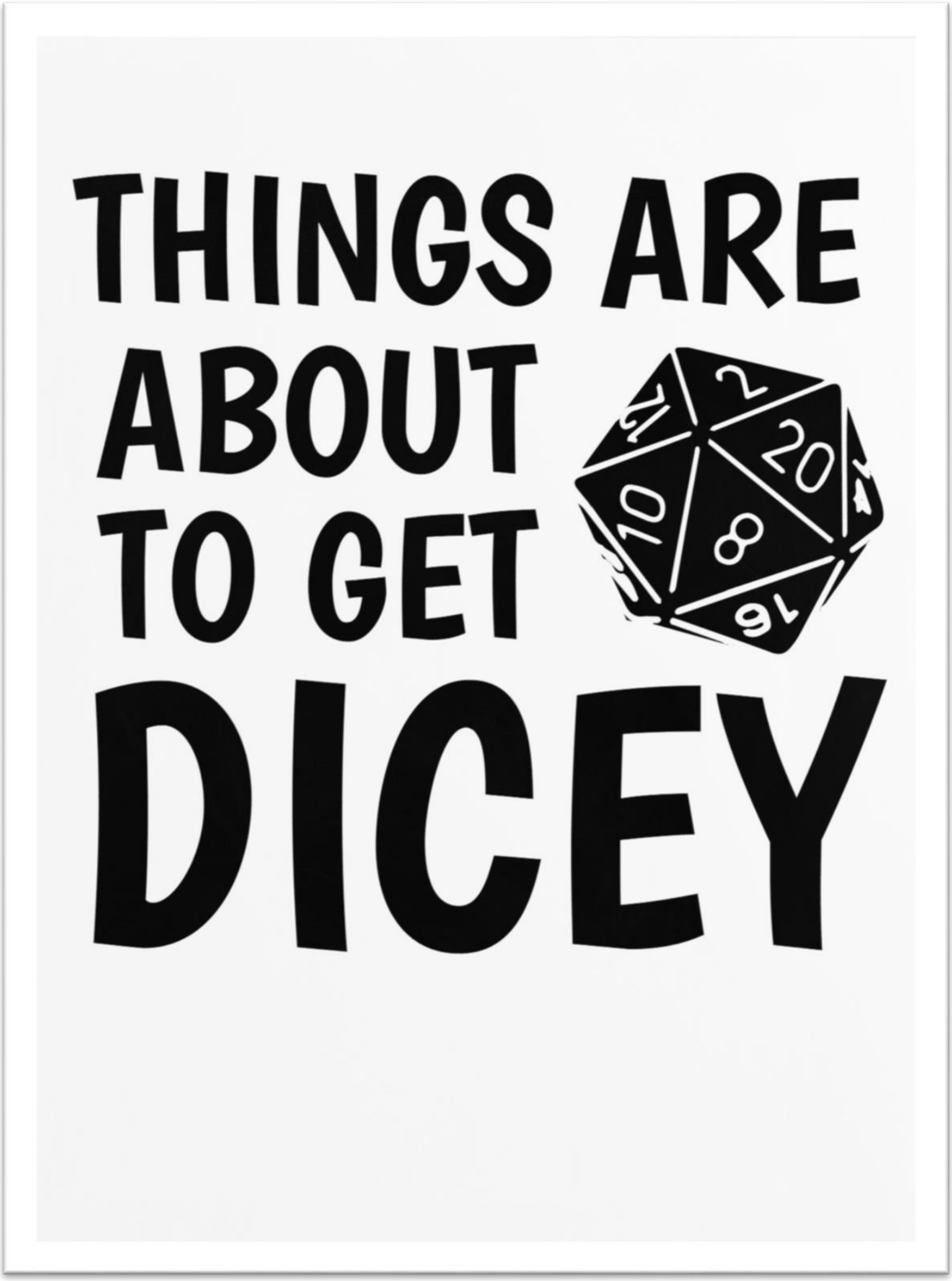
COMPUTER SCIENCE PROGRAMMING PROJECT



TOPIC: TWO PLAYER DICE GAME

Name: Sai Rishyanth Visinigiri

Class: XII A

School: Navy Children School, Mumbai

CERTIFICATE

This is to certify that Mr. /Ms ____________________________
of std. XI Navy Children School, Mumbai has put in sincere
and dedicated efforts towards the project. His/her work has
been highly appreciative.

______________________________ ______________________________

Sign of Head of department Sign of Principal

____

School stamp

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher Ms Sapna Boora as well as our principal Ms M Subramaniam who gave me the golden opportunity to do this wonderful project in computer science on the topic dice game, which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to them.

Secondly I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Table of Contents

1. PROJECT DESCRIPTION

1.2 Requirements

Drop Dead is 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.

1.3 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.

1.4 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 a database.

2. IMPLEMENTATION

2.1 File Handling

The concepts of file handling have been applied while designing the program. The user details while registering into the game will be stored in an external text file “playerDetails.txt”. The user’s name, age, username, and password would be written in the file while registering. On the other hand, while logging into the game the text file is read to authenticate the login details entered by the user.

2.2 User-defined Functions

Numerous user defined functions will be used for partitioning off the program into well-defined boundaries. This greatly reduces complexities to a large extent and reduces time to rewrite the code. Such as Login(), Register(), Player1(), Player2(), Verification(), Leaderboard() functions would be used in the design and development of the program.

2.3 Random Module

Random module would be imported in the program for random number generation. The randint() function in the random module would generate a random number between 1 and 6 representing a 6-sided dice.

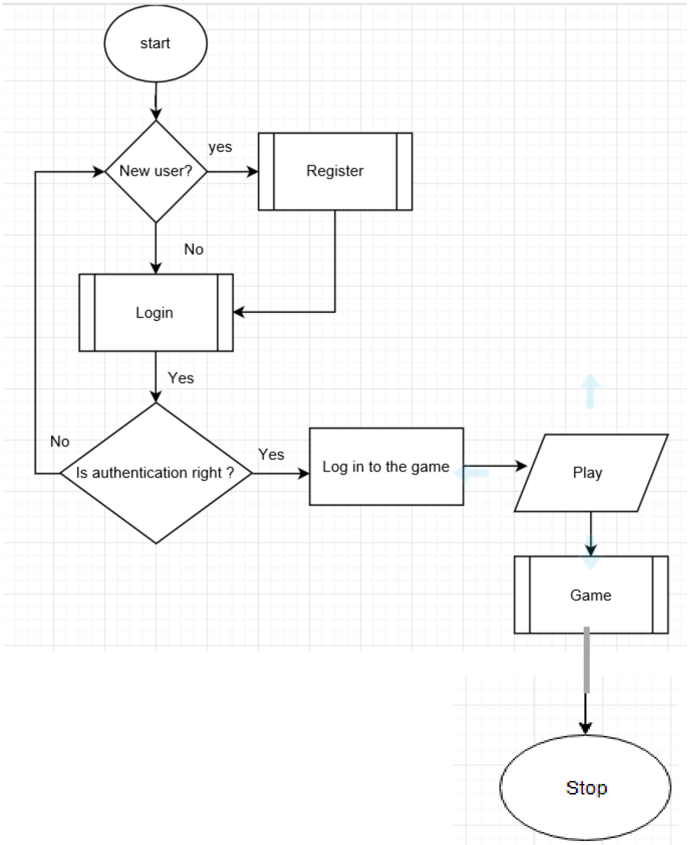
2.4 Python Interfacing with MYSQL

Python interfacing with MYSQL is used in the leader-board part of the program where the details of the winner of the game would be stored in a database. Then, the records from the table Leader-board would be selected that are ordered in the descending order, from which the five highest scorers would be retrieved using the fetchmany() function.

