Task completed:

| Date started | Date completed |
| --- | --- |
| **10th June 2019** | 1st November 2019 |

Scenario

Katarina is developing a two-player dice game.

The players roll two 6-sided dice each and get points depending on what they roll. There are 5 rounds in a game. In each round, each player rolls the two dice.

* The points rolled on each player’s dice are added to their score.
* If the total is an even number, an additional 10 points are added to their score.
* If the total is an odd number, 5 points are subtracted from their score.
* If they roll a double, they get to roll one extra die and get the number of points rolled added to their score.
* The score of a player cannot go below 0 at any point.
* The person with the highest score at the end of the 5 rounds wins.
* If both players have the same score at the end of the 5 rounds, they each roll 1 die and whoever gets the highest score wins (this repeats until someone wins).

Only authorised players are allowed to play the game.

Where appropriate, input from the user should be validated.

**Design, develop, test and evaluate a program that:**

1. Allows two players to enter their details, which are then authenticated to ensure that they are authorised players.
2. Allows each player to roll two 6-sided dice.
3. Calculates and outputs the points for each round and each player’s total score.
4. Allows the players to play 5 rounds.
5. If both players have the same score after 5 rounds, allows each player to roll 1 die each until someone wins.
6. Outputs who has won at the end of the 5 rounds.
7. Stores the winner’s score, and their name, in an external file.
8. Displays the score and player name of the top 5 winning scores from the external file.

Analysis

Describe in your own words what the task is asking you to do:

2 player’s w/ Login System (Authorised Users only (ask for input and check against a file/list or just ask for a username/password (check against a list [for loop then if loop]))

User input should be validated to make program robust 🡪 check name = string 🡪 don’t let game progress unless it is valid input

Sum of 2 random integers between 1 and 6 each (1,2,3,4,5,6), save as variable (total), add this number to user’s score (total[User] = dice1user[User] + dice2user[User]). If total = odd, score = score – 5, If total = even, score = score + 10. If a double is rolled (i.e. 6 and 6 or 3 and 3), they get an extra roll of one die, which is added to their score. If score < total, score = 0. There are 5 total rounds (FOR Loop?), the winner is the one who gets the highest score. If both players are equal, role a dice each until there is a winner. Save the winners name and score into a file, and display a scoreboard when prompted (main menu?)

Try and create 3 or more key success criteria for your program.

