

Sangyun Lee

[Leesa6@oregonstate.edu](mailto:Leesa6@oregonstate.edu)

Oct 16 2019

## Lab 3

### Requirements

- 2 players
- 2 types of die
- Rolling dice, and score the game
- Using inheritance
- 3 class

### Challenges

- Input validation & Menu for games
- Base class and derived class
- Generating the higher avg number for loaded die

### How to solve the challenges

- Menu for games & Input validation
  - Reuse the function I created from the last project, which is display menu and validate the input value with getInput.cpp.
- Base class and derived class
  - Since the base class doesn't have complex elements and function, It will be easier to set the class relationship
- Control output for loaded die
  - Since one function should generated different result for different class object, I will overload the function on base class

## Test table

### 1. game::set function with Menu() & getInput()

Since the Menu() & getInput() is validated from the project 1, I tested game::set func whether it works well with the two functions or not.

Test Case	Input	Driver Func	Expected Output	Actual Output
Wrong	'a'	Menu() & getInput()	"your input is wrong...."	"your input is wrong...."
Correct	1	Menu() & getInput()	Return 1, and "welcome to..."	Return 1, and "welcome to..."
Correct	2	Menu() & getInput()	Return 2, and close the program	Return 2, and close the program

### 2. Die::setSide() & Die::getSide() for both die class object loadedDie class object

Since the menu() & getInput() works correctly, I am assuming the all inputs for test case are correct data. Additionally, I tested loadedDie class object more, because it uses parent class function.

Test Case	Input	Driver Func	Expected Output	Actual Output
Die class Object	10	.setSide() & .getSide()	10	10
loadedDie object	7	.setSide() & .getSide()	7	7
loadedDie object	6	.setSide() & .getSide()	6	6

### 3. Die::randNum() && loadedDie::randNum()

I would like to focus on overload function. Overload function have few conditions

If side <= 4, lowest rolls will be 2

If 5 <= side <= 6, lowest rolls will be 3

If  $7 \leq \text{side} \leq 8$ , lowest rolls will be 4

If  $\text{side} \geq 9$ , lowest rolls will be 5

Test Case	preset	Driver Func	Expected Output	Actual Output
Die class Object correct	setSide(10)	Die::randNum()	Random num Between 1 and its side	7
loadedDie object correct	setSide(9)	loadedDie::randnum()	Random num Between 5 and its side	5
loadedDie object	setSide(9)	loadedDie::randnum()	Random num Between 5 and its side	5

## Reflection

It was easy to build the inheritance class because requirements are not too complex, however it was kind tricky to have such a function that generate different output for different class object. For reality, I was planning to create overload func only when the reg dice is leading the game. So, I tried make 'game class' as base class of 'die class', so die class object can access to the game class variable. However, it didn't work as I expected. And it got more complex as I work on code modification. Hence, after few tries, I ended up with the simply overload function that the lowest side of loaded dice is always higher than the reg dice.

Additionally, at the beginning, I set the base side of the loadedDie class object is one larger than die class object, it looks okay when the side of dices were under 5. Once the side of dices are higher than 5, my original overload func was useless. Therefore, I set up different range of base side of loaded dice based on the side of the dice.