NEA

Programming Project Report

SAI RISHYANTH VISINIGIRI

YEAR 11, GCSE

ST. JOSEPHS COLLEGE, LONDON

Table of Contents

`1

Analysis

Requirements

The task requires to create a two-player dice game. The players need to be registered to play the game to ensure that they are authorised players. The game involves 5 rounds and each player rolls two 6-sided dice in each round. The points rolled on each player's dice is added to their score. If the total points is an even number then an additional 10 points are added to their score or if the total points are an odd number 5 points are subtracted from their score. If the player rolls a double, they get to roll one extra die and get the points rolled added to their score. Each player gets to roll 1 die if both players have the same score at the end, whoever gets the highest score wins. This repeats until any one of the player wins. The program then outputs the player's name who has won and store their score and their name in an external file.

Success criteria

- ✓ Each player should roll two 6-sided dice
- ✓ The points rolled on each player's dice are added to their score
- ✓ The score should be incremented by 10 if the total is an even number
- ✓ The score should be subtracted by five if the total is an odd number
- ✓ The player should get an extra die if they roll a double, the points rolled adds to their score
- ✓ The score should never be 0 at any point
- ✓ Both the players should get one extra die if their scores are equal. The program should repeat until one of the player gets higher points rolled than the other player.
- ✓ The name and score of the player who won should be outputted
- ✓ The winning players name and score should be stored in an external file and the score and player name of the top 5 winning scores should be outputted from the external file.

Problem Decomposition

- ✓ Allow the user to input their details that are authenticated to ensure they are authorised players by using a function
- ✓ The user details are stored in an external file
- ✓ Allow each player to roll two 6-sided dice by generating two random number between 1 and 6
- ✓ Calculate the points for each round for each player's total score.
- ✓ Calculate if the player's total score is an odd or even number and add or subtract based on the program rules

- ✓ Use a function to allow the user to play 5 rounds.
- ✓ The program then should output the winning player's score and name. The player's name and score is then stored in an external file.

Data Structure/Variable & Validation

Variable name	Data type	Validation	Use	
Name	string	Ensure that the name only consists of letters (and symbols)	Used further in the program to output the winning name	
Age	Integer	Ensure that the name doesn't include any letters or symbols	Personal user details	
Gender	String	Ensure that the name only consists of letters	Personal user details	
Username	String	Ensures that the username is already registered	Allows the user to login to access the game	
Password	String	Ensures that the password is already registered	Allows the user to login to access the game	
Score	Integer	Score at the start of the game equals to 0	Each player's score after each round is stored. Previous rounds score is added.	
Dice number	Integer	Ensures that the number is between 1 and 6	Generates two random numbers between 1 and 6	
Total_score	Integer		Adds the values rolled	
Num	Integer	Ensures that the remainder is either 0 or 1	Stores the remainder of Total_score / 2	

Test Plan & Test Data

Test number	Test data	Expected outcome	Actual outcome	Action required
1	If user enters string for age	Error message displayed when he enters the string.		
2	If user enters either wrong username or password	Outputs ‘ incorrect credentials’ and asks the username and password again.		
3	If the player rolls a double	The players gets an extra die to roll		
4	If the player’s total in a round equals to an even number	The rolled points are added to their score as wells as an additional 10 points added to their score		
5	If the player’s total in a round equals to an odd number	The rolled points are added to their score as well as a deduction of 5 points from their score.		
6	If the player’s score that is below 5 is deducted by 5	The score of the player would be negative		
7	If both the player’s score after 5 rounds are equal.	Both the players get an extra die to roll until any one of them gets higher than the other.		