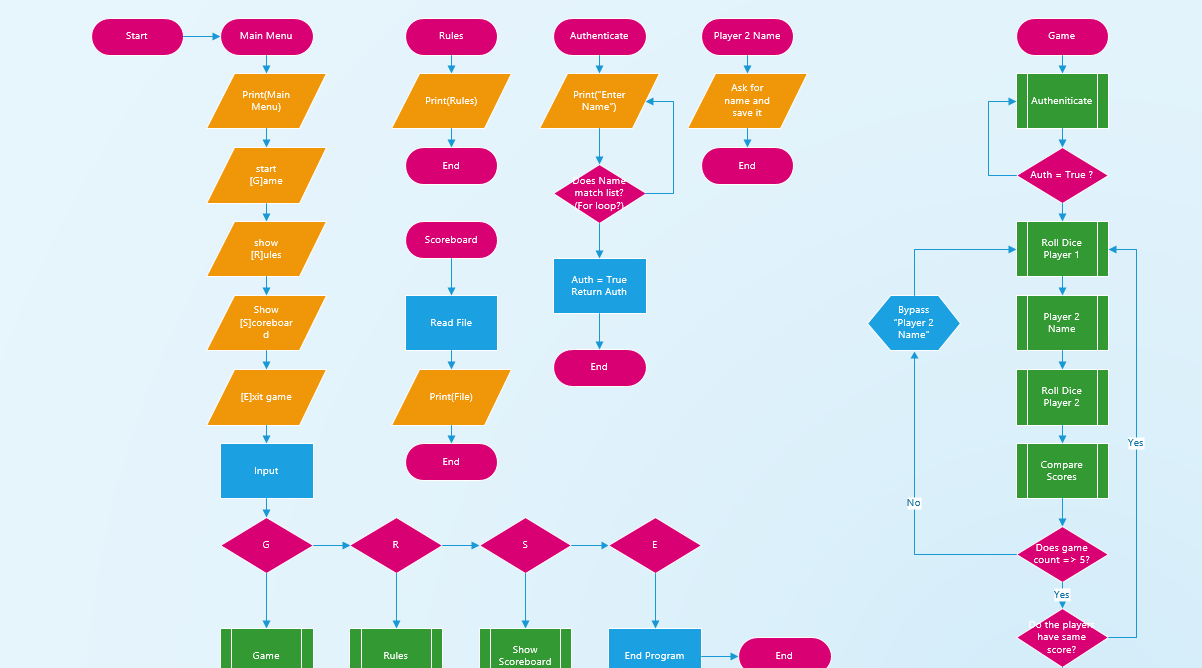
**Success Criteria:**

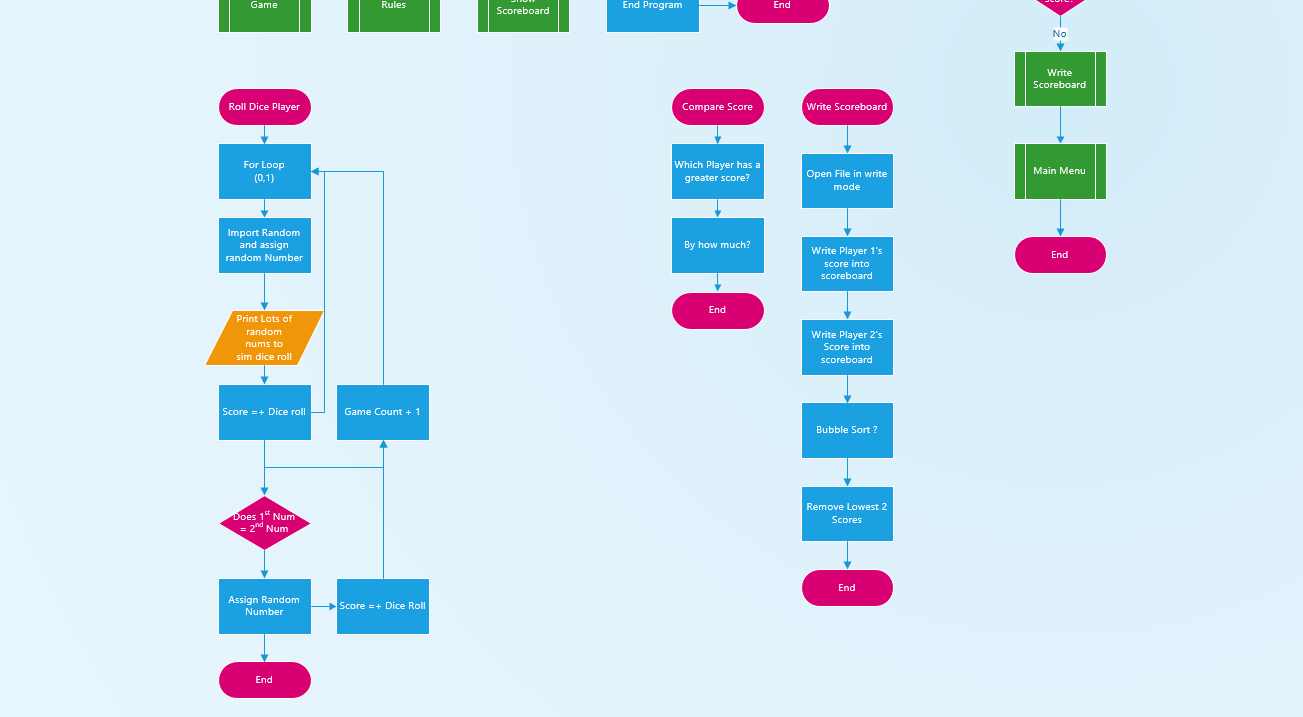
1. The program should be robust, and not crash even if invalid data is entered – the program must have input validation.
2. The program should only allow authorised users to play – these users will be set within a list, which the programme will check against.
3. The game must allow 2 players to play.
4. The game must save winning scores, and sort this score into a scoreboard after the game has ended. This scoreboard should be able to be accessed from a main menu.
5. Each player gets 5 turns, each consisting of rolling 2 dice with 6 digits on each.
6. The programme should alert the user to the rules at the beginning of the game. Could be an optional section in the main menu.
7. After the game has finished, the game returns to the main menu, where the user can decide whether they want to “LEAVE GAME” or “PLAY AGAIN”.