3. DESIGN

3.1 User Details and Authentication

Flow chart

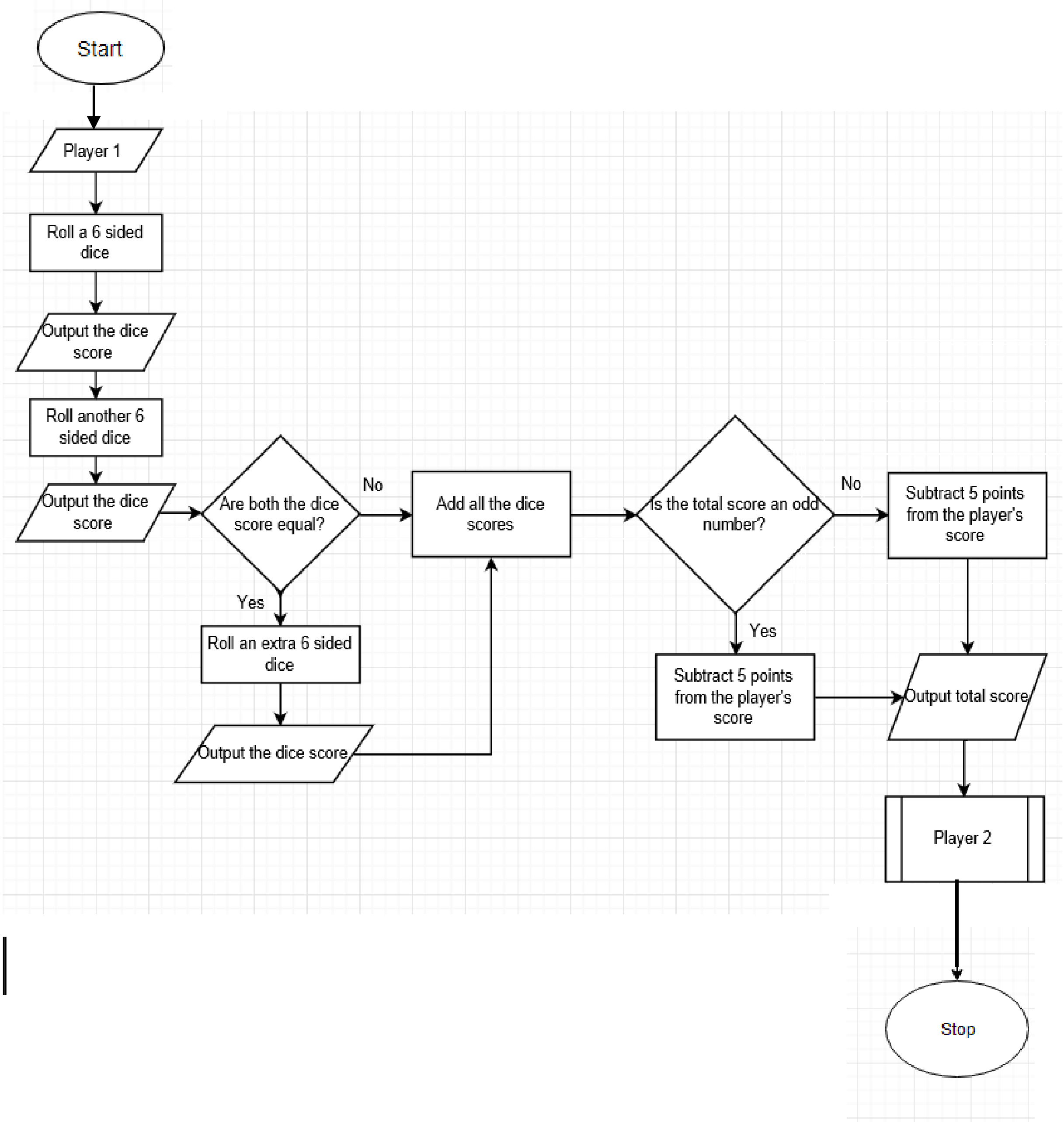


```
FUNCTION Login ()
   WHILE TRUE
       INPUT username
       INPUT password
       OPEN file "playerDetails.txt"
       Split the line in the text file by ('   ')
       READFILE "playerDetails.txt"
       IF the username and password equals to any first & second words in the csv files THEN
    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

Pseudocode

```
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
       ENDWHILE
END FUNCTION
```

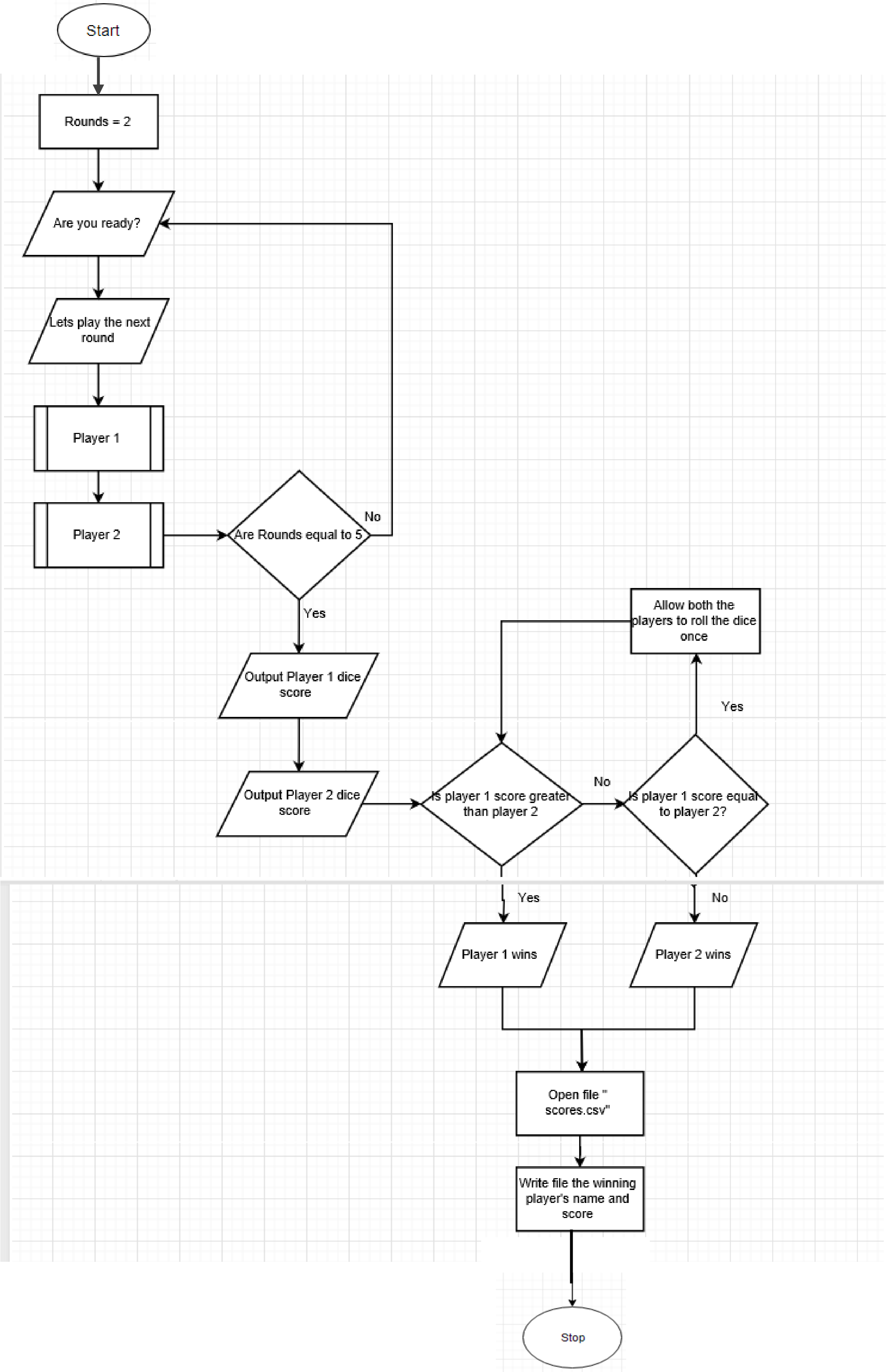
3.2 Calculation of the score –Flowchart


Start
Player 1
Roll a 6 sided dice
Output the dice score
Roll another 6 sided dice
Output the dice score
Are both the dice score equal?
No
Yes
Roll an extra 6 sided dice
Output the dice score
Add all the dice scores
Is the total score an odd number?
No
Yes
Subtract 5 points from the player's score
Subtract 5 points from the player's score
Output total score
Player 2
Stop

3.2 Calculation of the score – Pseudocode