Date Started

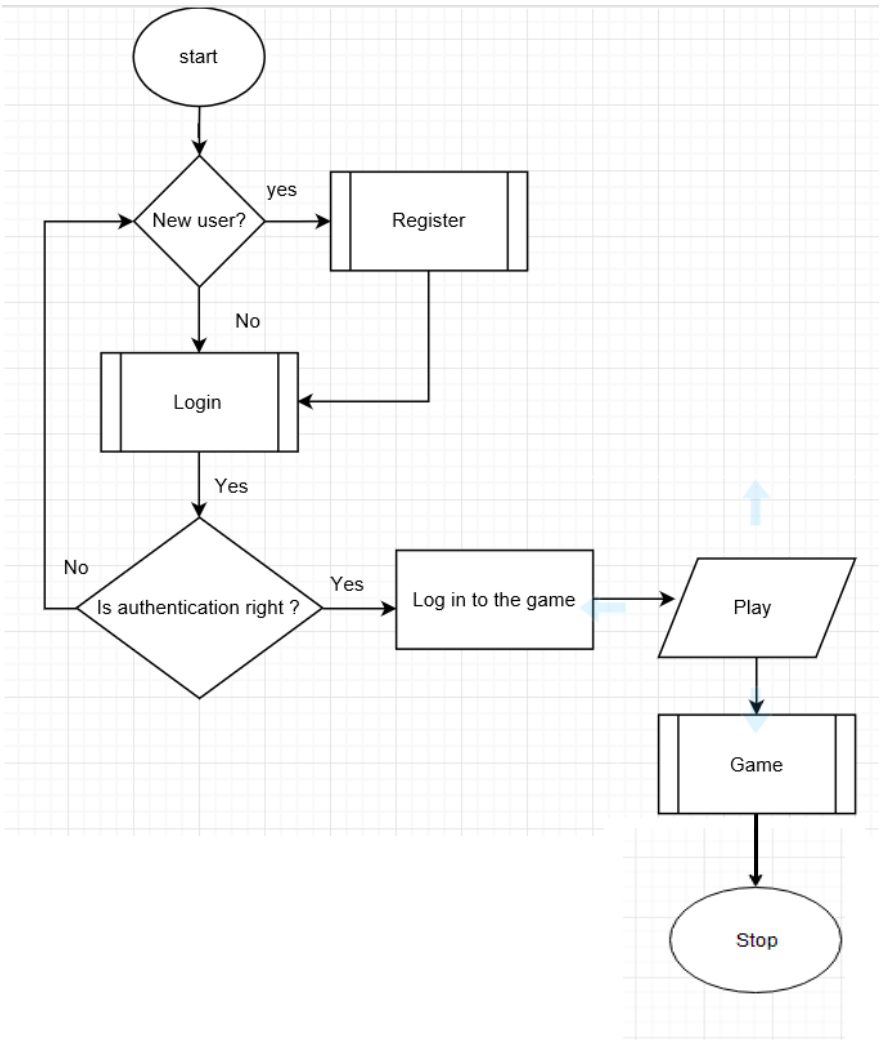
Date Completed

Design

Flow chart

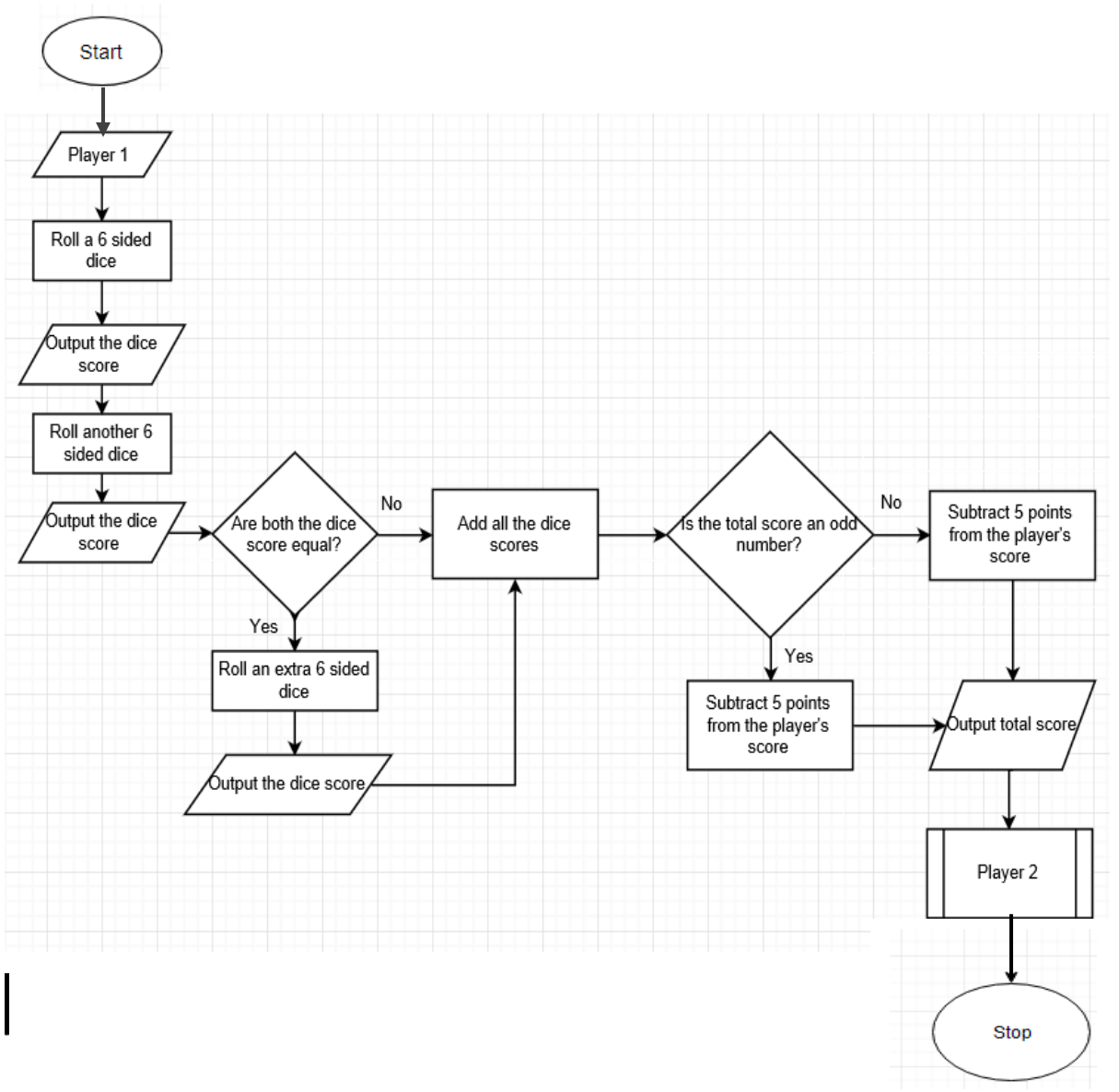
Pseudocode

User details and authentication



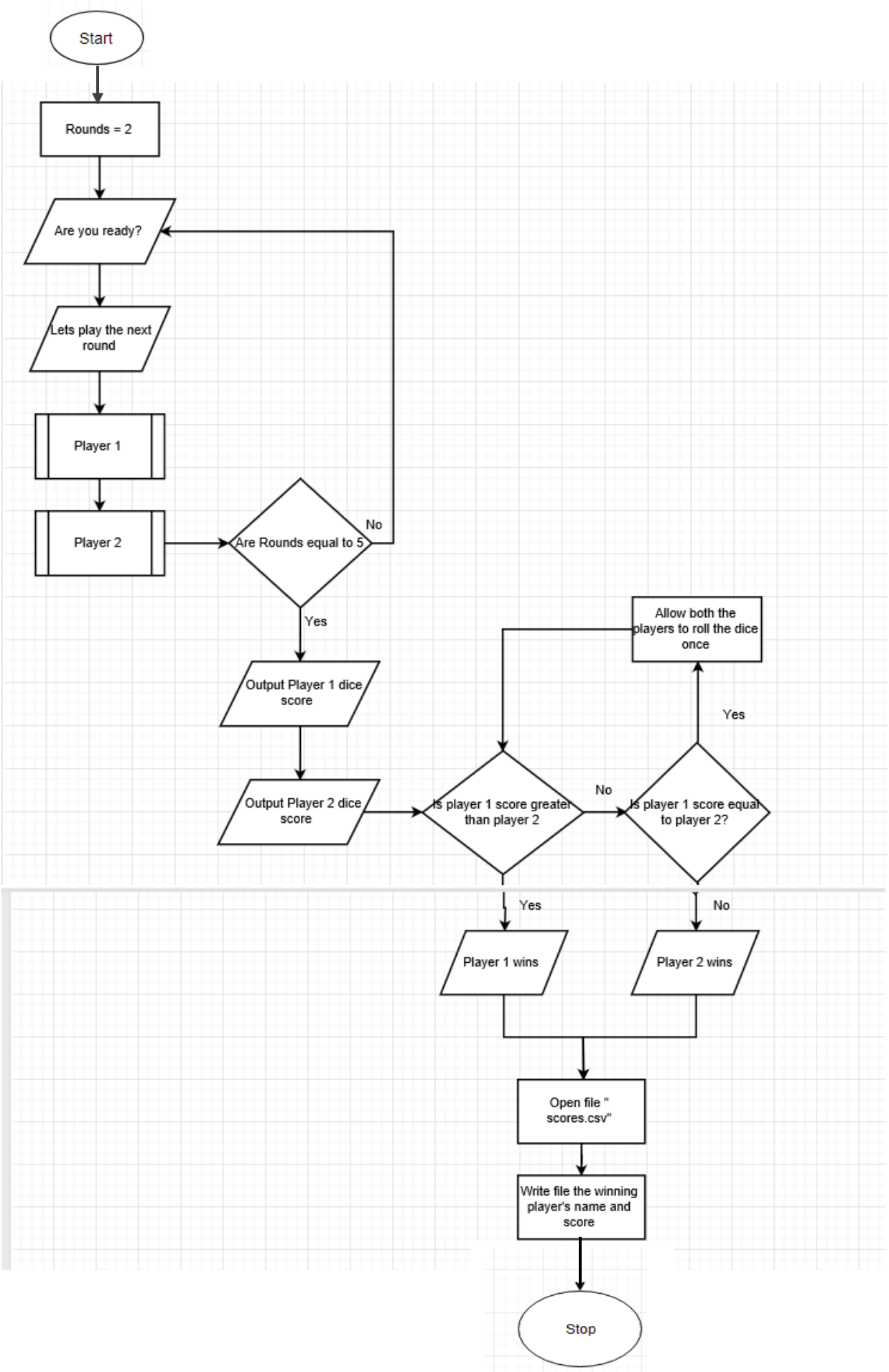

```
FUNCTION start ()
  WHILE TRUE
      INPUT user inputs whether he is a new user
      IF the user is a new user THEN
        Register()
      ELSE
        Login()
      ENDIF
   ENDWHILE
END FUNCTION

FUNCTION Register ()
    INPUT user's full name
    INPUT user's age
    INPUT user's gender
   while TRUE
      INPUT username
      OPEN file "user.csv"
      READFILE "user.csv"
      IF username is already taken in the "user.csv" file THEN
          OUTPUT " username  is already taken "
      ELSE
          BREAK
      ENDIF
      INPUT password
      WRITEFILE "user.csv" username
      WRITEFILE "user.csv" password
      CLOSEFILE "user.csv"
END FUNCTION

FUNCTION Login ()
   WHILE TRUE
      INPUT username
      INPUT password
      OPEN file "playerDetails.txt"
      Split the line in the text file by (' ')
      READFILE "playerDetails.txt"
      IF the username and password equals to any first & second words in the csv files THEN
          OUTPUT "Login Successful"
          return TRUE
      ELSE
          OUTPUT "Incorrect credentials"
          BREAK
Start ()
```

Calculation of the score



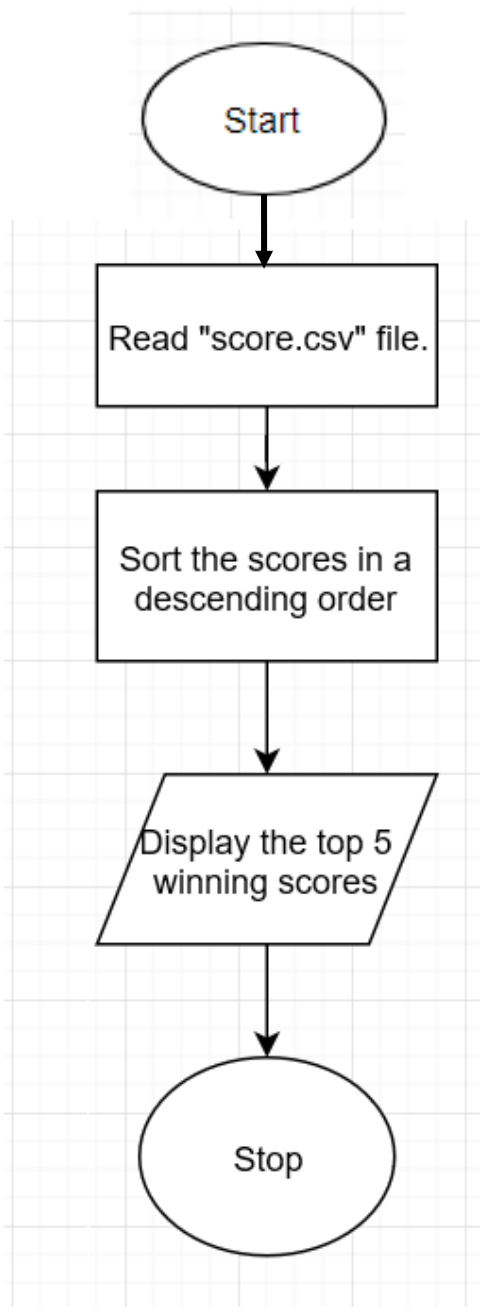

```
OUTPUT " Lets roll the dice!"
FUNCTION Player 1()
    VAR tries = 0
    Global VAR Player1_socre = 0
    Import random number
    VAR Dice1 is between 1 and 6
    Output Dice1 score
    Import random number
    VAR Dice2 is between 1 and 6
    OUTPUT Dice2 score
    IF Dice 1 equals to Dice 2 THEN
        OUTPUT "It's a double – roll and extra die"
        Import random number
        VAR Dice3 is between 1 and 6
        OUTPUT Dice3 score
        VAR Dice = Dice1 + Dice2 + Dice3
    ELSE
        VAR Dice = Dice1 + Dice2
    ENDIF
    Global VAR Player1_score is equal to Player 1_score + Dice
    VAR number equals Player1_score / 2
    IF number equals to 1 THEN
        OUTPUT " Its and odd number – 5 points subtracted"
        Player1_score = Player1_score -5
    ELSE
        OUTPUT " Its an even number – 10 additional points.
        Player1_score = Player_score + 10
    ENDIF
END FUNCTION
Player1()
Player2()
```

