

CSc 110 Assignment 4 Focusing on If / Else, While Loops, and Methods

A Game of Chance:

“Chuck-a-luck”



Reminders:

Submit your java file name EXACTLY as specified (ChuckALuck.java) using the 'Assignments' link of the course connex site. ** Don't forget to follow the Style Guidelines posted – this is part of the requirements of every assignment.

It is OK to talk about your assignment with your classmates, and you are encouraged to design solutions together, but each student must implement their own solution. We will be using plagiarism detection software on your assignment submissions.

Learning Outcomes:

When you have completed this assignment, you should understand:

- How to implement conditional statements (if-statements and if/else-statements)
- How to compare Strings.
- How to analyze the flow of choices to ensure correct program logic.
- How to design a suite of tests that will check the various cases in a branching program (i.e. one that contains conditional statements).
- Logical decomposition using methods.
- How to design and use indefinite loops.
- How to generate random numbers.

Assignment Requirements:

This assignment requires you to implement the game of “Chuck-a-luck”.

In order for your assignment to be graded, we **require** you to use the following code for your main method...

```
public static void main(String args[]) {  
    Scanner scan = new Scanner(System.in);  
    Random rand = new Random(System.currentTimeMillis());  
    play(scan, rand);  
}
```

And your implementation **must** include the following method EXACTLY as shown...

```

/*
    PURPOSE:  to play the chuck a luck game
    INPUT:    Scanner scanner, to collect input from the user for the
              the bet amount and if they want to continue playing
              Random randomObject, to simulate dice roll
    OUTPUT:   prompts to ask user for bet and updates of bankroll
*/
public static void play(Scanner scanner, Random randomObject)

```

You **must** decompose your program into multiple methods, but you are free to choose what those methods look like. Remember, if a method you are writing needs a Scanner or a Random object you can pass the ones you have in your play method to that next method. For example..

```

public static void play(Scanner scanner, Random randomObject) {
    myMethod1(scanner);
    myMethod2(randomObject);
}

public static void myMethod1(Scanner scan) {
    String s = scan.next();
}
public static void myMethod2(Random rand) {
    int num = rand.nextInt(10);
}

```

Programming Problem Description:

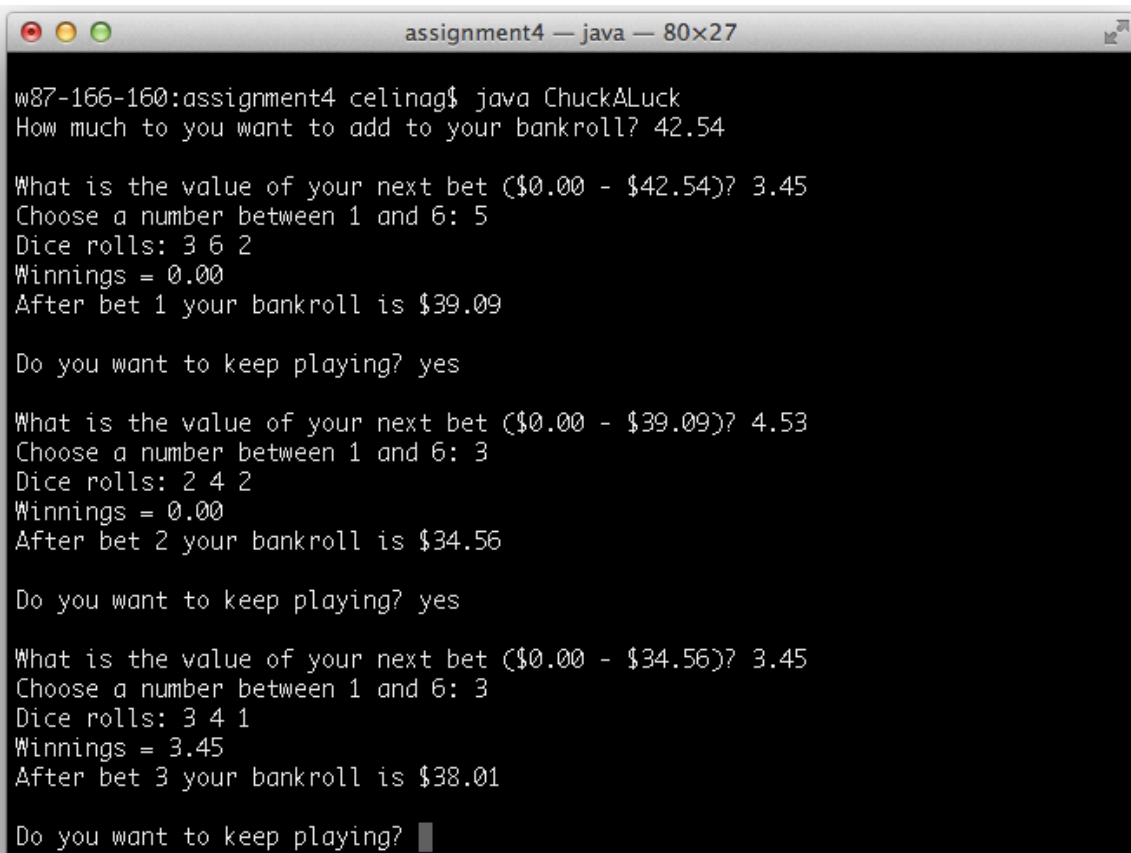
Rules for Chuck-a-luck are available here: <http://en.wikipedia.org/wiki/Chuck-a-luck>. Your version of Chuck-a-luck only needs to implement a *Single Dice Bet* and must perform the following steps:

- 1) Ask the player to input how much money they want to add to start their bankroll (how much money they have in total to play with).
- 2) Take a bet from the player via the command line (again, bets are decimal values). The player will input a number between 1 and 6 and the amount of money they want to bet. NOTE: you must ensure they pick a valid number and they can't bet more money than is available in their bankroll.
- 3) Roll the three dice, determine how many of the dice match the player's number and then calculate the player's winnings. NOTE: Your "dice" are implemented with a random number generator that simulates a 6-sided die.
 - a. To calculate winnings, follow the *Single Dice Bet* odds:
 - i. 1 dice matching player gets 1:1 odds on their wager
 - ii. 2 dice matching player gets 2:1 odds on their wager
 - iii. 3 dice matching player gets 10:1 odds on their wager
 - iv. 0 dice matching player loses their wager on their wager
- 4) Your game must keep track of and print out: a) how many bets the player made and b) the player's bankroll after each bet (this should print out as decimal value with two significant figures (ie. \$5.67) HINT: check out String.format (<http://stackoverflow.com/questions/7548841/round-a-double-to-3-significant-figures>))
- 5) The game should continue to allow the player to bet if: a) they still want to and b) they still have money in their bankroll. '

The program (ChuckALuck.java) must...

- Use the *Random* class to simulate a roll of the die (i.e. a random number between 1 and 6). See the text to learn how to use this class.
- Use *if* statements and *while* loop. Other loops and condition statements can be used as needed as well.
- Provide informative prompts when user input is required. The user will input “y” or “n” (or some similar String values) for whether or not to bet again.
- Give informative output, such as: the number rolled on each die, the amount of money won/lost at the end of each turn, the player’s bankroll total at the end of each turn, and if the player is up and by how much OR if the player is down and by how much.
- Be logically decomposed into smaller parts, using methods.
- Be well documented, following the style guidelines.

Sample Run



```
w87-166-160:assignment4 celinag$ java ChuckALuck
How much to you want to add to your bankroll? 42.54

What is the value of your next bet ($0.00 - $42.54)? 3.45
Choose a number between 1 and 6: 5
Dice rolls: 3 6 2
Winnings = 0.00
After bet 1 your bankroll is $39.09

Do you want to keep playing? yes

What is the value of your next bet ($0.00 - $39.09)? 4.53
Choose a number between 1 and 6: 3
Dice rolls: 2 4 2
Winnings = 0.00
After bet 2 your bankroll is $34.56

Do you want to keep playing? yes

What is the value of your next bet ($0.00 - $34.56)? 3.45
Choose a number between 1 and 6: 3
Dice rolls: 3 4 1
Winnings = 3.45
After bet 3 your bankroll is $38.01

Do you want to keep playing? █
```