

Lab #4: Powerball

Overview: Powerball is a game where players buy tickets that consist of five unique lottery numbers that range between 1 - 69 and one number between 1-26, called the Powerball. The lottery numbers must be unique meaning they cannot be repeated, whereas the Powerball may be the same as one of the five lottery numbers. A player's set of chosen numbers are then compared to the winning set which have been randomly chosen.

Assignment: Write a well-documented Java program that will simulate a Powerball lottery. Your program will automatically generate a winning Powerball ticket at the beginning of each game. A player will start with \$1,000.00 and can purchase any number of \$2 Powerball tickets until they decide to quit, or they run out of money. You must compare each lottery number on the player's ticket to every lottery number on the winning ticket – sequence is not important. Lastly, you will compare the winning Powerball number to the Powerball number on the player's ticket.

There are a couple details that your program must follow, such as:

- Decompose your functions into smaller and smaller functions. If you have a lot of code in main, or in another function, the program will receive -20%.
- You must sort, in ascending order, both the player's numbers and the winning numbers before printing them out. Sort the numbers using a sorting algorithm from our Java Textbook. **DO NOT USE ARRAY.SORT or any other array library function!!!**
- Print the winning Powerball number in red. (This may not work in Eclipse. Use Dr. Java to test this.)
- At the end of a drawing, show matching numbers in red, and how much a ticket won.
- Add the winning dollar amount to a player's total.
- A player's list of tickets must line up right-justified as shown in the below example. (Only use *System.out.print()* and *System.out.println()* commands, **NO** *System.out.printf()* commands!)
- Required constants include (type int):
 - The total amount of numbers on a ticket (this includes both the lottery numbers and the Powerball number).
 - The cost of one ticket.
 - Initial amount of money given to a player at the beginning of a game.
 - The cash amount given by each of the different winning combination.

Supplied Classes: This class will be provided to you for your convenience. You must write your own functions that extend these and serve a specific purpose.

- IR4: Contains functions that will ask users for input and returns a value such as integer, String, double, etc. Look in this program and see what it does!

Output: Here is some sample output from your program, provided so that you can see the expected formatting:

```
*****
```

```
Let's play Powerball!
```

```
*****
```

```
5 numbers correct plus powerball = $100,000,000
5 numbers correct, no powerball = $5,000,000
4 numbers correct plus powerball = $50,000
4 numbers correct, no powerball = $100
3 numbers correct plus powerball = $100
3 numbers correct, no powerball = $7
2 numbers correct plus powerball = $7
1 number correct plus powerball = $4
0 numbers correct plus powerball = $4
```

```
*****
```

```
You have $1000
```

```
How many $2 lottery cards do you want to purchase?
```

```
5
```

```
The winning lottery numbers, followed by the Powerball Number, are:
```

```
8 22 42 46 55 25
```

```
Your lottery ticket numbers, followed by the Powerball Number, are:
```

```
1 7 38 50 61 12
```

```
22 46 47 51 53 25 You won $7
```

```
6 34 57 60 63 8
```

```
3 9 34 44 49 25 You won $4
```

```
10 21 58 59 62 2
```

```
*****
```

```
Let's play Powerball!
```

```
*****
```

```
5 numbers correct plus powerball = $100,000,000
5 numbers correct, no powerball = $5,000,000
4 numbers correct plus powerball = $50,000
4 numbers correct, no powerball = $100
3 numbers correct plus powerball = $100
3 numbers correct, no powerball = $7
2 numbers correct plus powerball = $7
1 number correct plus powerball = $4
0 numbers correct plus powerball = $4
```

```
*****
```

```
You have $1001
```

```
How many $2 lottery cards do you want to purchase?
```

```
0
```

```
You have $1001
```

```
Goodbye!
```

Notice the blank lines to make it easier for the user to read!

Note the numbers are sorted and right justified.

Notice the red text. This will work in Dr. Java, but it may not work in a different IDE such as Eclipse. Test your code with Dr. Java before turning it in!

Hand In: You are required to submit your completed lab as a compressed .zip file which includes:

- Your completed Java source code file (*do not submit the .class file!*)
- Your test plan file that contains an appropriate number of test cases and all possible types of input and output. (*I expect that your test cases will document any possible invalid input that might be entered by the user – I WILL check your work!.*)

Other Requirements and Hints:

- **Start Early!** You will be writing a fair amount of code for this assignment! This usually means there will be lots of mystery Java exceptions to figure out. Trying to write code at 9 p.m. on the due date is a lousy time to be stuck. Remember that tutors will not hand you the answers—you have to come up with solutions on your own!
- Use System.out.println statements in sections of your code to help you identify errors if you are not getting the behavior you expect.

If you don't turn this in, or if you turn in the assignment and it doesn't work at an acceptable level, you will receive a zero on the assignment. Can your program be used by the state of Arizona? If not, it earns a zero.

Verify your output looks almost EXACTLY like the sample output shown above. Blank lines matter, as does spelling.

This lab should demonstrate your ability to solve a more difficult problem with code.

This problem will be easier to solve if you break up the problem into small modules and functions. Identify WHAT must be done before trying to code it. If you can't say what must be done, you can't code it. Try telling a friend what the program must do and have them make notes for you.