Iteration and displaying the winner



```
Rounds = 2
While Rounds is not equal to 2 do
    OUTPUT “Round” Rounds
    Player1()
    Player2()
    Rounds equals Rounds + 1
    IF Rounds equal to 5 THEN
        Winner()
        break
    ENDIF
FUNCTION Winner ()
  While True
    OUTPUT Player1_score
    OUTPUT Player2_score
    IF Player1_score is greater than Player_2 score THEN
        OUTPUT “Player 1 wins”
        OUTPUT  Player 1 name and their score
        Top5Players()
        break
    ELIF Player1_score equals to Player2_score THEN
         OUTPUT “Scores are equal; Each player rolls an extra die”
         OUTPUT “Player 1”
         Import random
         VAR extrDie1 is between 1 and 6
         OUTPUT extraDie1
         OUTPUT “Player 2”
         Import random
         VAR extrDie2 is between 1 and 6
         OUTPUT extraDie2
         IF extraDie1 is greater than extraDie2 then
             OUTPUT “Player 1 wins”
             OUTPUT Player 1 name and their score
             Top5Players ()
             Break
         ELIF extraDie1 is less than extraDie2 then
             OUTPUT “Player 2 wins”
             OUTPUT Player 2 name and their score
             Top5Players ()
             Break
         ENDIF
   ELSE
      OUTPUT “Player 2 wins”
             OUTPUT Player 2 name and their score
             Top5Players ()
             Break
    ENDIF
END FUNCTIONS
  Winner ()
```

Displaying the top 5 winning scores




```
FUNCTION Top5Players ()
        Open “score.csv” file
        Read “score.csv” file
        VAR lines equals to sorted lines (descending order)
        For line in line find (“ , “) do
            VAR score equals split the line by (“ , “)
            VAR score equals sorted score
         ENDFOR
        OUTPUT row [0] to row [4]
END FUNCTION
Top5Players()
```

Development

Main Menu

```
# Dice Game
# User Authentication

print("""
-----------------------------------------------
-- ----   ------------------------------  ----- --
          ====   ======  ======  =======
          =  =     ==    ==      =
          =   =    ==    =       ======
          =   =    ==    =       ======
          =   =    ==    ==      =
          ====   ======  ======  =======
-- -----  ------------------------------  -----  --
-- -----  ------------------------------  -----  -- """)
print("""
                                        =============
                                      =  ==          ==
                                     =   ==     .    ==
                                     =   ==          ==
                                     = . ==          ==
                                     =   =============
                                     =  =            =
                                     = =            =
                                      =============
                                     """)

def Start():                                    # Start function allows the user to login or register based on their choice
    print("-----------------Welcome  to DROP DEAD ---------------------")
    print("\n")
    print("Are you a new user ? Y or N ")
    choice=str(input("Please choose an option: ")) # Variable choice asks the user to login or register

    if choice == "Y":
        Register()                              # Allows the user to register if their choice is Yes
    elif choice == "N":
      Login()                                   # Allows the user to login if their choice is No
    else:
      print("\n")
      print("Please enter a valid option ( Y or N )")
      print("\n")
      Start()                                  # Takes the user back to the start menu if their choice is invalid.
```

Summary

- **The Start function takes the user to the main menu where he has the choice to login if he is an existing member or register if he is a new member.**
- **The program asks the user if he is a new user**
- **The user has the choice either to enter Y (Yes) or N (No).**
- **If the user enters Y it would allow the user to register and would run the Register function.**
- **If the user enters N it would allow the user login and would run the login function**

Output of def Start () function:

```
-----------------------------------------------
-- ----   ------------------------------  ----- --
          ====   ======  ======  =======
          =  =     ==    ==      =
          =   =    ==    =       ======
          =   =    ==    =       ======
          =   =    ==    ==      =
          ====   ======  ======  =======
-- -----  ------------------------------  -----  --
-- -----  ------------------------------  -----  --


                                        =============
                                      =  ==          ==
                                     =   ==     .    ==
                                     =   ==          ==
                                     = . ==          ==
                                     =   =============
                                     =  =            =
                                     = =            =
                                      =============


 Player 1


---------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option:
```

The programs starts by allowing the user to create a new account or login to the existing one.

```
----------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option: gpifj


Please enter a valid option ( Y or N )


----------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option:
```

The program doesn't accept any other character apart from Y or N

Outputs to the reader to enter a valid option and allows the user to re-enter their option

User Authentication

Register

```
def Register ():

    # Allows the user to input personal details

    global name
    name=str(input("Please enter your first name: "))
    name=name.capitalize()
    age=int(input("Please enter your age (AA) : "))
    age=str(age)
    import random
    number=random.randint(1,10)
    number=str(number)
    global username
    username = name[:3] + age[:2] + number   # Randomly Generates a username based on their name and age
    print("\n")
    print(" Your username is",username)
    print("\n")

    password=str(input ( " Create a password: ")) # Allows the user to create a password


  # Amends the name, username and password of the player to the text file ( playerDetails.txt )

    file=open("playerDetails.txt","a")

    file.write (name)
    file.write (",")
    file.write (username)
    file.write (",")
    file.write (password)
    file.write (" \n ")

    file.close()    # Saves the text file.
```

Asks the user their name and age

Writes the name, username and password of the player to playerDetails.txt file.

```
username=str()       # Variable username assigned outside the function to delcare a global varia

print("\n")
print (" Player 1 ")
print("\n")
Start()

player1username=username  # Assigns the username of the first player as Player 1 username
player1Name=name          # Assings the name of the first player as Player 1 Name


print("\n")
print ( " Player 2 ")
print("\n")
Start()
print("\n")

player2Name=name          # Assigns the username of the first player as Player 2 username
player2username=username  # Assings the name of the first player as Player 2 Name
```

Username and Name after the start function is run is assigned to player 1

Username and Name after the start function is run is assigned to player 2

Output of def Register():

```
----------Welcome  to DROP DEAD ----------


Are you a new user ? Y or N
Please choose an option: Y
Please enter your first name: Rishyanth
Please enter your age (AA) : 15


 Your username is Ris154


 Create a password: Rishy


You registered successfully
Waiting for player 2 to login/register
Processing..........................................................


 Player 2

----------Welcome  to DROP DEAD ----------


Are you a new user ? Y or N
Please choose an option:
```

The user enters their details here and the program generates a username and allows to create a password.

Program shifts to the main menu for player 2 to login/register if player has registered successfully.

playerDetails - Notepad

File Edit Format View Help

Rishyanth,Ris151,Rishy15

Amends the player name, username and password to “playerDetails” text file.

Summary

If the player is a new user then the player is supposed to register in order to be an authorised player to play the dice game. The program asks the user his first name and age and generates a username for the player based on his name and age. The player is then expected to create a password. The details are stored in an external file “ playerDetails.txt” . Then player 2 is asked to either login or register.

Login

```
def Login():

    print('-------------Welcome to the login page----------------')


    username=input('username: ')     # Asks the user to input username

    password=input('Password: ')     # Asks the user to input password

    for line in open ('playerDetails.txt','r').readlines():
        logininformation=line.split(",")
        if username!=logininformation[1] and password!=logininformation[2]:
            print ("Incorrect Credentials")
            Start()

        else:
            print("Correct credentials")
```

The program takes the player to login function if is an existing player.

Reads the text file “playerDetails”.

Splits the text file into lines by separating them through commas.

The programs takes the player to the main menu if his username and password are incorrect

The program starts the game or either moves onto the login page of player 2 if the username and password are correct.

Output: The username and password are correct

```
 Player 1


----------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option: N
-------------Welcome to the login page----------------
username: Ris151
Password: Rishy15
Correct credentials
----------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option:
```

Takes the player to login page if the player enters N

The program reads the text file playerDetails to check whether the username and password are valid

Output: The username or password (or both) are incorrect

```
----------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option: N
-------------Welcome to the login page----------------
username: Max123
Password: Kbalan
Incorrect Credentials
----------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option:
```

The username and password entered by the player are not found in the text file then program outputs the details entered as incorrect

The program heads to the main menu if the player entered incorrect credentials to either login again or

playerDetails - Notepad

File Edit Format View Help

```
Rishyanth,Ris151,Rishy15
Maxwell,Max156,Kablan
David,Dav249,Stephen
Rishyanth,Ris154,Rishy
Rishyanth,Ris143,Rishy
Max,Max1510,Kablan
```

Reads the text file to authenticate for valid username and password

Summary

If the player is an existing user the player inputs “N”. The program proceeds to the login page where the player inputs the username and password. The program reads the text file “playerDetails” to authenticate if it is a valid username and password i.e. the player is an existing player or not. If the username and password the player entered matches to those in the text files, “ correct credentials” message is outputted and either the game stars if the player is player 2 or either proceeds to the main menu to allow player 2 to login if the person is player 1. If the username or the password is incorrect (or both) the program displays that “ incorrect credentials “ are entered and moves to the main menu that allows the player either to login again or to register to the game

