 2) and win \$7.
- White balls range in value from 1-59. You must validate this both for user input and for randomly generated numbers.
- Red balls range in value from 1-39. You must validate this both for user input and for randomly generated numbers.
- Numbers are never repeated in a single draw. You must verify that the user doesn't enter the same number twice and that the same number is not randomly generated for the draw.

You are to write a program to simulate the Powerball game. You will ask the user to select their chosen lottery numbers (5 white and 1 red). You will then randomly create a draw by using a pseudo random number generator and display the user's numbers and the numbers drawn. Then, you will display a message indicating if the user has won and add the amount of money won to the user's money pot. Each user starts with \$100 in their pot and can play as many times as they like until they run out of money.

Sample Run

Screen 1:

```
*****
* The Lottery Game
*****
You have $100.00 remaining

Enter 6 white ball numbers
-----
White Ball Number 1: 12
White Ball Number 2: 25
White Ball Number 3: 14
White Ball Number 4: 35
White Ball Number 5: 42

Enter powerball number
-----
Red Ball Number: 11

Press any key to continue . . .
```

Screen 2:

```
*****
* The Lottery Game
*****

User Numbers
-----
----
| 12 | 25 | 14 | 35 | 42 |
----

Power Ball
-----
----
| 11 |
----

Press any key to continue . . .
```

Screen 3:

```
*****
* The Lottery Game
*****

User Numbers
-----
-----
| 12 | 25 | 14 | 35 | 42 |
-----

Power Ball
-----
-----
| 11 |
-----

Press any key to continue . . .

Lottery Numbers
-----
-----
| 2 | 3 | 16 | 13 | 58 |
-----

Power Ball
-----
-----
| 31 |
-----

You got 0 white numbers correct.
You have won nothing. Better luck next time.
You have $99.00 remaining

Play again [y,n]? y
```

Screen 4:

```
*****  
* The Lottery Game  
*****  
You have $99.00 remaining  
  
Enter 6 white ball numbers  
-----  
White Ball Number 1: 1  
White Ball Number 2: 1  
White Ball Number 2: 65  
White Ball Number 2: 2  
White Ball Number 3: 3  
White Ball Number 4: 4  
White Ball Number 5: 5  
  
Enter powerball number  
-----  
Red Ball Number: 6  
  
Press any key to continue . . .
```

Screen 5:

```
*****
* The Lottery Game
*****
```

User Numbers

```
-----
-----
| 1 | 2 | 3 | 4 | 5 |
-----
```

Power Ball

```
-----
----
| 6 |
----
```

Press any key to continue . . .

Lottery Numbers

```
-----
-----
| 16 | 5 | 47 | 49 | 19 |
-----
```

Power Ball

```
-----
----
| 21 |
----
```

You got 1 white numbers correct.
You have won nothing. Better luck next time.
You have \$98.00 remaining

Play again [y,n]? **n**

You must use the following functions:

- `void displayHeading(string heading);`

Prints the **heading** passed to the function, including the border of stars, to the screen.

- `void getUserNumbers(int userNumbers[], int numWhiteBalls, int maxWhiteValue, int maxRedValue);`

Reads in **numWhiteBalls** white ball numbers and one red ball number from the user and places them in the array **userNumbers**. **numWhiteBalls** is the number of white ball numbers to be read in (5 in this case), **maxWhiteValue** is the maximum value for the white balls (59 in this case), and **maxRedValue** is the maximum value for the red ball (39 in this case).

- `void drawLotteryNumbers(int drawnNumbers[], int numWhiteBalls, int maxWhiteValue, int maxRedValue);`

Randomly generates **numWhiteBalls** white ball numbers and one red ball number and places them in the array **drawnNumbers**. **numWhiteBalls** is the number of white ball numbers to be read in (5 in this case), **maxWhiteValue** is the maximum value for the white ball (59 in this case), and **maxRedValue** is the maximum value for the red ball (39 in this case).

- `void displayLotteryNumbers(int drawnNumbers[], int numWhiteBalls, string heading);`

Prints to the screen the **heading** followed by dashes followed by the values of **numWhiteBalls** white balls that are stored in **drawnNumbers** and the value of the one red ball. Your output must look like the one on page 2

- `void calculateMatches(int userNumbers[], int drawnNumbers[], int numWhiteBalls, int &numWhiteMatches, int &numRedMatches);`

Compares the values in **userNumbers** with the values in **drawnNumbers**. **numWhiteMatches** will contain the number of white balls that matched and **numRedMatches** will contain the number of red balls that matched.

- `void calculateWinnings(int numWhiteMatches, int numRedMatches, double & pot);`

Calculates the winnings based on the **numWhiteMatches** and **numRedMatches** as described on page 1. The winnings are added to the user's pot of money.

You may create additional functions:

You might find it's useful to create additional functions other than those listed above to help further break up the program. For instance, you see that you must get six different user numbers, but you might find it's easier to get a single user number and call that function repeatedly. Further, you need to generate a random number in various ranges, so another function might be useful. Feel free to add additional functions as you see necessary. Smaller and simpler functions will aid in program debugging and understanding.

Remember

1. Do not code and test more than one function at a time.
2. The debugger is your friend. If you need help, please see me.

Question

1. How will we test your program to make sure that your program handles all of the winning possibilities?

To complete this assignment you must submit the following:

1. An electronic copy of your program on Turing

- a. Add a new project named 08_Powerball to your previously created assignment solution called PUNetIDAssignments. It is *vital* that you name your project correctly!
- b. Type your program (fully documented/commented) into the project. You must follow the coding standards!
- c. Pay attention to the example output! Your program's output must look **exactly** like the example output! The spacing and newlines in your output must match exactly.
- d. Make sure that your program compiles and runs correctly. If you get any errors, double check that you typed everything correctly.
- e. Make sure that your program does not produce any warnings.
- f. Once you are sure that the program works correctly it is time to submit your program. You do this by logging on to Turing and placing your complete solution folder in the **CS150-02 Drop** folder. This solution folder must contain eight projects: 01_Fraction, 02_Currency, 03_Bank, 04_Retirement, 05_Hotel, 06_Craps, 07_QuizGrader, 08_Powerball.
- g. The program must be in the drop folder by 9:15am on the day that it is due. Anything submitted after that will be considered late.

2. A hard copy of your program

- a. The hard copy must be placed on the instructor's desk by 9:15am on the day that it is due.
- b. The hard copy must be printed in color, double-sided, and stapled if necessary.
- c. Your tab size must be set to 2 and you must not go past column 80 in your output.

3. THERE IS NO GIFT TO BE USED ON THIS LAST ASSIGNMENT!!!!!!

Good luck! And remember, if you have any problems, come and see straight away. ☺