```
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()
```

3.3 Iteration and displaying the winner - Flowchart


Start
Rounds = 2
Are you ready?
Lets play the next round
Player 1
Player 2
Are Rounds equal to 5
No
Yes
Output Player 1 dice score
Output Player 2 dice score
Is player 1 score greater than player 2
No
Is player 1 score equal to player 2?
Yes
Allow both the players to roll the dice once
Yes
No
Player 1 wins
Player 2 wins
Open file " scores.csv"
Write file the winning player's name and score
Stop

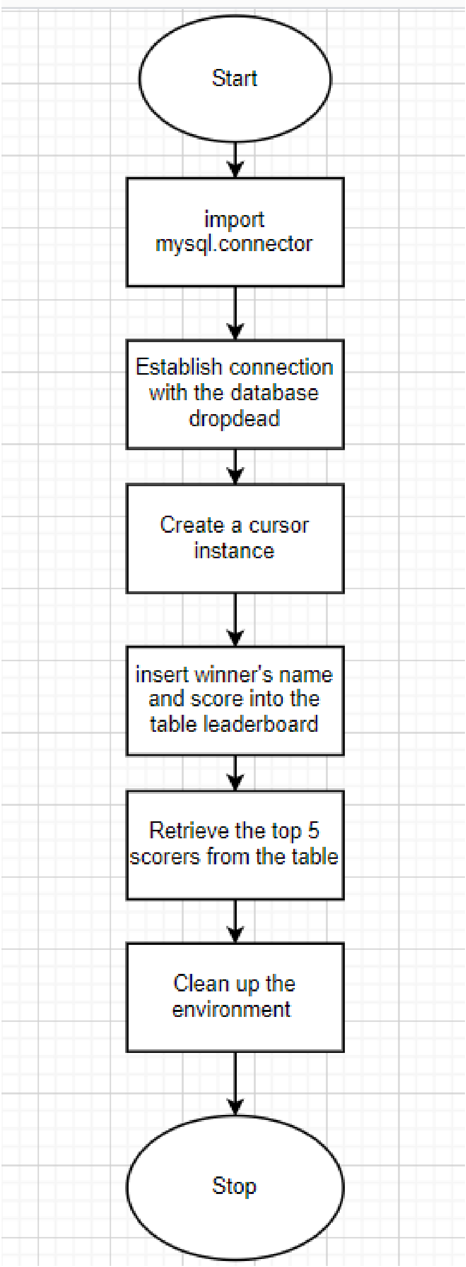
3.3 Iteration and displaying the winner - Pseudocode

```
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

   ELSE
       OUTPUT "Player 2 wins"
               OUTPUT Player 2 name and their score
               Top5Players ()
               Break
    ENDIF
END FUNCTIONS
Winner ()
```

3.4 Leader board

Flow chart:

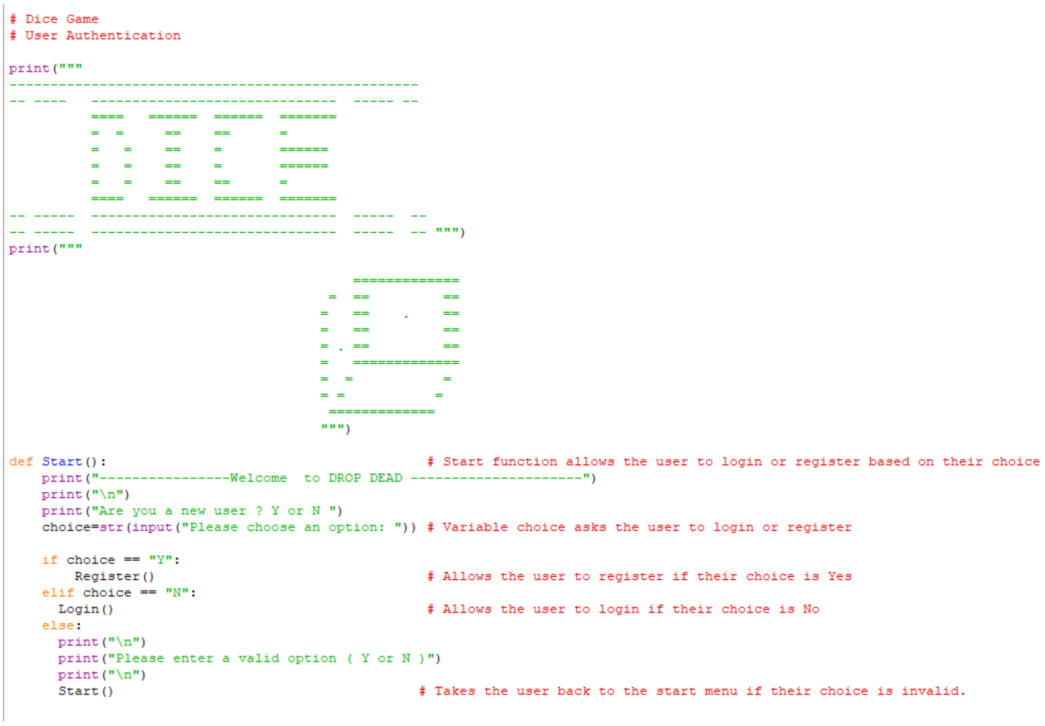

Start
import mysql.connector
Establish connection with the database dropdead
Create a cursor instance
insert winner's name and score into the table leaderboard
Retrieve the top 5 scorers from the table
Clean up the environment
Stop

Python interfacing with MySQL:

1. Start Python
2. Import database programming package
3. Establish connection with the database drop-dead.
4. Create a cursor instance
5. Insert into the table Winner's name and score
6. Execute the query
7. Select rows from the table in descending order of the score using fetchmany function.
8. Execute the query
9. Obtain the top 5 scorers
10. Clean up the environment

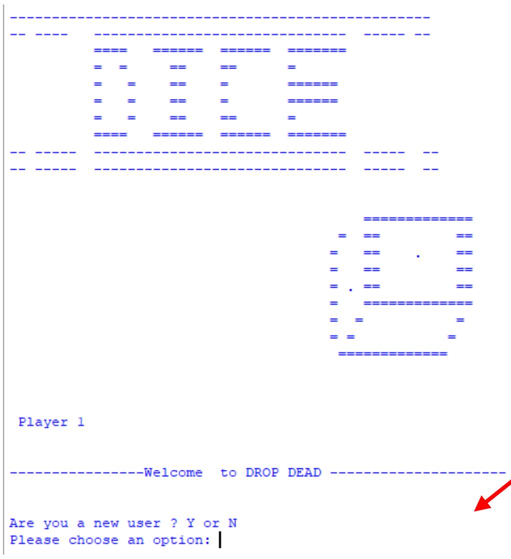
4. DEVELOPMENT

4.1 Main Menu



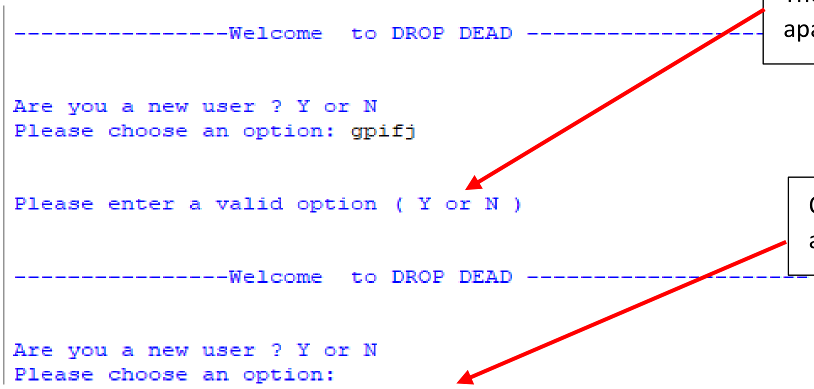
```
# Dice Game
# User Authentication