Game Rules

```
print("\n")
print("""

 ==      ======   =======  ======    ======  ======  =======   ======   ===  =   ==
 ==      ==         ==     ==        =    =  ==      =           =      = =  =   ==
 ==      ======     ==       ====    =====   ======  =    ==     =      =  = =   ==
 ==      ==         ==         ====  =    =  ==      ===  ==     =      =   ==
 ======  ======     ==     =====     ======  ======   =====    ======   =    =   ..
                                                                                 ..

""")
print("\n")
print("""
      ======================================================= RULES ============================================================
   =                                                                                                                          =
   =  1. The games includes 5 rounds                                                                                          =
   =  2. Each player gets a chance to roll the dice twice in each round                                                       =
   =  3. The dice number is added to the player's score                                                                       =
   =  4. If the two dice numbers are equal the player gets to roll an extra dice                                              =
   =  5. If the player rolls an even number, an extra 10 points are added.                                                    =
   =  6. If the player rolls an odd number, 5 points are removed from their score                                             =
   =  7. The person with the highest score at the end of round 5 wins                                                         =
   =  8. if the score at the end of round 5 of both the plaayers are equal, they get to roll the dice until one of them wins  =
   =                                                                                                                          =
      ========================================================================================================================  """)
```

Outputs the rules of the game before the game starts

```
def diceNumber ():    # Imports a random number between 1 and 6

    import random
    global diceScore
    diceScore=random.randint(1,6)
    print("Your Dice Score is",diceScore)
    global Score              # Gloabl Varibale Socre
    Score=Score+diceScore     # Adds the dice score to the total score
    Dice=diceScore%2
    if Dice==0:
      print("Its an EVEN number!")  # if the dice score is an even number adds additional 10 points
      Score=Score+10

      print("Your current score in this round is",Score)  # Outputs the currect score in the round
    else:
      print("Its an ODD number!")  # If the dice scor eis an odd number it takes away 5 points from the total score
      Score=Score - 5
      if Score<0:
        Score=0    # makes sure that the player's score is not below 0
      print("Your score at the end of round",Round,"is",Score)
```

Generates a random number between 1 and 6 and append it to the variable dice score.

The program then decides whether the dice score is odd or even and appends and reduces the score based on the rules

Output:

```
==      ======   =======  ======    ======  ======  =======   ======   ===  =   ==
==      ==         ==     ==        =    =  ==      =           =      = =  =   ==
==      ======     ==       ====    =====   ======  =    ==     =      =  = =   ==
==      ==         ==         ====  =    =  ==      ===  ==     =      =   ==
======  ======     ==     =====     ======  ======   =====    ======   =    =   ..
                                                                                 ..


   ======================================================= RULES ============================================================
  =                                                                                                                          =
  =  1. The games includes 5 rounds                                                                                          =
  =  2. Each player gets a chance to roll the dice twice in each round                                                       =
  =  3. The dice number is added to the player's score                                                                       =
  =  4. If the two dice numbers are equal the player gets to roll an extra dice                                              =
  =  5. If the player rolls an even number, an extra 10 points are added.                                                    =
  =  6. If the player rolls an odd number, 5 points are removed from their score                                             =
  =  7. The person with the highest score at the end of round 5 wins                                                         =
  =  8. if the score at the end of round 5 of both the plaayers are equal, they get to roll the dice until one of them wins  =
  =                                                                                                                          =
     ========================================================================================================================
```

Summary

Once both the players have gone through user authentication the game rules are displayed for the players. The diceNumber function is declared to roll a 6 sided dice and append the dice score to the player's score. If the dice score is an even number an additional 10 points are added to the score of the player. If the dice score is an odd number the score is reduced by 5. The program ensures that the score never goes down than 0.

Game – Rolling the dice (Player 1 and Player 2)

```
name=str()
diceScore=int()
Round=int()
Round=1
Score=0
player1Score=int(0)
player2Score=int(0)
Winner=str()
WinnerScore=str()
diceWinner=str()
```

Global variables assigned in the program are initially declared outside the function

```
def Player1 ():          # Allows Player 1 to roll the dice twice in each round

    print("\n")
    print(" Welcome to round", Round)
    print("\n")
    print(" Player 1, you play first")
    print("\n")
    print("Press ENTER to Start")
    Enter=input()
    print("Are you ready!")
    print("Lets roll the dice!....3...2...1..begin!")
    print("\n")

    global player1Score
    diceNumber()          # Defines the DiceNumber function to add the dice score to player 1 score in eavery round
    diceNumber1 = diceScore
    print("Press ENTER to roll the dice")
    ENTER=input()
    diceNumber()       # Rolls the dice twice
    diceNumber2 = diceScore
    if diceNumber1 == diceNumber2:     # Allows the user to roll a double if the to dice numbers are equal.
      print("You got a DOUBLE!")
      print("Press ENTER to roll the dice")
      ENTER=input()
      diceNumber()


  player1Score=Score

ef Player2 ():          # Allows Player 2 to roll the dice twice in each round
  print("\n")
  print(" Player 2 its your turn")
  print("\n")
  print("Press ENTER to start")
  enter=input()
  print("\n")
  print("Are you ready!")
  print("Lets roll the dice!....3...2...1..begin!")
  print("\n")
  global player2Score
  diceNumber()       # Defines the DiceNumber function to add the dice score to player 2 score in eavery round
  diceNumber1 = diceScore
  print("Press ENTER to roll the dice")
  ENTER=input()
  diceNumber()      # Rolls the dice twice
  diceNumber2 = diceScore
  if diceNumber1 == diceNumber2:   # Allows the user to roll a double if the to dice numbers are equal.
    print("You got a DOUBLE!")
    diceNumber()
  player2Score=Score
```

Calls the function "diceNumber" to roll the dice for player 1.

Assigns the dice score to player 1's score after the player rolls the dice once.

Assigns the dice score to player 1's score after the player rolls the dice the second time.

Repeats the program for player 2 similar to player 1

```
def Winners ():     # Declares round as a global variable
  global Round

number=int()
def DiceRoll():  # Dice roll function generates a random number between 1 and 6
  import random
  global number
  number=random.randint(1,6)
  print("Your Dice Score is",number)

while Round<=5:  # Allows both the players to play until the end of round 5 through loop
  Player1()
  Player2()
  Round=Round+1
  choice=str(input("Press enter to move to the next round"))    # Moves on to the next round if round is less and not equal to 5
  Winners()
```

The function roll is declared to roll a 6-sided dice

Allows both the players to roll dice twice until the rounds are equal to 5. Increments 1 to variable Round until Round is equal to 5

```
if Round==6:        #  If rounds are equal to 6

    print("\n")
    print("Processing.................................")
    print("\n")
    print("Waiting to display the winner.............. ")
    print("Player 1, your score is",player1Score)
    print("\n")
    print("Player 2, your score is,",player2Score)

    # Outputs both the players score and proceeds to the verfication function where the winner is decided.
    Verification()
```

The total scores of player 1 and player 2 are outputted if the rounds equal to 6.

Output:

```
 Welcome to round 1

 Player 1, you play first

Press ENTER to Start

Are you ready!
Lets roll the dice!....3...2...1..begin!

Your Dice Score is 2
Its an EVEN number!
Your current score in this round is 12
Press ENTER to roll the dice

Your Dice Score is 1
Its an ODD number!
Your score at the end of round 1 is 8

 Player 2 its your turn

Press ENTER to start

Are you ready!
Lets roll the dice!....3...2...1..begin!

Your Dice Score is 2
Its an EVEN number!
Your current score in this round is 20
Press ENTER to roll the dice

Your Dice Score is 5
Its an ODD number!
Your score at the end of round 1 is 20
Press enter to move to the next round

 Welcome to round 2

 Player 1, you play first

Press ENTER to Start
```

Player 1 gets to roll the dice first

Each player gets to roll the dice twice in each round

Player's score at the end of each round is outputted

Player 2 gets the chance to roll the dice after player 1 has rolled the dice twice and the score is calculated in that round

After both the players have rolled the dice the game then moves onto the next round. The program is repeated until 5 rounds

Output (If the player rolls a double):

```
 Player 1, you play first

Press ENTER to Start

Are you ready!
Lets roll the dice!....3...2...1..begin!

Your Dice Score is 1
Its an ODD number!
Your score at the end of round 5 is 46
Press ENTER to roll the dice

Your Dice Score is 1
Its an ODD number!
Your score at the end of round 5 is 42
You got a DOUBLE!
Press ENTER to roll the dice

Your Dice Score is 4
Its an EVEN number!
Your current score in this round is 56
```

The player gets a chance to roll an additional dice if he rolls a double

Player rolls an additional dice

Output (When 5 Rounds are finished):

```
Processing.................................


Waiting to display the winner..............
Player 1, your score is 122


Player 2, your score is, 148
```

Summary

Player 1 and Player 2 roll the dice twice in each round. The scores are added to their total score. If a player rolls a double he gets to roll and additional dice. If both the players roll the dice twice then the game moves onto the next round. The program is repeated unitl the rounds are less or equal to 5. If the rounds are equal to 6 the scores of both player 1 and player 2 are outputted after the end of 5 rounds and then the program proceeds to the verification function where the winner is decided.