Decomposition

* *Break the task down into small manageable chunks which can be coded separately:*

Design





Development

* *Copy and paste your code into this section*
* *Remember to try and add comments to your code to make it more readable!*
* *Show where you have tested your code and any changes you have made.*

**My program code:**

# more functions / procedures (scoreboards)

global gameOn

gameOn = True

#################

# Authenticator #

#################

# Defines a function called authenticateUser()

# This function is used to check the users credentials by cross-referencing their name against a list in a word file.

# Later on in the program, the function is called to variables, player1 and player2.

def authenticateUser(i):

# Authenticator

# Opens a file containing username list

# Splits file into a list

# Closes file

f=open("usernames.txt", 'r')

userList = f.readline()

userList = userList.split()

f.close()

# Var assignments

attempts = 0

locked = False

login = False

# While loop allows me to break the loop easily

while locked == False:

# Asks user for input

userName = input("Player %s: Please enter your username: " % (i))

# Checks user input against list

if userName in userList:

print("Logged In\n")

login = True

locked = True

return userName

# If the name isn't on the list, checks how many attempts have been taken

# Too many attempts "locks" the game

else:

print("Unsuccessful login.")

attempts += 1

if attempts >= 5:

locked = True

# if you have too many attempts, it will lock the

# program for 5 minutes

# this will help to prevent brute force attacks, increasing

# how robust the program is

import time

print("Locked out - please wait - 5 Minutes")

time.sleep(300)

attempts -= 1

locked = False

else:

pass

#########

# Rules #

#########

global player1Score

global player2Score

player1Score = 0

player2Score = 0

def rules():

#shows the rules of the game

print('''Two players roll two 6-sided dice.

Points are added based on the result of the dice.

The points rolled are added to the score. e.g. 3 and 4 --> 7

If the total is an even number, the user gets an extra 10 points.

If the total is odd, the user loses 5 points.

If the two dices roll the same, the user gets an additional dice roll.

The score cannot fall below 0. If this happens, your score will be reset to 0.

The person with the highest score after 10 rounds will win.

If the users have the same end result, the game will enter sudden death.

Only authorised users can play the game.

''')

# generates random number

def rollDice():

import random

dice = random.randint(1,6)

return dice

# if the two dices are doubles, this function will roll an extra dice

def doubles():

import random

dice = random.randint(1,6)

return dice

# adds the value of dice1 and dice2

def diceAdd(dice1, dice2,dice3):

diceTotal = dice1 + dice2 + dice3

return diceTotal

# Player 1 Turn

def player1Turn(player1Score):

import time

time.sleep(0.5)

# formatting

print("It is now %s's turn." % (player1))

print("Press ENTER / RETURN to roll the dice!")

input()

# runs rollDice function, x2

dice1 = rollDice()

dice2 = rollDice()

# compares for doubles

if dice1 == dice2:

print("Doubles! Extra roll!")

dice3 = rollDice()

else:

dice3 = 0

# uses diceTotal function to calculate score

diceTotal = diceAdd(dice1,dice2,dice3)

# prints diceTotal

print("You got a score of %s!\n" % (diceTotal))

# checks if odd / even

if diceTotal % 2 == 1:

# if odd, -5 points

print("However, this is an odd number. You lose 5 points.")

diceTotal -= 5

else:

# if even +10 points

print("You gain 10 points for an even number!")

diceTotal+= 10

print("Your overall score for this round is %s!\n" % (diceTotal))

player1Score = player1Score + diceTotal

return player1Score

# player 2 turn

def player2Turn(player2Score):

import time

time.sleep(0.5)

print("It is now %s's turn." % (player2))

print("Press ENTER / RETURN to roll the dice!")

input()

# runs rollDice function x2

dice1 = rollDice()

dice2 = rollDice()

#compares and checks for doubles

if dice1 == dice2:

print("Doubles! Extra roll!")

dice3 = rollDice()

else:

dice3 = 0

# finds dice total

diceTotal = diceAdd(dice1,dice2,dice3)

print("You got a score of %s!\n" % (diceTotal))

if diceTotal % 2 == 1:

print("However, this is an odd number. You lose 5 points.")