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.
```

Output of def Start () function:



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


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


 Player 1


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


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

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

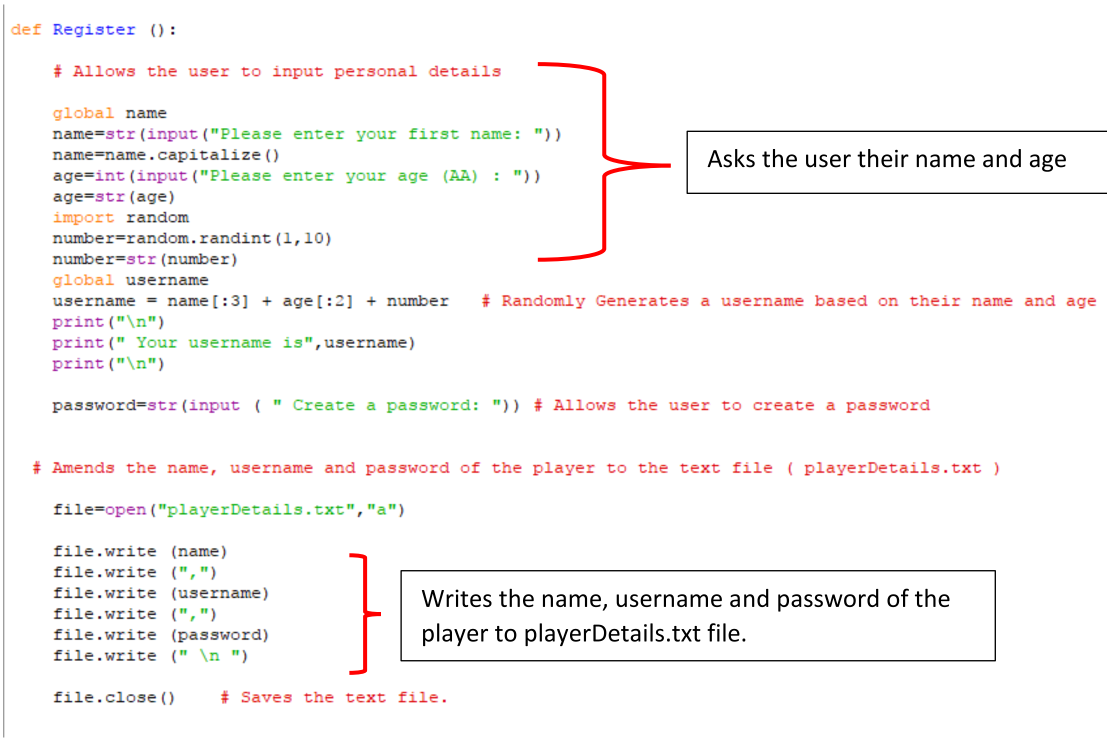


```
----------------Welcome  to DROP DEAD ---------------------


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

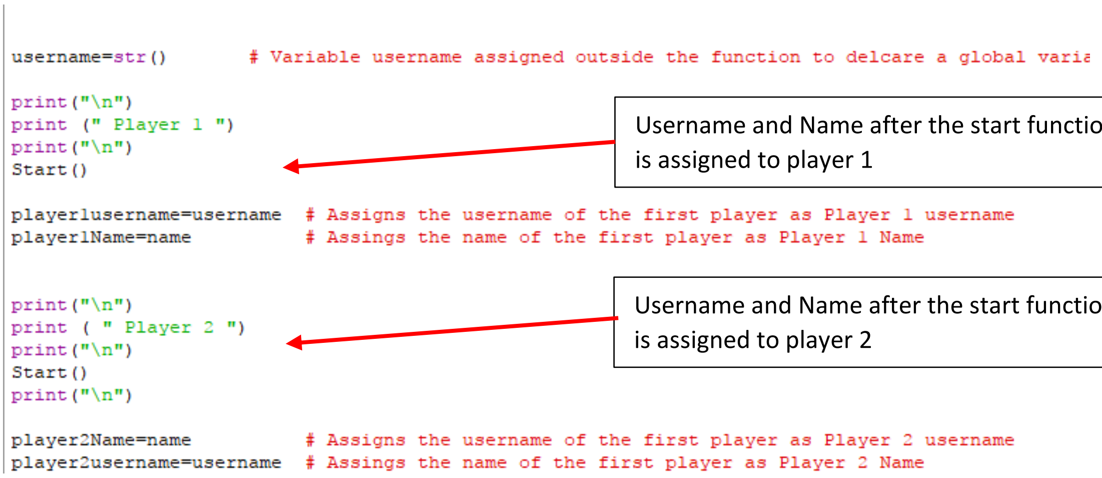
4.1.1 Register



```
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.

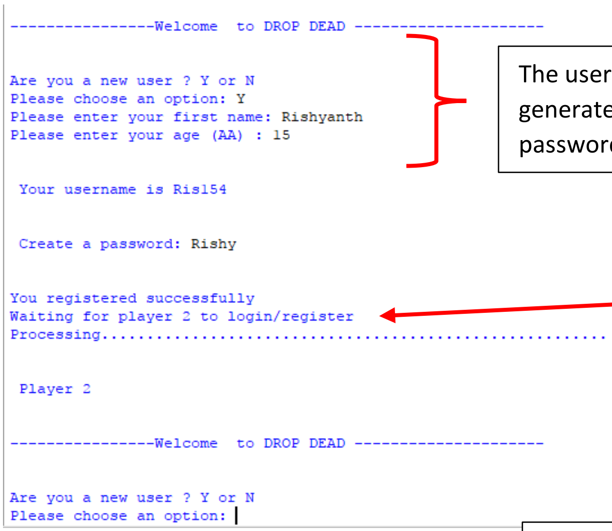


```
username=str()       # Variable username assigned outside the function to delcare a global varia

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

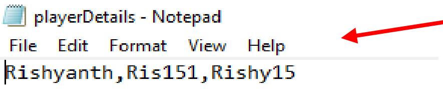
Output of def Register():



```
----------------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.

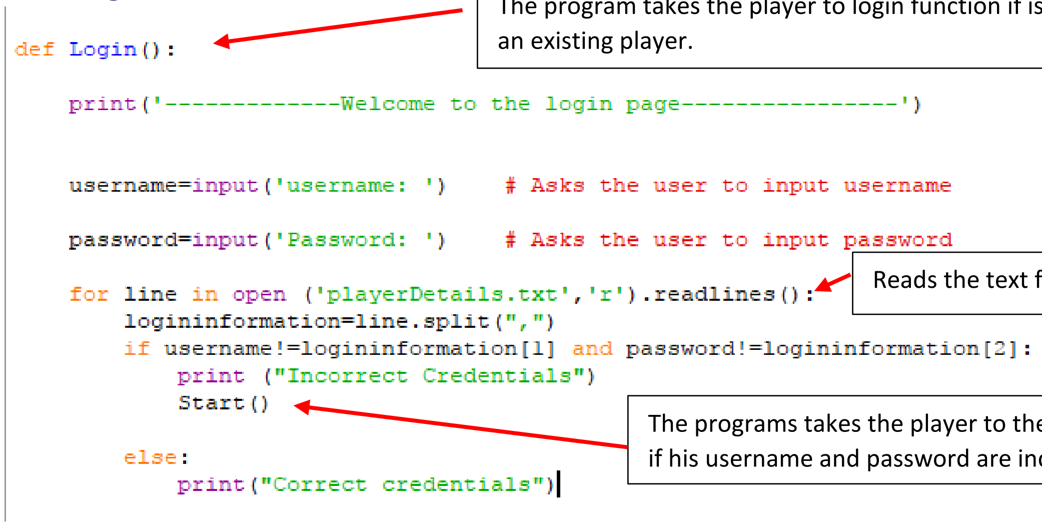
Program shifts to the main menu for player 2 to login/register if player has registered successfully.