Winner – (Displaying the Winner)

```
def Verification():            #  Verifys the player's score to decide the winner of the game after 5 rounds

  if player1Score > player2Score:

      global player1Name
      print(player1Name,"(Player 1) Wins !")
      print("Well Done")
      print("Processing........")
      print("Don't miss out the leaderboard !")
      print("\n")
      print('   These are our Top 5 scorers ')
      print("""
  =     =====     =      =====  ====  =====  =====   ===      =      =====    =====
  =     =        ==      =    = =     =   =  =   =  =   =    =    ===     =   =    =
  =     ====    = = =    =    = ===   = =    ====   =   =   =====    = = =    =    =
  =     =      =     =   =    = =     =   =  =   =  =   =  =      =  =    =   =    =
  ====  ===== =       =  =====  ====  =    = === =   ===   =      =  =     =  =====
  """)
      global Winner
      Winner=player1Name

      global WinnerScore
      WinnerScore=str(player1Score)

      Leaderboard()       # The program then proceeds to the leaderboard function

  elif player2Score > player1Score:
      global player2Name
      print(player2Name,"(Player 2 )Wins !")
      print("Well Done")
      print("Processing.........................")
      print("Don't miss out the leaderboard !")
      print("Press ENTER to open the Leaderboard")
      print("\n")
      print('---------THESE ARE OUR TO 5 SCORERS---------- ')
      print("""
  =     =====     =      =====  ====  =====  =====   ===      =      =====    =====
  =     =        ==      =    = =     =   =  =   =  =   =    =    ===     =   =    =
  =     ====    = = =    =    = ===   = =    ====   =   =   =====    = = =    =    =
  =     =      =     =   =    = =     =   =  =   =  =   =  =      =  =    =   =    =
  ====  ===== =       =  =====  ====  =    = = = =   ===   =      =  =     =  =====
  """)

      Winner=player2Name
      WinnerScore=str(player1Score)

      Leaderboard()
```

If the score of player 1 is higher than player 2 at the end of 5 rounds then the program outputs Player 1 as the winner

The winner name and the score is declared as a global variable to append to the leader board

As the program proceeds to the leader board function, the winner is outputted the leader board.

If Player 2 has higher score than player 1 at the end of 5 rounds, then the player is displayed as the winner.

The winner name and score is declared as global variable and is appended to the leader board

```
  elif player1Score==player2Score:      # If scores are equal of both player

      print("Your scores are equal")
      print("\n")
      print("Both of you will roll the dice again")
      print("\n")
      print("Player 1, your turn")
      print("\n")
      print(" Are you ready!")
      print("Lets roll the dice!....3...2...1..begin!")
      print("\n")
      DiceRoll()        # Rolls the dice
      Player1=number
      print("Player 2, your turn")
      print("\n")
      print(" Are you ready!")
      print("Lets roll the dice!....3...2...1..begin!")
      print("\n")
      DiceRoll()       # Rolls the dice
      Player2=number
      if Player2==Player1:
        Winners()
         # The program repeats the code where the players roll the dice until the both the scores are different
```

Player 1 rolls the dice

Player 2 rolls the dice

Repeats the Verification function to decide the winner.

If one of the players as higher score than the other, then the program proceeds to the leader board function whereas it loops the program until one of the player has different dice score than the other.

Output (If the scores of both the players' scores are different) :

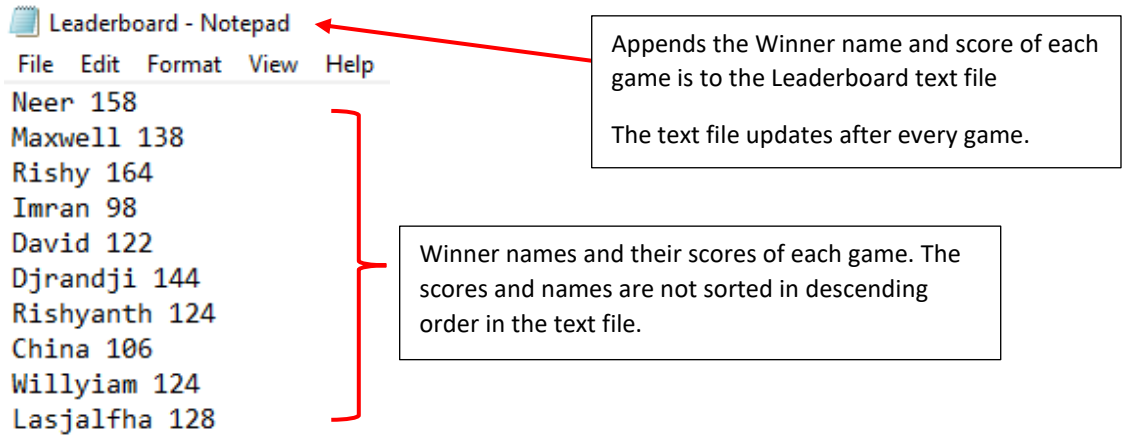
```
Processing.................................


Waiting to display the winner..............
Player 1, your score is 122


Player 2, your score is, 148
Max (Player 2 )Wins !
Well Done
Processing.............................
Don't miss out the leaderboard !
Press ENTER to open the Leaderboard
```

Summary

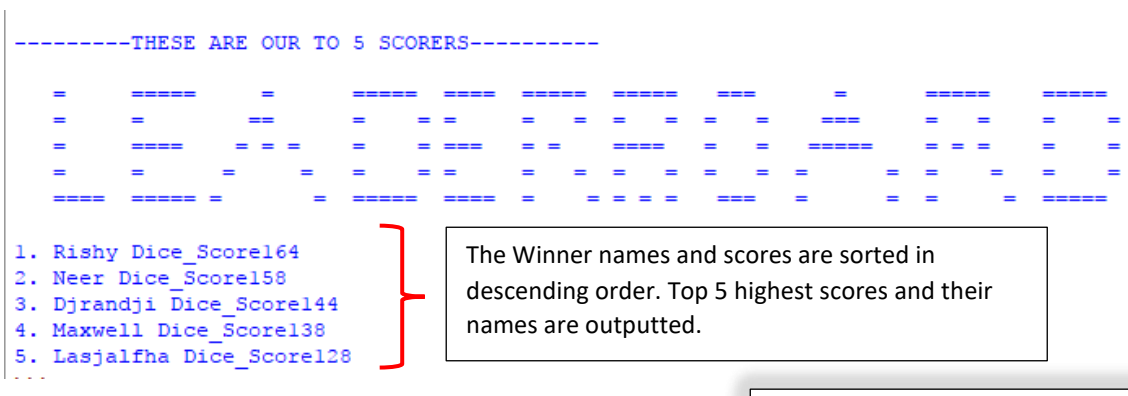
The verification function is declared to compare player 1 and player 2 scores. If player 1 score is higher than player 2 then Player 1 is displayed as winner. If player 2 score is higher than player 1 then player 2 is displayed as a winner. The Winner's name and score are declared as global variable. The program then proceeds to the leader board function where the winner's name and score is appended to the leader board text file.

Leader Board

```
def Leaderboard():                          # Leaderboard function stores the winner name and score onto the text file Leaderboard.txt

  file=open("Leaderboard.txt","a")
  file.write(Winner)                        # Appends the winner name to the text file
  file.write(" ")
  file.write(WinnerScore)                   # Appends the winner score to the text file
  file.write("\n")
  file.close()

  scores=[]
  for line in open('Leaderboard.txt','r').readlines():      # Reads the leaderboard tect file.
    info = line.split(' ')                                  # Splits the lines in the tect file.
    LeaderBoard={('Dice_Score'):int(info[1]),('Winner_Name'):info[0]}
    scores.append(LeaderBoard)          # Appends the dice scores to the list named scores

    # Dictonary stores the list of dice scores and winner names

    scores = sorted(scores, key=lambda s: s['Dice_Score'],reverse=True)    # Sortes the winner scores in the list "scores" into descending order

    lines=int(len(scores))          # Variable lines is assigned the length of thre
    if lines > 5:                   # If the players are more than 5 then select the top 5 players from the list.
        lines=5
    ranks=1
  for number in range(lines):

    rank=str(scores[number])        # Assigns the the top 5 player's name and score to the variable rank

    rank= rank.replace(' ', '')
    rank= rank.replace("'", '')
    rank= rank.replace(":","")
    rank= rank.replace(","," ")
    rank= rank.replace(',Winner_Name:', '\t')    # Removes the inverted commas and unecessary text from the list

    location=str(ranks)

    print((location+'.'),rank[12:-1])           # Outputs the player's name and score with top score
    ranks+=1                                    # Appends 1 to the the variable ranks to move to the next player with highest score.
```



Output:



Summary

The Leaderboard function is declared to append the Winner's name and score to the leader board text file. The winner's name are mapped to the winner's score in a dictionary names "LeaderBoard". The winner's name and score are then appended into a list names "scores". The Dice scores are then sorted into descending order. The players with top 5 high scores in the list are displayed. The text file updates after every game as it stores the winner name and score. The Leader board is then sorted and displays the top 5 highest scorers.