diceTotal -= 5

else:

print("You gain 10 points for an even number!")

diceTotal+= 10

print("Your overall score for this round is %s!\n" % (diceTotal))

player2Score = player2Score + diceTotal

return player2Score

def winnerCheck(player1Score,player2Score):

# checks who has the higher overall score

if player1Score > player2Score:

return True

elif player1Score == player2Score:

draw = True

# Runs sudden death function

suddenDeath()

else:

return False

# procedure

def suddenDeath():

import time

# wait for added effect :)

time.sleep(1)

print("SUDDEN DEATH")

time.sleep(1)

print("%s, roll dice!" % (player1))

input()

# rolls dice

suddenScore1 = rollDice()

print("\n%s, roll dice!\n" % (player2))

input()

# rolls dice

suddenScore2 = rollDice()

# tension

print("One person got a score of %s. The other, a score of %s\n" % (suddenScore1, suddenScore2))

time.sleep(1)

# drama

print("%s had a score of %s!" % (player1,suddenScore1))

print("%s had a score of %s!" % (player2,suddenScore2))

winnerCheck(suddenScore1,suddenScore2)

# Writes the top users score to a .txt file

# Add sorting and removing # but doesnt work

def writeScoreboard(playerName,playerScore):

# write into a file

playerScore = str(playerScore)

if len(playerScore) == 2:

playerScore = "0" + playerScore

elif len(playerScore) == 1:

playerScore = "00" + playerScore

f = open("scoreboard.txt","a")

f.write(str(playerScore)+" "+str(playerName)+"\n")

f.close()

f = open("scoreboard.txt","r")

lines=f.readlines()

f.close()

lines.sort(reverse = True)

f = open("scoreboard.txt","w")

f.writelines(lines)

f.close()

# Shows the scoreboard .txt file

def showScoreboard():

print("\nScoreboard:")

f = open("scoreboard.txt","r")

for lines in f:

print(lines)

f.close()

print("\n\n")

# open file

# read

# sort

# close

# reopen file

# write

# close

# overall main loop / control loop

def mainLoop(player1Score,player2Score):

import time

# for loop to have 10 rounds

for i in range(1,11):

time.sleep(0.2)

print("\nRound %s:\n" % (i))

# runs player turn functions

player1Score = player1Turn(player1Score)

player2Score = player2Turn(player2Score)

# checks score maintains over 0

if player1Score < 0:

player1Score = 0

elif player2Score < 0:

player2Score = 0

else:

pass

print("%s has a current score of %s." % (player1,player1Score))

print("%s has a current score of %s.\n" % (player2,player2Score))

winner = False

# checks who won the game

winner = winnerCheck(player1Score,player2Score)

if winner == True:

print("\n%s wins!\n" % (player1))

writeScoreboard(player1,player1Score)

else:

print("\n%s wins!\n\n" % (player2))

writeScoreboard(player2,player2Score)

#############

# Main Menu #

#############

def mainMenu():

player1Score = 0

player2Score = 0

menu = '''Play [G]ame

Show [R]ules

Show [S]coreboard

[E]xit

'''

print(menu)

gameSelect = input()

gameSelect = gameSelect.lower()

if gameSelect[0] == "g":

# Runs the authenticateUser() function and assigns the returned value to variables

# Also checks the names are different.

dif = False

while dif == False:

global player1

player1 = authenticateUser(1)

global player2

player2 = authenticateUser(2)

if player1 == player2:

print("You have used the same login both times.")

print("Please re-login using different users.\n")

else:

dif = True

print("Welcome to the Dice Game!")

mainLoop(player1Score,player2Score)

elif gameSelect[0] == "r":

rules()

elif gameSelect[0] == "s":

showScoreboard()

elif gameSelect[0] == "e":

gameOn = False

exit()

while gameOn == True:

mainMenu()

# End break

input()

Testing

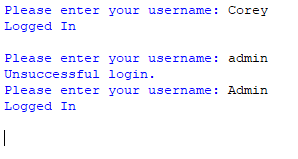
* *Show you have completed the tests you thought of*
* *Identify if you needed to make changes to your program*
* *Include the screenshots of the tests*