```
Rishyanth,Ris151,Rishy15
```

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

4.1.2 Login



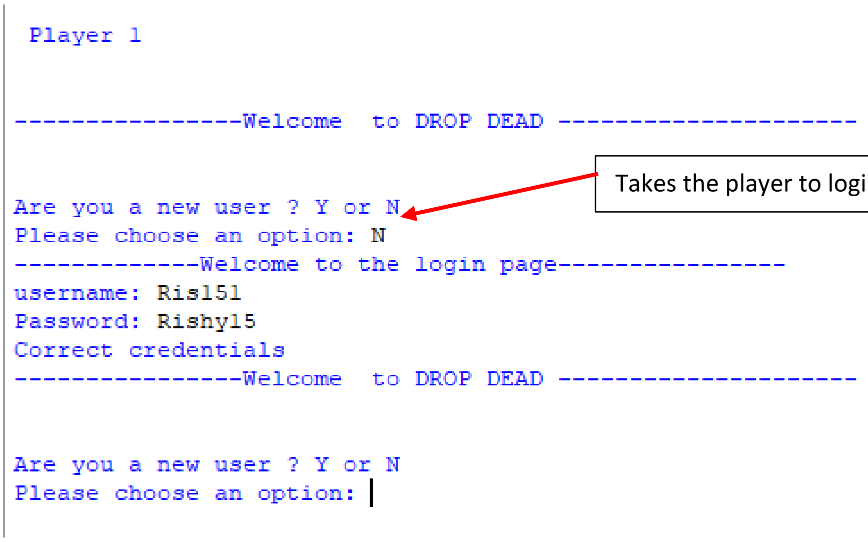
```
def Login():

    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”.

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

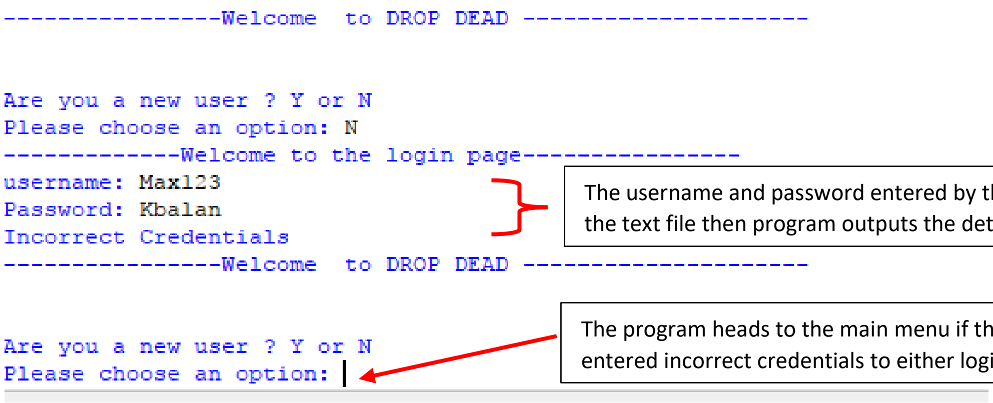
Output: The username and password are correct



```
 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

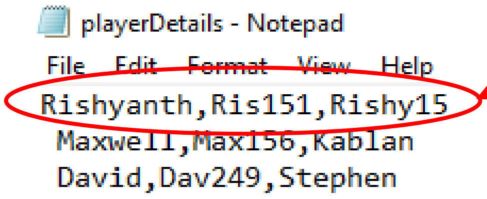
Output: The username or password (or both) are incorrect



```
----------------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

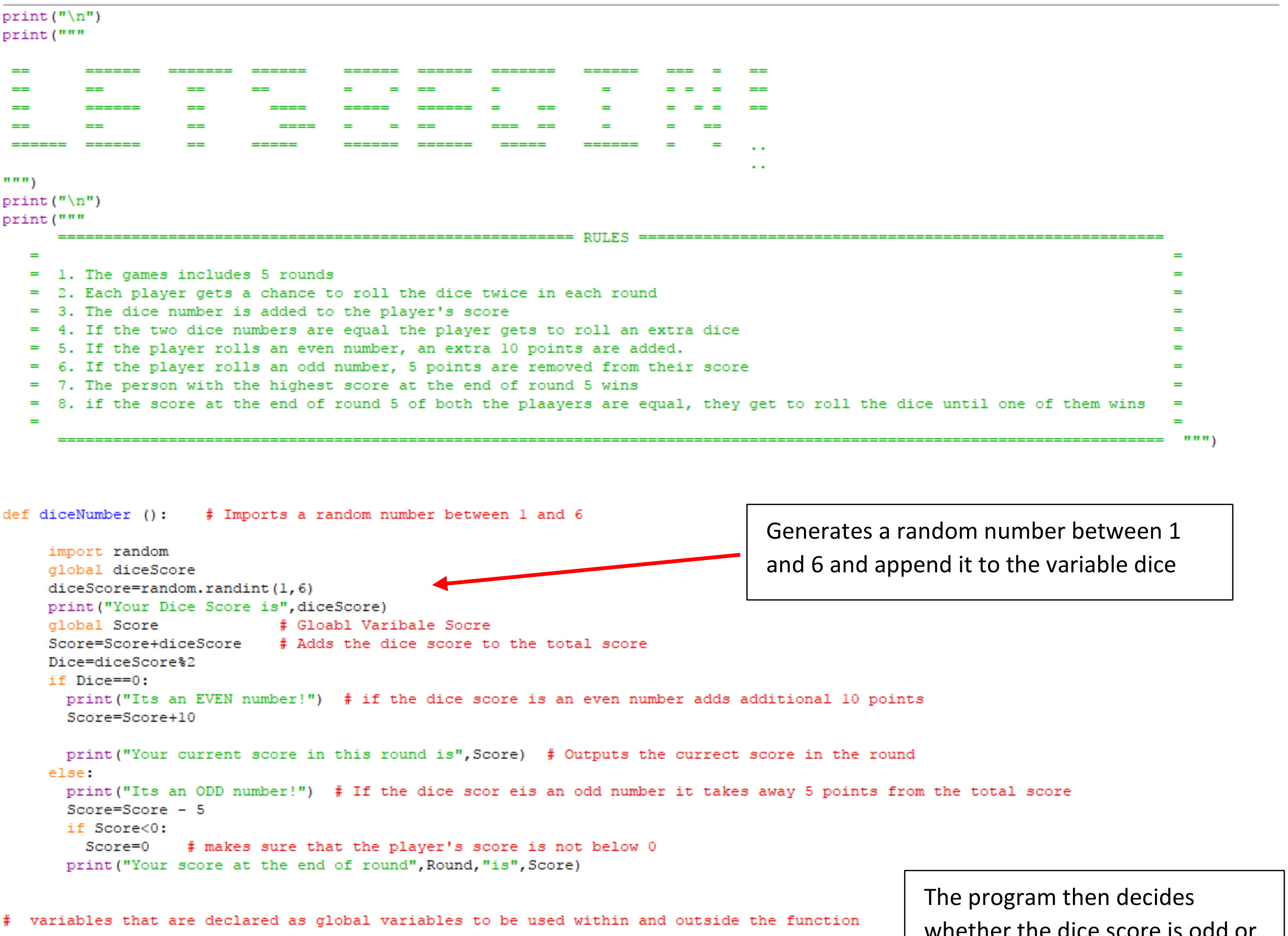
The program heads to the main menu if the player entered incorrect credentials to either login again or



```
Rishyanth,Ris151,Rishy15
Maxwell,Max156,Kablan
David,Dav249,Stephen
```