Evaluation

Test Plan

Test number	Test data	Expected outcome	Actual outcome	Action required
1	If user enters string for age	Error message displayed when he enters the string.	Syntax error was displayed as the program expected an integer.	Format of the way age to inputted is displayed to the user to prevent syntax error.
2	If user enters either wrong username or password	Outputs ‘ incorrect credentials’ and asks the username and password again.	Incorrect credentials outputted and displayed main menu to allow the user to either Login or register .	
3	If the player rolls a double	The players gets an extra die to roll	Program allows the reader to roll an extra dice.	
4	If the player’s total in a round equals to an even number	The rolled points are added to their score as wells as an additional 10 points added to their score	Outputs that the player has rolled an even number and outputs the total score after adding the 10 points.	
5	If the player’s total in a round equals to an odd number	The rolled points are added to their score as well as a deduction of 5 points from their score.	Outputs that the player has rolled an odd number and outputs the total score after reducing the score by 5 points.	
6	If the player’s score that is below 5 is deducted by 5	The score of the player would be negative	Negative number is displayed if the score is reduced below 0	If statement used to set the score to 0 if it goes below 0
7	If both the player’s score after 5 rounds are equal.	Both the players get an extra die to roll until any one of them gets higher than the other.	Both the players get an extra dice to roll. Loops the program until one of them wins.	
8	If unexpected string is entered in the main menu other than Y or N	Output the player to enter a valid option and allow him to re-enter	Syntax error displayed if a string is entered other than Y or N	If statement could be used to call the start function until the option is Y or N.
9	If the Rounds are equal to 5	Output the scores of the players and decide the winner	Moves onto the next round. Continues to loop the program	Assign variable rounds value 1. Validate variable round. If round equals 6 then output the players’ score
10	If the player’s in the leader board are less then 5	Display the sorted leader board with winner’s name and score of the players below 5 people.	Error message displayed.	If statement used to find the number of lines in the list “scores” that contains the sorted list of top 5 winners. If the lines are less than 5 then output the highest scorers that are below 5.

11.	If Round 5 is completed	Both the player's score is outputted and the winner is decided. A leader board with top 5 highest scorers' names and score are displayed.	The program displayer the scores of the both the players and displayed the winner and an updated leader board. However, an error was displayed as there were less then 5 players name in the external file.	`lines=int(len(scores))`
				`if lines > 5:`
				`lines=5`
				`ranks=1`
				`for number in range(lines):`
				`rank=str(scores[number])`
				The names and scores of the winners' in the list score are validated to ensure whether the players are less than 5. If the players are less then 5 then the player's name and score are outputted of those who are below 5 players.

12.	The player inputs wrong username and password	The program should display that incorrect credentials are entered and should return to the main menu	Incorrect Credentials displayed and proceeded to the main menu	

Meeting the Success Criteria

	Success Criteria	How the success criteria are meted	Evidence

1.	Each player should roll two 6-sided dice	A function called DiceNumber() is declared that allows the player to roll the dice. Random module is used to generate a random number between 1 and 6. The DiceNumber() function is called twice for each player in each round so that the player rolls the 6 sided dice twice.	`Welcome to round 1`
			`Player 1, you play first`
			`Press ENTER to Start`
			`Are you ready!`
			`Lets roll the dice!....3...2...1..begin!`
			`Your Dice Score is 2`
			`Its an EVEN number!`
			`Your current score in this round is 12`
			`Press ENTER to roll the dice`
			`Your Dice Score is 1`
			`Its an ODD number!`
			`Your score at the end of round 1 is 8`

2.	Points rolled on each dice are added to their score.	A Variable named Score is created to assign the points rolled on each dice to the variable Score	`diceNumber ():  # Imports a random number between 1 and 6`
			`import random`
			`global diceScore`
			`diceScore=random.randint(1,6)`
			`print("Your Dice Score is",diceScore)`
			`global Score  # Gloabl Varibale Socre`
			`Score=Score+diceScore  # Adds the dice score to the total score`
			`Dice=diceScore%2`

3.	The score should be incremented by 10 if the total is an even number	An arithmetic operator " % " which is a modulo is used to find the remainder when the score is divided by 2. The remainder value is assigned to a variable named Dice. The value is validated through if statement. The score is incremented by 10 if the value is a 0.	`Dice=diceScore%2`
			`if Dice==0:`
			`print("Its an EVEN number!")`
			`Score=Score+10`
			`Your Dice Score is 2`
			`Its an EVEN number!`
			`Your current score in this round is 12`
			`Press ENTER to roll the dice`

<table>
<tr>
<td>4.</td>
<td>The score should be subtracted by five if the total is an odd number</td>
<td>An arithmetic operator “ % “ which is a modulo is used to find the remainder when the score is divided by 2. The remainder value is assigned to a variable named Dice. The value is validated through if statement. The score is reduced by 5 if the value is a 1.</td>
<td>

```
  else:
    print("Its an ODD number!")
    Score=Score - 5
```

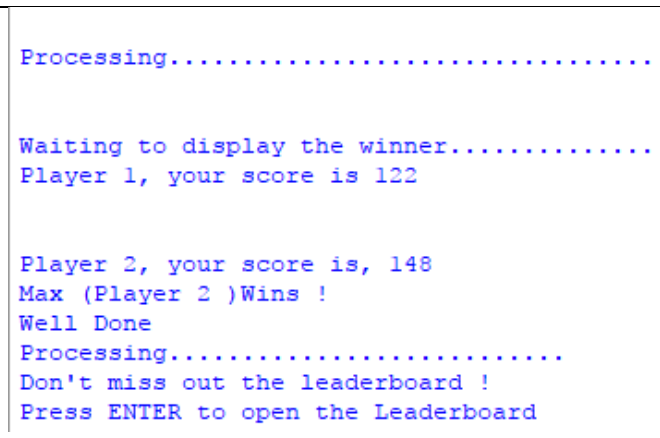
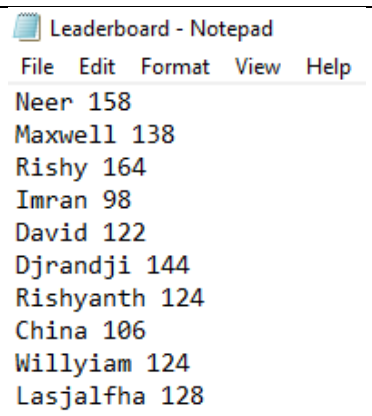
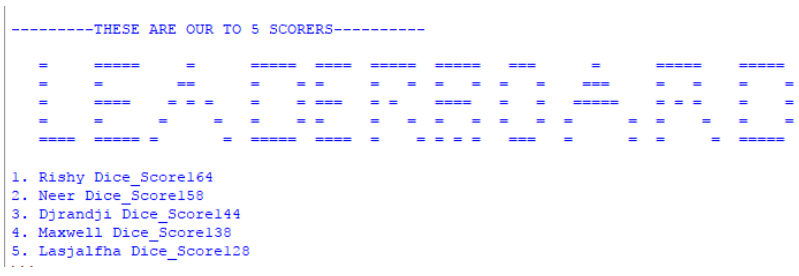
```
Your Dice Score is 1
Its an ODD number!
Your score at the end of round 1 is 8
```
</td>
</tr>
<tr>
<td>5.</td>
<td>The score should never be 0 at any point</td>
<td>If statement is used to validate if the Score is reduced to or less than 0. If the score is reduced to less then 0 then the variable Score is assigned the value 0</td>
<td>

```
if Score<0:
  Score=0    # makes sure that the player's score is not
print("Your score at the end of round",Round,"is",Score)
```
</td>
</tr>
<tr>
<td>6.</td>
<td>The player should get an extra die if they roll a double, the points rolled adds to their score</td>
<td>The DiceNumber() function is called to roll the dice. Variable DiceNumber1 is assigned the 1st dice score and Variable DiceNumber2 is assigned the 2nd dice score. If statement is used to validate if both the scores are same. If the scores are the same then the DiceNumber() function is called to roll and extra dice.</td>
<td>

```
global player1Score
diceNumber()        # Defines the DiceNu
diceNumber1 = diceScore
print("Press ENTER to roll the dice")
ENTER=input()
diceNumber()        # Rolls the dice twic
diceNumber2 = diceScore
if diceNumber1 == diceNumber2:     # All
  print("You got a DOUBLE!")
  print("Press ENTER to roll the dice")
  ENTER=input()
  diceNumber()
```

```
 Player 1, you play first


Press ENTER to Start

Are you ready!
Lets roll the dice!....3...2...1..begin!


Your Dice Score is 1
Its an ODD number!
Your score at the end of round 5 is 46
Press ENTER to roll the dice

Your Dice Score is 1
Its an ODD number!
Your score at the end of round 5 is 42
You got a DOUBLE!
Press ENTER to roll the dice

Your Dice Score is 4
Its an EVEN number!
Your current score in this round is 56
```
</td>
</tr>
<tr>
<td>7.</td>
<td>Both the players should get one extra die if their scores are equal. The program should repeat until one of the player gets higher points rolled than the other player.</td>
<td>If statement is used to validate the scores of player 1 and 2 at the end of the 5 rounds. If the scores are equal then both the players roll a dice each where a function DicRoll() is declared and called that allows the player to roll a 6-sided dice. The program validates their scores through if statements after they each roll the dice and proceeds to the leader board if their dice scores are different as the</td>
<td>

