

P2 Easy Win Dice Game

- Due Oct 5, 2017 by 10am
- Points 100
- Available Sep 27, 2017 at 12am - Dec 1, 2017 at 11:59pm 2 months

This assignment was locked Dec 1, 2017 at 11:59pm.

Programming Assignment #2

The Easy Win Dice Game.

Program Description: Your assignment is to write the Easy Win Dice Game program. The game is played with three 6 sided dice. Each time a person wants to roll the dice they must first pay one dollar. Depending on the outcome of the dice rolls, the person can win some money. Here are the outcomes and the amount that they win.

Dice Results	Wins
All three dice are equal to 1	\$ 20
All three dice are equal to each other (but not equal to 1)	\$ 10
Any two of the three dice are equal to each other	\$ 2
All other results	0

Start the person with \$20 and let them continue as long as they have money and as long as they wish to play again. You should use a constant variable for the win amounts. The name of a constant variable should be all caps; for example: `const int BIG_WIN = 20;`

[Example Program](#)

Extra Odds and Ends:

To get the program to keep allowing the user to roll the dice many times, you will need to use a while loop that will continue as long as the user has money or until they want to quit.

Testing your program: Since the real program will be using random numbers for the dice rolls, it will be difficult to test the correct results for each situation. To solve this problem put in some user prompts (just for testing) to be able to set the dice rolls from the user. In this way you can test your winning conditions without needing to wait to randomly hit the numbers that you need to use to test your program.

When you are done testing your program, comment out the lines used to let the user set the dice roll values, but leave them in the program when you turn in your program.

Example of program in testing mode

Hints: Take some time to think about the program and how you can set it up. Then write the program a piece at a time. Write a few lines of code, and then test those lines of code. Use cout's to tell you about what is happening in the program. Ask me if you have questions...about the requirements or when programming.

Turn in: A paper copy of the source files (cpp) with the testing code commented out (using // so the compiler will ignore them). Also turn in an example run in test mode, and in random mode. For the test run, **you must use the following dice rolls in this order to test your program:**

Dice rolls	Tests
1 1 1	Big win
2 2 2	Medium win
1 2 2	Small win where die2 == die3
2 4 2	Small win where die1 == die3
5 5 1	Small win where die1 == die2
1 2 3	no win

For the output from the random run, have the game for at least 8 rolls.

Grading: I will be grading your program on the following:

- Did you follow the style shown in the class and book (indenting inside of a block)?
- Did you include the Program Header Comments as shown below?
- Does your program include the commented out lines that you used to allow the user to enter their dice rolls during testing?
- Did your output have good spacing?
- Did you use descriptive names for your variables?
- Did you use all capital letters for constant variables (yes, you should use constant variables for the win amounts)?
- Did you get the right output?
- Did you use the input I told you to use?
- Did you avoid line wrap (when a line is too long and will wrapped around to the next line when printed)?
- Did you staple all your papers together?

Comments: Comments are a way of documenting a program (explaining who did what and how). All programs for the rest of the course are required to have the following program header documentation and inline documentation to explain any tricky pieces of code.

```
////  
// Name: Bob Programmer  
// Section: A, B, or S  
// Program Name: Hello World  
//  
// Description: A brief description of the program. What does the  
// program do (not how it does it: for example, it uses loops)? Does  
// the program get input? What kind? What information is output  
// from the program and to where (screen or file)  
////  
  
#include <...>  
  
.....the rest of the program
```