Reads the text file to authenticate for valid username and password

4.2 Game Rules and Random Module

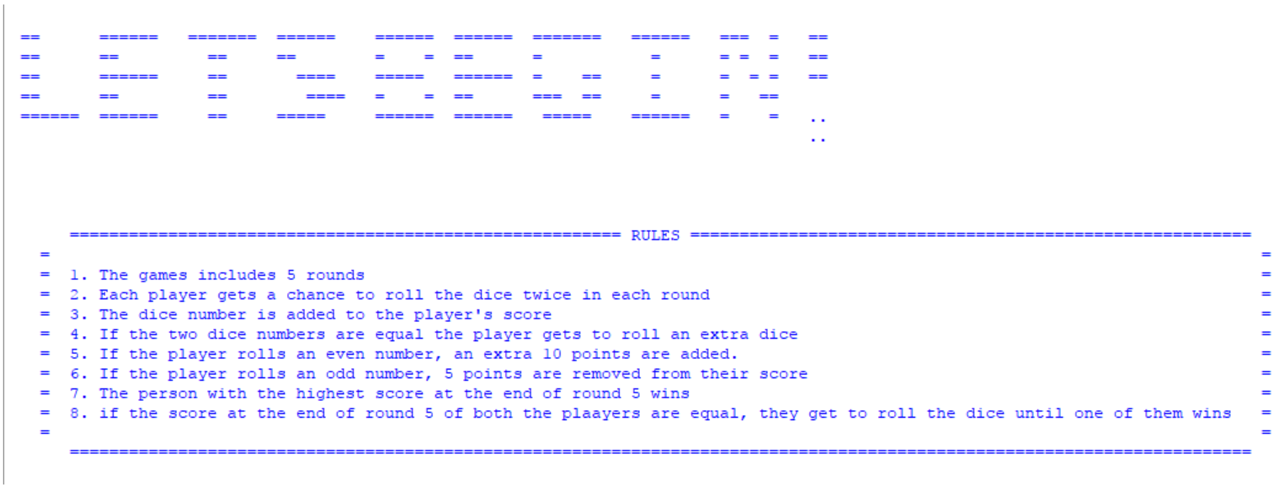


```
print("\n")
print("""

 ==      ======   =======  ======    ======  ======  =======   ======   ===  =   ==
 ==      ==         ==     ==        =    =  ==      =           =      = =  =   ==
 ==      ======     ==       ====    =====   ======  =    ==     =      =  = =   ==
 ==      ==         ==        ====   =    =  ==      ===  ==     =      =   ==
 ======  ======     ==     =====     ======  ======   =====    ======   =    =   ..
                                                                                 ..
""")
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    =
   =                                                                                                                            =
      ========================================================================================================================  """)


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)

#  variables that are declared as global variables to be used within and outside the function
```

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

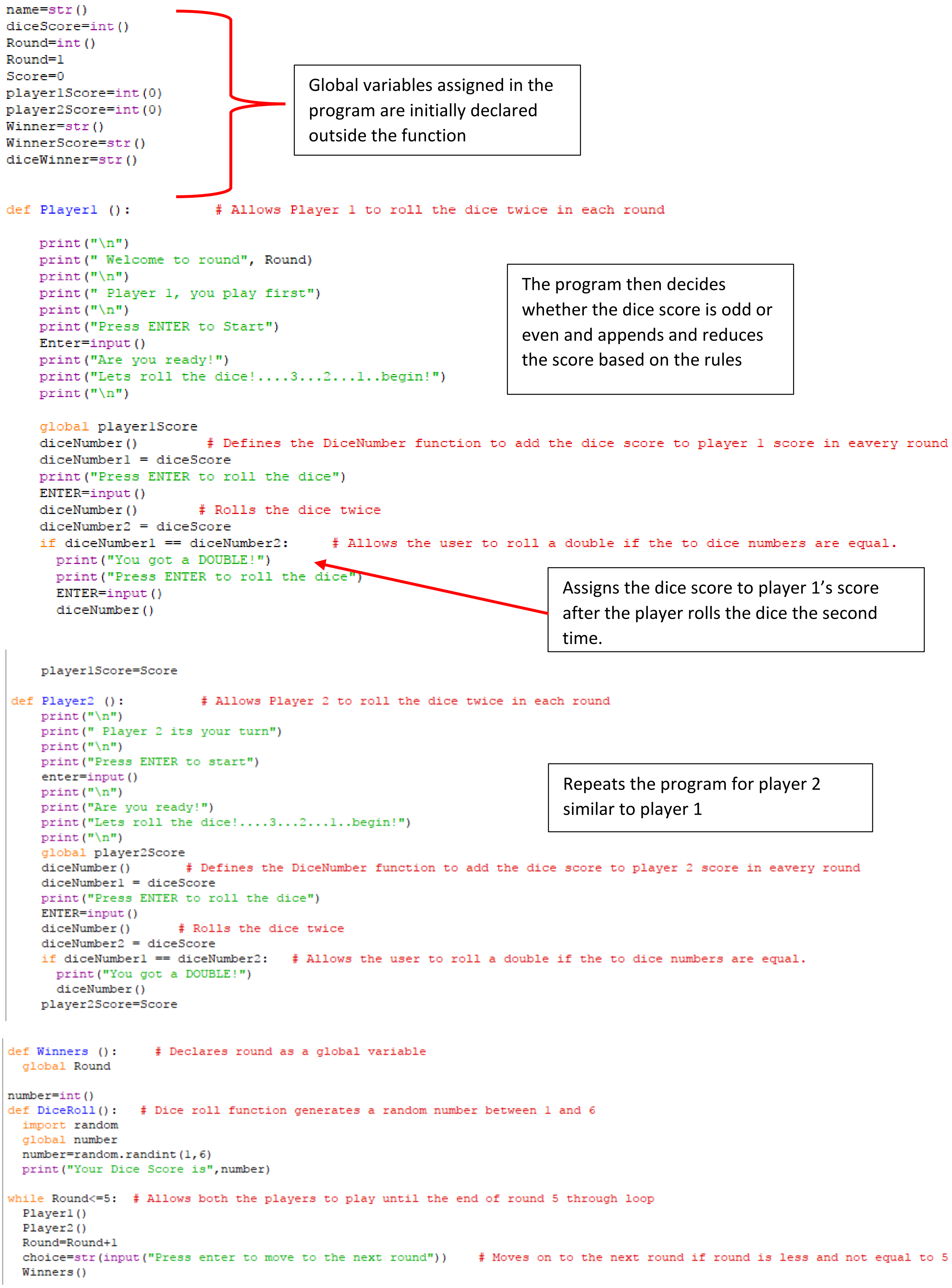
The program then decides whether the dice score is odd or even and appends and reduces the score based on the rules

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    =
   =                                                                                                                            =
      ========================================================================================================================
```

4.3 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
```

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

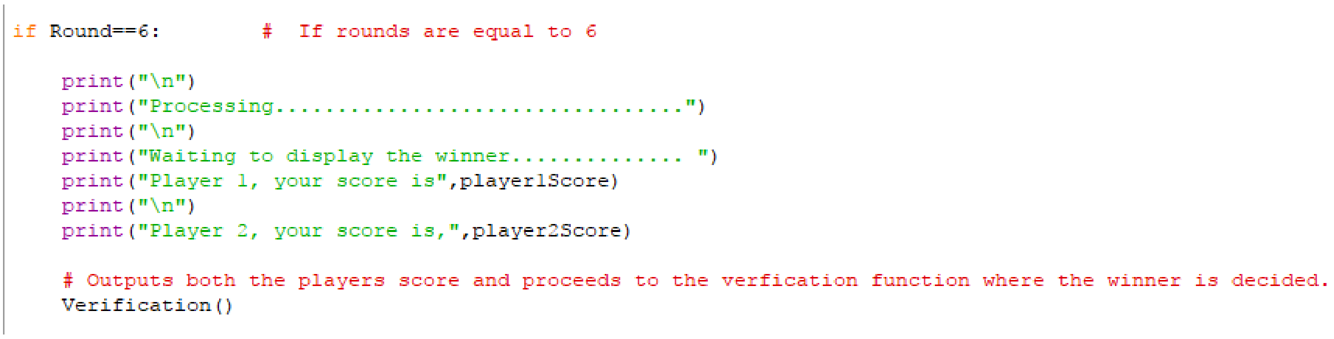
Assigns the dice score to player 1's score after the player rolls the dice the second time.

```
def 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
```