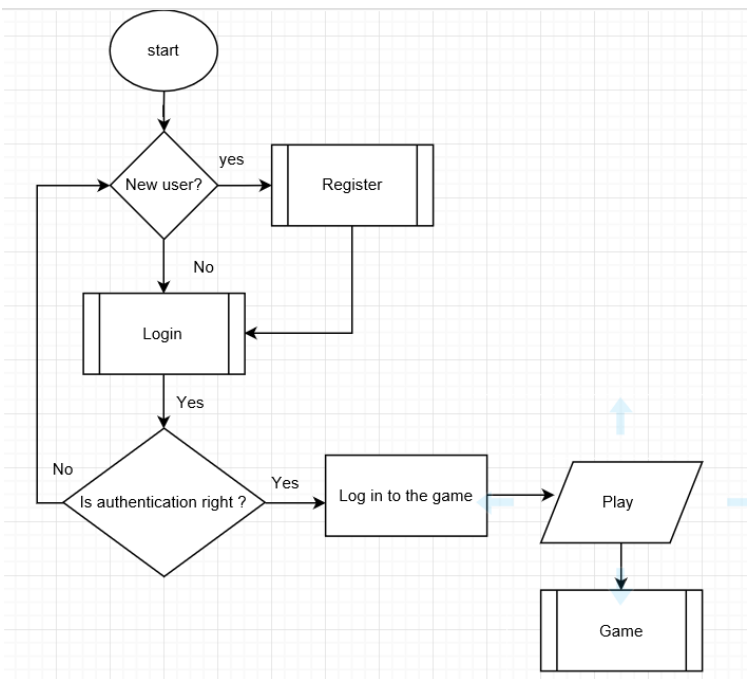
```
elif player1Score==player2Score:      # If scores are equal of both player

    print("Your scores are equal")
    print("\n")
    print("Both of you will roll the dice again")
    print("\n")
    print("Player 1, your turn")
    print("\n")
    print(" Are you ready!")
    print("Lets roll the dice!....3...2...1..begin!")
    print("\n")
    DiceRoll()       # Rolls the dice
    Player1=number
    print("Player 2, your turn")
    print("\n")
    print(" Are you ready!")
    print("Lets roll the dice!....3...2...1..begin!")
    print("\n")
    DiceRoll()      # Rolls the dice
    Player2=number
    if Player2==Player1:
      Winners()
```
</td>
</tr>
</table>

		Leaderboard() function is called.	
8.	The name and score of the player who won should be outputted	The variable Winner and WinnerScore are declared as global variable names. The winner's name is assigned to the global variable Winner and the winner's score is assigned to the global variable WinnerScore. Print function is used to output the player's name and score who has won.	
9.	The winning players name and score should be stored in an external file and the score and player name of the top 5 winning scores should be outputted from the external file.	An external text file called Leaderboard.txt is created. The variables Winner and WinnerScore are written to the text file. A list names Scores[] is declared. A dictionary named Leaderboard is created where the Winners' names are stored under the column Winner Name and Winners' score under the column Dice Score. The dictionary Leaderboard is appended into the list named scores. The Dice Scores column is sorted through the sorted function and lambda into descending order. If statement is used to select the top 5 highest scorers and output their names and score through the print function.	 

Resolved and Unresolved Issues.

Login

Initially when planning the program in the design stage the pseudocode conveyed an overview of the players' login and register. The program was initially planned to register any player who is not an existing user then allow the player to login.



However, while developing the code it was discovered that the login function doesn't authenticate the user and loops the program even if the username and password are correct and doesn't proceed to the game. The issue was left unresolved until the end of the development stage. The code only allowed the players to create a new account even if they are an existing user as the program doesn't authenticate the username and password by reading the text file PlayerDetails.txt projecting a logical error. The game didn't meet one of the success criteria i.e. user authentication. As the login was declared as a function it didn't have an overall impact on the code as it wasn't called when testing. The test plan was conducted by registering the players into the game and excluded the testing of the login function. In the beginning, it was planned that if the username and password are correct then the player would have to wait for the other player to login/register or would proceed to the game and if the username or password (or both) are incorrect then the program would proceed to the start function where the player has a choice to either login/ register

```
def Login():

    print('-------------Welcome to the login page----------------')


    username=input('username: ')    # Asks the user to input username

    password=input('Password: ')    # Asks the user to input password

    for line in open ('playerDetails.txt','r').readlines():
        logininformation=line.split(",")
        if username==logininformation[1] and password==logininformation[2]:
            print("Correct credentials")


        else:
            print ("Incorrect Credentials")
            Start()
```

Logical error Identified

.

```
Are you a new user ? Y or N
Please choose an option: N
-------------Welcome to the login page----------------
username: Max156
Password: Kablan
Incorrect Credentials
----------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option:
```

Logical error displayed as "Incorrect Credentials "is outputted even if the details inputted are correct

playerDetails - Notepad

File Edit Format View Help

Rishvanth,Ric151,Rishy15
Maxwell,Max156,Kablan
David,Dav249,Stephen

The code outputs a logical error even if the player inputs correct username and password

However, the issue was resolved after the development of the code was completed. The logical error was identified and the reason for the error was highlighted. In the initial code, if the user authentication is valid the program doesn't move onto the game as their isn't any function called and would eventually proceed to the else statement where it loops the message Incorrect credentials every time the player inputs username and password even though they are incorrect. The issue has been resolved through changing the if statements. The if statement in the resolved version initially validates if the details are wrong. The else statement validates if the user details are correct and, in this case, doesn't require any function to be called as it directly proceeds to the next segment of the code. Eventually, the issue with the login function is resolved and is well tested.

```
def Login():

    print('-------------Welcome to the login page----------------')


    username=input('username: ')     # Asks the user to input username

    password=input('Password: ')     # Asks the user to input password

    for line in open ('playerDetails.txt','r').readlines():
        logininformation=line.split(",")
        if username!=logininformation[1] and password!=logininformation[2]:
            print ("Incorrect Credentials")
            Start()

        else:
            print("Correct credentials")
```

Code modified

```
 Player 1


----------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option: N
-------------Welcome to the login page----------------
username: Ris151
Password: Rishy15
Correct credentials
----------------Welcome  to DROP DEAD ---------------------


Are you a new user ? Y or N
Please choose an option:
```

The issue is resolved after the code is modified

playerDetails - Notepad

File Edit Format View Help

Rishyanth,Ris151,Rishy15
Maxwell,Max156,Kabian
David,Dav249,Stephen

The code successfully reads the text file to read the username and password inputted.

Leader Board

The leader board was initially planned to store the names and scores of the winners that would update after every game and would be sorted to output the top 5 highest scorers. However, the scores were stored in a text file named leaderboard.txt and couldn't sort the dice scores in a descending order. Therefore, the leader board displayed an unsorted list of 5 winners whose names and scores were not sorted in order and projected a logical error. This eventually prevented in meeting one of the success criteria of the project: Outputting top 5 winning scores from an external file. Arguably, the program didn't meet the user requirement and was left as a significant issue yet to be resolved.

```
def Leaderboard():                      # Leaderboard function stores the winner name and score onto the text file Leaderboard.txt

  file=open("Leaderboard.txt","a")
  file.write(Winner)                    # Appends the winner name to the text file
  file.write(" ")
  file.write(WinnerScore)               # Appends the winner score to the text file
  file.write("\n")
  file.close()

  scores=[]
  for line in open('Leaderboard.txt','r').readlines():      # Reads the leaderboard tect file.
    info = line.split(' ')                                  # Splits the lines in the tect file.
    LeaderBoard={('Dice_Score'):int(info[1]),('Winner_Name'):info[0]}
    scores.append(LeaderBoard)        # Appends the dice scores to the list named scores

    # Dictonary stores the list of dice scores and winner names

    scores = sorted(scores, key=lambda s: s['Dice_Score'],reverse=True)    # Sortes the winner scores in the list "scores" into descending order


    # Removes the inverted commas and unecessary text from the list

    print (scores)
```

