# CECS 277 – Lab 3 – Lists

## Ship, Captain, and Crew!

Create a program that allows the user to play the two-player dice game Ship, Captain, and Crew. Each player rolls a set of 5 six-sided dice up to three times to try to first get a 6 (the ship), then a 5 (the captain), and a 4 (the crew). Once they are obtained, then the value of the other two dice are the cargo, which are the player's points. As you obtain the 6, 5, and 4, the dice are set aside and not rolled again. These dice must be taken in order. For example, if a 5 is rolled, it cannot be taken unless a 6 has already been kept. The cargo points cannot be scored unless a 6, 5, and 4 have been kept. If a player has extra rolls after obtaining the ship, captain, and crew, then they can either stop or they can try for a higher cargo roll, but both dice must be rolled. However, the player must keep the new score, even if it is lower than their previous cargo value.

Create the following functions for your program:

1. `roll_dice(dice)` – Passes in a list of dice. Randomize the values of each of the dice 1-6. Sort the values and place them in descending order.
2. `display_dice(name, dice)` – Passes in the name of the list, and the list of dice. Display the dice values separated by spaces.
3. `find_winner(player_points)` – Passes in the two player's points as a list. Displays the two player's points values, and displays who won. If the scores are the same, then display that it was a tie.

In your main function, create lists to store the two player's scores, the dice to keep, and the dice to roll. Create a loop that repeats twice, once for the first player and then again for the second player. Create another loop that repeats three times, one for each of the player's rolls. Roll and display the dice, then use if statements to check to see if the user rolled a 6, 5, or 4, if they have, then add them to the keep list, but only if they have already kept the previous values. If 6, 5, and 4 have been obtained, sum the final two dice in the roll list to calculate the cargo score. Prompt the user to roll again until they like their cargo score or until they run out of rolls. Record the player's score in the score list. After both player's turns have been taken, call find_winner to display both player's points and the winner.

**Example Output** (user input is in italics)**:**

```
- Ship, Captain, and Crew! -

Player #1's Turn:
Roll = 6 4 4 3 3
Yo ho ho! Ye secured a ship!
Keep = 6

Roll again? y
Roll = 5 5 4 2
Shiver me timbers!  A Capt'n!
Ye bribed a crew with Grog!
Keep = 6 5 4
Cargo = 5 2

Your cargo points are: 7

Roll again? n
Player #1 points = 7

Player #2's Turn:
Roll = 5 5 4 3 2
Keep =

Roll again? y
Roll = 6 4 2 2 1
Yo ho ho! Ye secured a ship!
Keep = 6
```

```
Roll again? y
Roll = 5 4 2 1
Shiver me timbers!  A Capt'n!
Ye bribed a crew with Grog!
Keep = 6 5 4
Cargo = 2 1
Your cargo points are: 3
Player #2 points = 3

Score:
Player #1 =  7
Player #2 =  3
Player #1 won!

- Ship, Captain, and Crew! -

Player #1's Turn:
Roll = 6 5 4 2 1
Yo ho ho! Ye secured a ship!
Shiver me timbers!  A Capt'n!
Ye bribed a crew with Grog!
Keep = 6 5 4
Cargo = 2 1
Your cargo points are: 3

Roll again? y
Roll = 6 2
Keep = 6 5 4
Cargo = 6 2
Your cargo points are: 8

Roll again? n
Player #1 points = 8

Player #2's Turn:
Roll = 6 4 1 1 1
Yo ho ho! Ye secured a ship!
Keep = 6

Roll again? y
Roll = 6 4 2 2
Keep = 6

Roll again? y
Roll = 6 5 3 3
Shiver me timbers!  A Capt'n!
Keep = 6 5
Player #2 points = 0

Score:
Player #1 =  8
Player #2 =  0
Player #1 won!
```

**Notes:**

1. Place your name, date, and a brief description in a comment block at the top of your program. Add brief comments in your program to describe sections of code.
2. Your code should be defined within the functions. Please do not create extra functions.
3. Use the check_input module provided on Canvas to check the user's input for yes/no.
4. Use the random module to generate your random numbers.
5. Please read through the Coding Standards reference document on Canvas for guidelines on how to name your variables and to format your program.
6. Thoroughly test your program before submitting:
    a. Make sure the game resets the necessary values and lists for the next player.
    b. Make sure the game displays the list of kept values and roll values correctly.
    c. Make sure that the game cannot keep a 5 if the player did not already keep a 6. Similarly, it cannot keep a 4 if it did not already keep a 6 and a 5.
    d. Make sure that the game doesn't keep 6s, 5s, or 4s, if they have already been kept.
    e. Make sure the player gets cargo points if they kept 6, 5, and 4. Similarly, make sure the player does not get any cargo points if they did not keep 6, 5, and 4.
    f. Make sure each player only gets three rolls. Similarly, make sure the player can give up the rest of their rolls by saying 'no' when asked if they want to roll again.
    g. Make sure that the players scores are correctly compared to determine the winner.

**Ship, Captain, and Crew! Rubric – Time estimate: 3 hours**

| **Ship, Captain, and Crew!**<br>**10 points** | Correct.<br><br>2 points | A minor mistake.<br>1.5 points | A few mistakes.<br>1 point | Several mistakes.<br>0.5 points | No attempt.<br>0 points |
|---|---|---|---|---|---|
| **roll_dice and display_dice:**<br>1. Pass in correct parameters.<br>2. Does not return values.<br>3. roll_dice randomizes each of the 5 dice with values 1-6.<br>4. Sorts in descending order.<br>5. display_dice displays the title and set of dice values. | | | | | |
| **find_winner function:**<br>1. Passes in player points list.<br>2. Displays each of the player's points.<br>3. Compares each of the player's points and displays the winner (or tie). | | | | | |
| **Lists:**<br>1. Created player points, keep, and roll lists.<br>2. Checked if 6, 5, 4 values were in keep and roll lists.<br>3. Removed dice from roll list.<br>4. Added dice to keep list.<br>5. Added points to points list. | | | | | |
| **Main Function:**<br>1. Rolls and displays dice.<br>2. Correctly moves dice from roll list to keep list.<br>3. Correctly checks if player has 6, 5, and 4.<br>4. Correctly finds cargo points.<br>5. Prompts to roll again (y/n).<br>6. Repeats 3 times for each player (unless 'no' is chosen).<br>7. Repeats for both players.<br>8. Displays points and winner. | | | | | |
| **Code Formatting:**<br>1. Code is in functions.<br>2. Correct spacing.<br>3. Meaningful variable names.<br>4. No global variables.<br>5. Correctly documented. | | | | | |