Repeats the program for player 2 similar to player 1

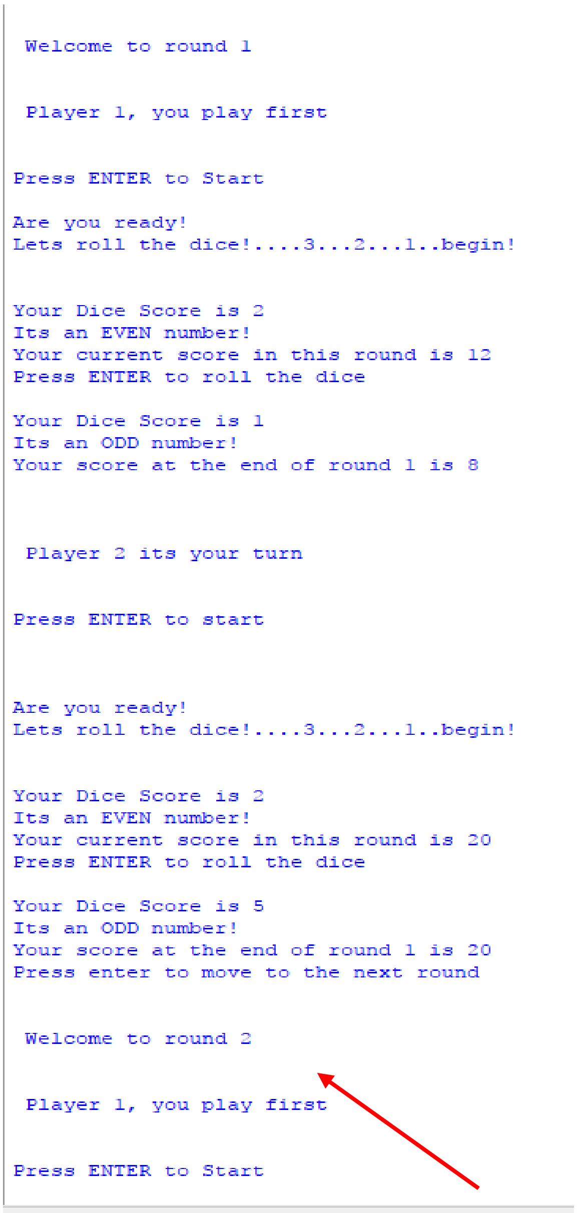
```
def Winners ():     # Declares round as a global variable
  global Round

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()
```

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



```
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()
```

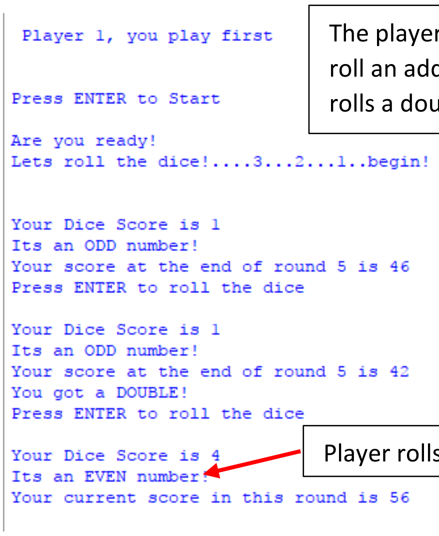
Output:



```
 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
```

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):

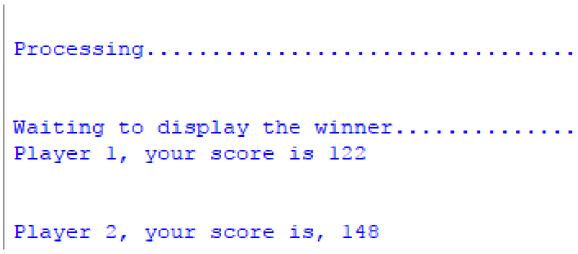


```
 Player 1, you play first


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

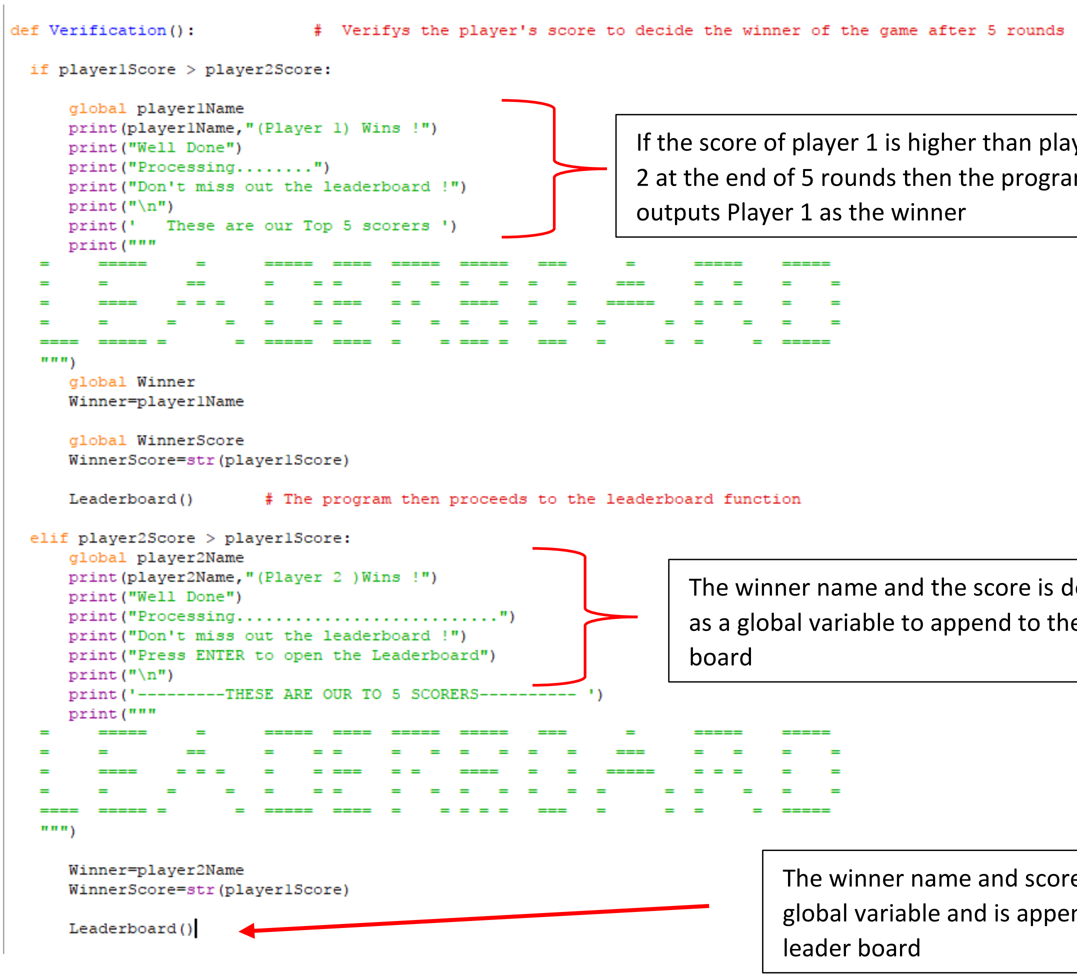
Player rolls an additional dice

Output (When 5 Rounds are finished):



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


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


Player 2, your score is, 148
```

