**Course: Introduction to Data Science (DS2006) - Laboratory 03**

**Student:**

**Task 1**: In the Battle of the dices, replace the code shown in Figure 1 with the code shown in Figure 2. What happens when you run the code with that modification?

print("Player 1 rolled: ",roll1)

print("Player 2 rolled: ",roll2)

Figure 1 - Code example.

print("Player 1 rolled: " + str(player1\_roll))

print("Player 2 rolled: " + str(player2\_roll))

Figure 2 - Another code example.

**Task 2**: Replace the code shown in Figure 2 with the code shown in Figure 3. What happens when you run the code with that modification?

print("Player 1 rolled: " + player1\_roll)

print("Player 2 rolled: " + player2\_roll)

Figure 3 - Yet another code example.

**Task 3**: Revert back to the code shown in Figure 1 and now replace the section of the code shown in Figure 4 so that it first checks if the 1st player won the round (i.e. got the higher score) if not checks if the 2nd player won the round, else assume the players had a tie.

if player1\_roll > player2\_roll:

print("Player 1 wins this round!")

print("Because ", player1\_roll," is greater than ", player2\_roll)

elif player1\_roll == player2\_roll:

print("Amaaazzinng! This round has a tie!")

else:

print("Player 2 wins this round!")

print("Because ", player2\_roll," is greater than ", player1\_roll)

Figure 4 - Code example used to check who won the round.

**Task 4**: Add variables to get information about who won the current round and then print the result of the round to the screen.

**Task 5**: What happens when you change the way you add to the variables created in Task 4 to use the format **variable += 1** instead of **variable = variable + 1**.

For example:

player1\_wins += 1

instead of

player1\_wins = player1\_wins + 1

**Task 6**: Change the content of the messages shown when a player wins (in Figure 5) to something cooler!

# Now we need to check if either player won.

if player1\_wins == 3:

print("Player 1 is the newest Battle of Dices Champion! ")

elif player2\_wins == 3:

print("Player 2 is the newest Battle of Dices Champion! ")

else:

print("This heated Battle of Dices is still going on! Who will win? ")

Figure 5 - Code example used to check the winner of the game.

**Task 7**: Create a new file called **battle-of-dices-bad.py .** In this file make sure the game runs for as many rounds as needed without using any type of loop-related control structure.

**Task 8**: In **battle-of-dices-bad.py** replace all the calls you are making to roll a D6 to roll a different dice number. Choose between a D4, D8, D12, D20 or D100 and make both players use the same dice you choose in all rounds. I.e. replace all the d6 roll calls to the dice you choose roll calls.

**Task 9**: Reflect and write your impressions about refactoring (i.e. changing the code) in the battle-of-dices.bad.py without using loops.

**Task 10**: Create a new file called **battle-of-dices-better.py** . In this file, refactor the code to use a loop control structure with while.

**Task 11**: Make the necessary changes in the code of **battle-of-dices-better.py** to allow the program to print the number of rounds it took to a player win the game.

**Task 12**: Refactor the dice.py file we created for the last lab to provide **functions** for rolling common dice: d4, d6, d8, d12, d20, d100. Each dice roll should have their own function as illustrated in Figure 6.

**Task 13**: Refactor the code in **battle-of-dices-better.py** to make use of the function calls from your dice class.

**Task 14**: Create a new file called **battle-of-dices-cooler.py** and refactor the code from **battle-of-dices-better.py** to roll 2 dice (of different sizes) for each player. The winner of the round will be decided by the sum of the values obtained for each player.

def rollD6():

"""

Roll a standard 6-sided dice.

Returns:

int: a random number between 1 and 6

"""

return random.randint(1, 6)

Figure 6 - Code example about a function used to roll a D6.