Initial code for the leader board

```
---------THESE ARE OUR TO 5 SCORERS----------

   =     =====     =      =====  ====  =====  =====   ===      =      =====    =====
   =     =        ==      =    = =     =   =  =   =  =   =    ===     =   =    =    =
   =     ====    = = =    =    = ===   = =    ====   =   =   =====    = = =    =    =
   =     =      =     =   =    = =     =   =  =   =  =   =  =     =  =   =    =    =
   ====  ===== =       =  =====  ====  =    = = = =   ===   =     =  =    =  =====

[{'Dice_Score': 158, 'Winner_Name': 'Neer'}]
[{'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 138, 'Winner_Name': 'Maxwell'}]
[{'Dice_Score': 164, 'Winner_Name': 'Rishy'}, {'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 138, 'Winner_Name': 'Maxwell'}]
[{'Dice_Score': 164, 'Winner_Name': 'Rishy'}, {'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 138, 'Winner_Name': 'Maxwell'}, {'Dice_Score': 98,
[{'Dice_Score': 164, 'Winner_Name': 'Rishy'}, {'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 138, 'Winner_Name': 'Maxwell'}, {'Dice_Score': 122,
[{'Dice_Score': 164, 'Winner_Name': 'Rishy'}, {'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 144, 'Winner_Name': 'Djrandji'}, {'Dice_Score': 138
_Score': 98, 'Winner_Name': 'Imran'}]
[{'Dice_Score': 164, 'Winner_Name': 'Rishy'}, {'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 144, 'Winner_Name': 'Djrandji'}, {'Dice_Score': 138
Dice_Score': 122, 'Winner_Name': 'David'}, {'Dice_Score': 98, 'Winner_Name': 'Imran'}]
[{'Dice_Score': 164, 'Winner_Name': 'Rishy'}, {'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 144, 'Winner_Name': 'Djrandji'}, {'Dice_Score': 138
Dice_Score': 122, 'Winner_Name': 'David'}, {'Dice_Score': 106, 'Winner_Name': 'China'}, {'Dice_Score': 98, 'Winner_Name': 'Imran'}]
[{'Dice_Score': 164, 'Winner_Name': 'Rishy'}, {'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 144, 'Winner_Name': 'Djrandji'}, {'Dice_Score': 138
Dice_Score': 124, 'Winner_Name': 'Willyiam'}, {'Dice_Score': 122, 'Winner_Name': 'David'}, {'Dice_Score': 106, 'Winner_Name': 'China'}, {'Dice_Score': 98, '
[{'Dice_Score': 164, 'Winner_Name': 'Rishy'}, {'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 144, 'Winner_Name': 'Djrandji'}, {'Dice_Score': 138
Dice_Score': 124, 'Winner_Name': 'Rishyanth'}, {'Dice_Score': 124, 'Winner_Name': 'Willyiam'}, {'Dice_Score': 122, 'Winner_Name': 'David'}, {'Dice_Score': 1
[{'Dice_Score': 164, 'Winner_Name': 'Rishy'}, {'Dice_Score': 158, 'Winner_Name': 'Neer'}, {'Dice_Score': 144, 'Winner_Name': 'Djrandji'}, {'Dice_Score': 138
Dice_Score': 124, 'Winner_Name': 'Rishyanth'}, {'Dice_Score': 124, 'Winner_Name': 'Willyiam'}, {'Dice_Score': 122, 'Winner_Name': 'David'}, {'Dice_Score': 1
```

Logical error displayed

```
ief Leaderboard():                      # Leaderboard function stores the winner name and score onto the text file Leaderboard.txt

  file=open("Leaderboard.txt","a")
  file.write(Winner)                    # Appends the winner name to the text file
  file.write(" ")
  file.write(WinnerScore)               # Appends the winner score to the text file
  file.write("\n")
  file.close()

  scores=[]
  for line in open('Leaderboard.txt','r').readlines():      # Reads the leaderboard tect file.
    info = line.split(' ')                                  # Splits the lines in the tect file.
    LeaderBoard={('Dice_Score'):int(info[1]),('Winner_Name'):info[0]}
    scores.append(LeaderBoard)        # Appends the dice scores to the list named scores

    # Dictonary stores the list of dice scores and winner names

    scores = sorted(scores, key=lambda s: s['Dice_Score'],reverse=True)    # Sortes the winner scores in the list "scores" into descending order


    rank=str(scores[number])      # Assigns the the top 5 player's name and score to the variable rank
    location=str(ranks)

    print((location+'.'),rank[12:-1])           # Outputs the player's name and score with top score
    ranks+=1                                    # Appends 1 to the the variable ranks to move to the next player with highest score.
```

The code was further modified to output the top 5 players in ranks to prevent logical errors of looping the leader board

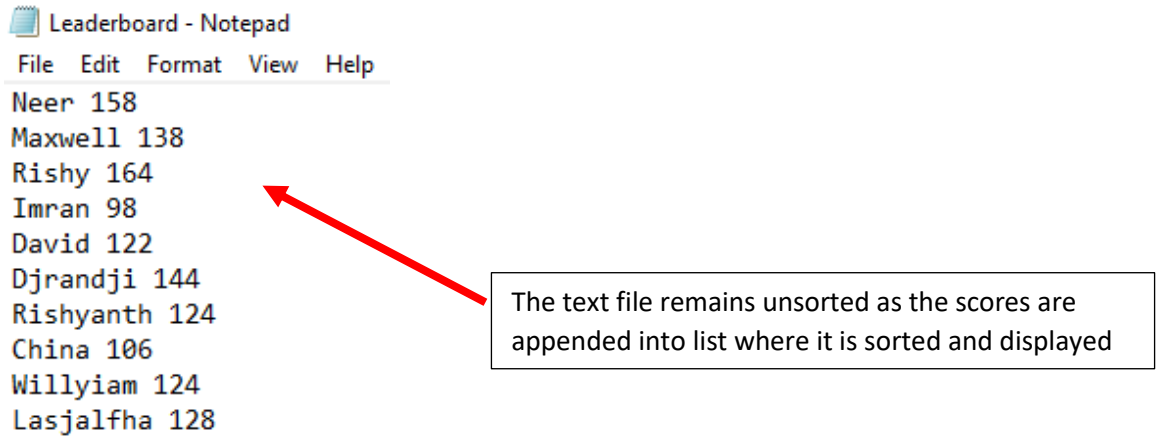
```
---------THESE ARE OUR TO 5 SCORERS----------

   =     =====     =      =====  ====  =====  =====   ===      =      =====    =====
   =     =        ==      =    = =     =   =  =   =  =   =    ===     =   =    =    =
   =     ====    = = =    =    = ===   = =    ====   =   =   =====    = = =    =    =
   =     =      =     =   =    = =     =   =  =   =  =   =  =     =  =   =    =    =
   ====  ===== =       =  =====  ====  =    = = = =   ===   =     =  =    =  =====

Traceback (most recent call last):
  File "\\sfi01\intake16\sjc16187\Desktop\Component 3- Programming Project\NEA Code.py", line 397, in <module>
    Verification()
  File "\\sfi01\intake16\sjc16187\Desktop\Component 3- Programming Project\NEA Code.py", line 357, in Verification
    Leaderboard()
  File "\\sfi01\intake16\sjc16187\Desktop\Component 3- Programming Project\NEA Code.py", line 302, in Leaderboard
    location=str(ranks)
UnboundLocalError: local variable 'ranks' referenced before assignment
```

Syntax error displayed as the local variable 'ranks' is called before assignment.

The program was tested numerous times and was eventually planned to extract the dice score and winner name into a dictionary called Leaderboard that is appended into a list named Scores where the Dice Score is sorted in order through sort function. This eventually allowed to display the top 5 highest scorers however didn't sort the leader board text file. Arguably, the code still meets the user requirements and success criteria therefore couple of minor changes were made to the initial planning of the program.

```
scores=[]
for line in open('Leaderboard.txt','r').readlines():      # Reads the leaderboard tect file.
  info = line.split(' ')                                  # Splits the lines in the tect file.
  LeaderBoard={('Dice_Score'):int(info[1]),('Winner_Name'):info[0]}
  scores.append(LeaderBoard)        # Appends the dice scores to the list named scores

  # Dictonary stores the list of dice scores and winner names

  scores = sorted(scores, key=lambda s: s['Dice_Score'],reverse=True)    # Sortes the winner scores in the list "scores" into descending or

  lines=int(len(scores))        # Variable lines is assigned the length of thre
  if lines > 5:                 # If the players are more than 5 then select the top 5 players from the list.
      lines=5
  ranks=1
for number in range(lines):
```

```
---------THESE ARE OUR TO 5 SCORERS----------

   =     =====     =      =====  ====  =====  =====   ===      =      =====    =====
   =     =         ==     =    = =     =   =  =   =  =   =    ===     =   =    =    =
   =     ====    = = =    =    = ===   = =    ====   =   =   =====    = = =    =    =
   =     =      =     =   =    = =     =   =  =   =  =   =  =      =  =   =    =    =
   ====  ===== =       =  =====  ====  =    = = = =   ===   =      =  =     =  =====

1. Rishy Dice_Score164
2. Neer Dice_Score158
3. Djrandji Dice_Score144
4. Maxwell Dice_Score138
5. Lasjalfha Dice_Score128
```



Conclusion

In conclusion, the code meets all the success criteria and user requirements. This has been made easier through problem decomposition that significantly highlighted the requirements for each stage in the development. The Testing of each stage ensured that the program met all the success criteria. It was planned to be efficient and less memory intensive through declaring functions for each stage of development. This centralised the code as more organized through less lines of codes as calling function reduced repeating the same code again and again. Every stage of the project was done through setting timelines that helped deliver the project on time and skip the stages that had unresolved issues. The document contains table of contents that illustrates the project as more organized and accessible for a user to go through the document. Each stage of the code is well annotated through comments in the code and other annotations and the tested to convey the program is running successfully and as expected by the user. A table containing the success criteria of the project and how they are met are demonstrated with evidence of testing to convey proof that the code has met all the success criteria. Issues that were unresolved while developing the code are highlighted and clearly demonstrated by the way they are resolved through testing. Majority of the project has been delivered as initially planned with few limitations of unresolved issues with the player login which is later resolved and well-tested to prevent any logical and syntax error in the code.