**My tests:**

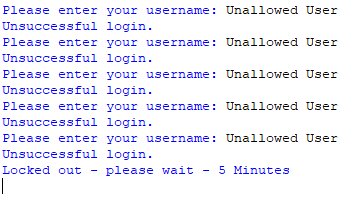
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | What am I testing? | Expected result | Pass/Fail | Do I need to change my program? If so, how? |
| 1 | Authentication block / sub-program | Lets user input username, checks against list and returns username. Repeats twice. If name is wrong, it will not allow the name to pass. | Pass | N/A |
| 2 | rules | Outputs the rules | Pass |  |
| 3 | rollDice | Outputs random number from 1 to 6 | Pass |  |
| 4 | Player1Turn | Allows user 1 to rollDice and prints result | Pass |  |
| 5 | Player2Turn | Same as above “ | Pass |  |
| 6 | doubles | If dice1 = dice2, roll dice3 | Pass |  |
| 7 | diceAdd | Finds total of dice1,dice2 and if rolled, dice3. | Pass |  |
| 8 | winnerCheck | Checks which player has a higher score | Pass |  |
| 9 | writeScoreboard | Writes to “scoreboard.txt” in append mode and sort | Fail – not sorting | Added place holder 0 before scores at 2 digits allowing the scoreboard to sort properly |
| 10 | showScoreboard | Opens file scoreboard.txt and reads lines in console | Pass |  |
| 11 | mainLoop | Control loop of the program | Pass |  |

**My test screenshots:**

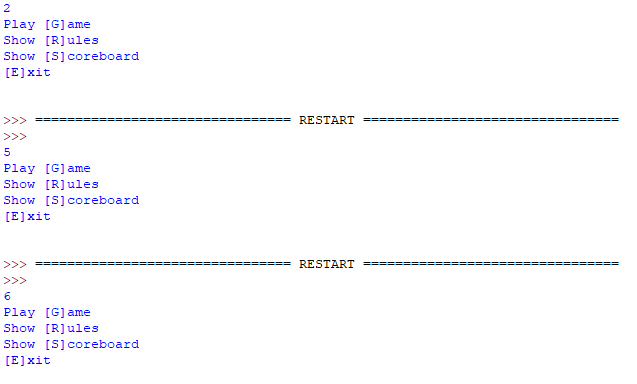
**Authentication: PASS –** Allows names on the list, doesn’t allow unauthorised users / mistyped names.



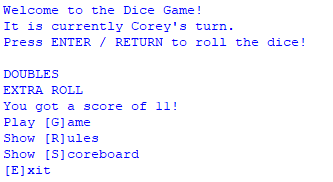
When an unallowed user tries to login 5 times, it will lock the program. This prevents brute force attacks.



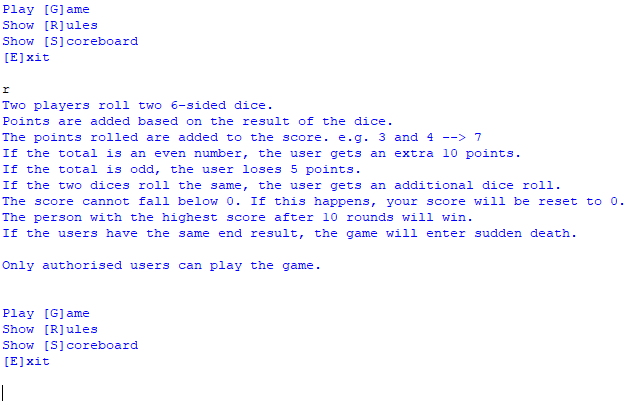
rollDice() – Random number generator



Doubles – If dice1 == dice2, roll again using doubles()



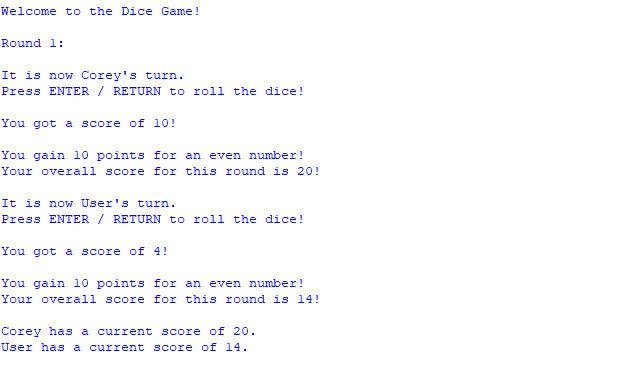
Rules:

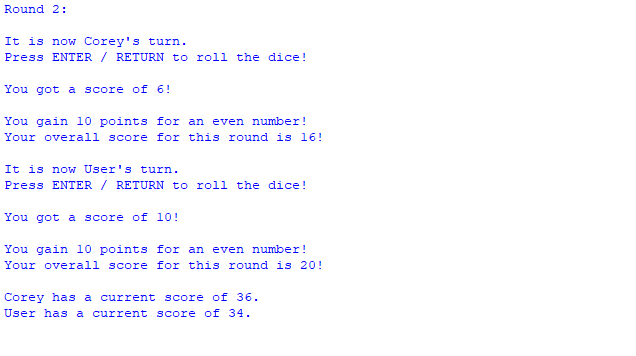


Show Scoreboard:

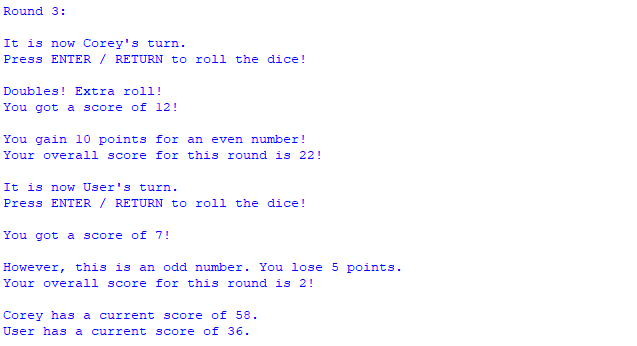


Player1Turn and Player2Turn, also main loop:

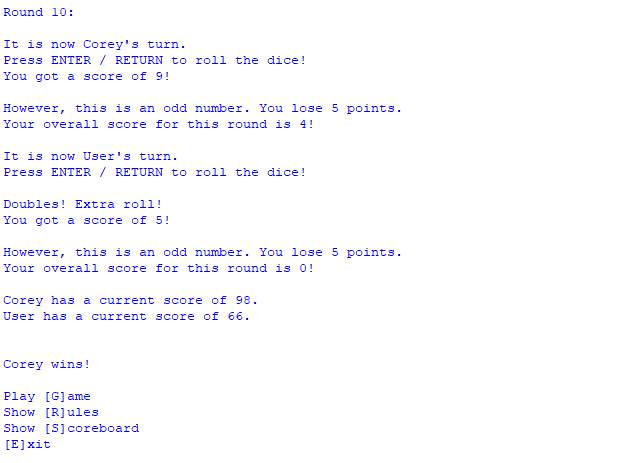




Doubles:



Winner Check:



Evaluation

* Evaluate how successful your program was. You may find it useful to refer back to your tests
* You should reflect on any new skills you have developed

This section should be approximately 200-500 words.

**How successful was my program?**

I believe my program is very successful at the goal of this coursework: to produce a dice game.

It allows the user to log in, only allowing pre-authorised users to play, before allowing them to roll the dice and play the game. It also gives them options in a main menu to view the scoreboard, view the rules or exit the game. It successfully recognises when the user rolls a double and allows them an extra roll. The program can add the scores together before subsequently adding the dice score to the users overall score. It then, after 10 rounds, checks which player has the highest score and informs the players.

In addition, I managed to add a feature, which sorts the scoreboard based on the user’s scores. This originally was a struggle and did not work, so I had to add a placeholder “0” or “00” when the user score has a length of only 1 or 2 digits.

**What new skills have I developed?**

How to return a value from a function and how to pass data in

Sorting data

Reinforcing previous knowledge

MOD % data operator

Global Variables

**Tests [IGNORE]:**

**Password Authentication:**

listOfusers = ["john","harry","steve"]

name1 = "frank"

name2 = "john"

name = input()

name = name.lower()

if name **in** listOfusers:

print("works2")

else:

pass

**IDEA: use “wait = input()” and “print(“Press enter to role your dice!”)”, then “import time” & “time.sleep()” to cause a delay. Random numbers could appear on screen to visualise the dice being rolled.**

**For I in range (0,5):**

**Print random integer (2,12)**