4.4 Displaying the Winner

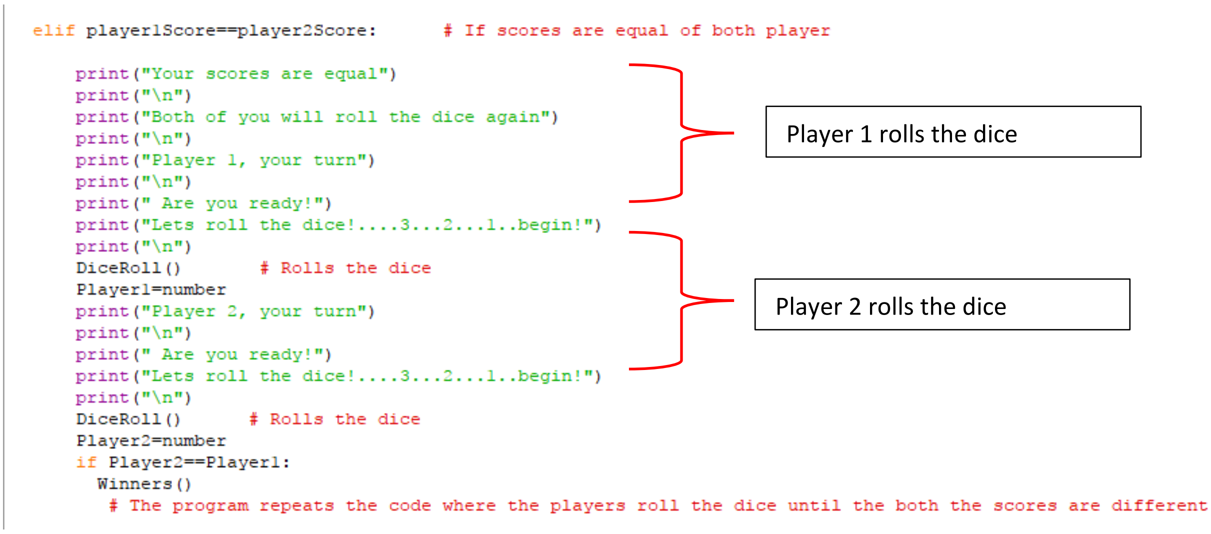


```
def Verification():            #  Verifys the player's score to decide the winner of the game after 5 rounds

  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

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

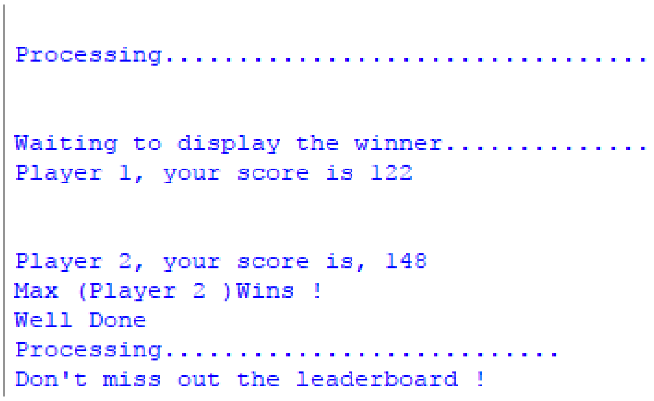


```
  elif player1Score==player2Score:       # If scores are equal of both player

      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

Output (if the scores of both the players are different):

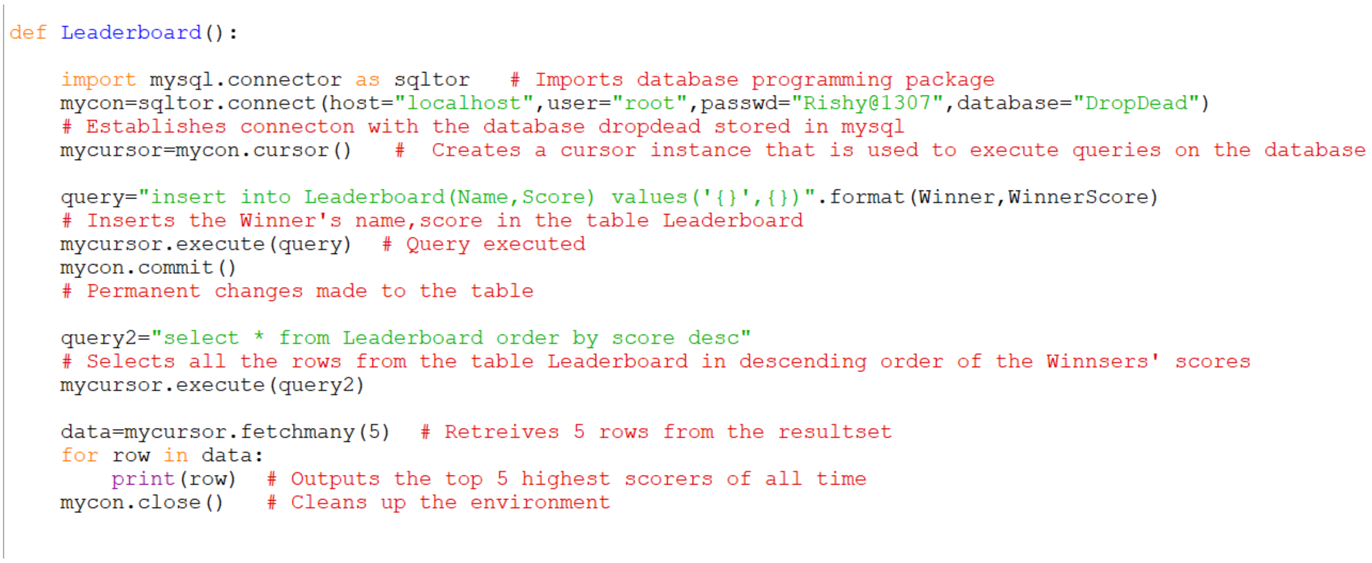


```
Processing...................................


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 !
```

Repeats the Verification function to decide the winner.

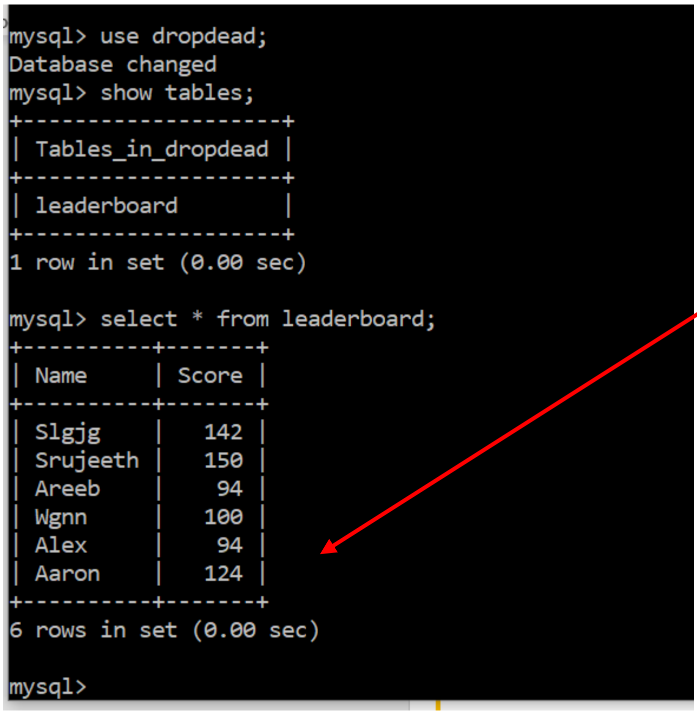
If one of the players as higher score than the other, then the program proceeds to the leader board function whereas it loops the program until one of the player has different dice score than the other.

4.5 Displaying the Leader board through Python interfacing MYSQL.



```
def Leaderboard():

    import mysql.connector as sqltor   # Imports database programming package
    mycon=sqltor.connect(host="localhost",user="root",passwd="Rishy@1307",database="DropDead")
    # Establishes connecton with the database dropdead stored in mysql
    mycursor=mycon.cursor()    #  Creates a cursor instance that is used to execute queries on the database

    query="insert into Leaderboard(Name,Score) values('{}',{})".format(Winner,WinnerScore)
    # Inserts the Winner's name,score in the table Leaderboard
    mycursor.execute(query)  # Query executed
    mycon.commit()
    # Permanent changes made to the table

    query2="select * from Leaderboard order by score desc"
    # Selects all the rows from the table Leaderboard in descending order of the Winnsers' scores
    mycursor.execute(query2)

    data=mycursor.fetchmany(5)  # Retreives 5 rows from the resultset
    for row in data:
        print(row)  # Outputs the top 5 highest scorers of all time
    mycon.close()   # Cleans up the environment
```

Table Leader board in the database drop dead:

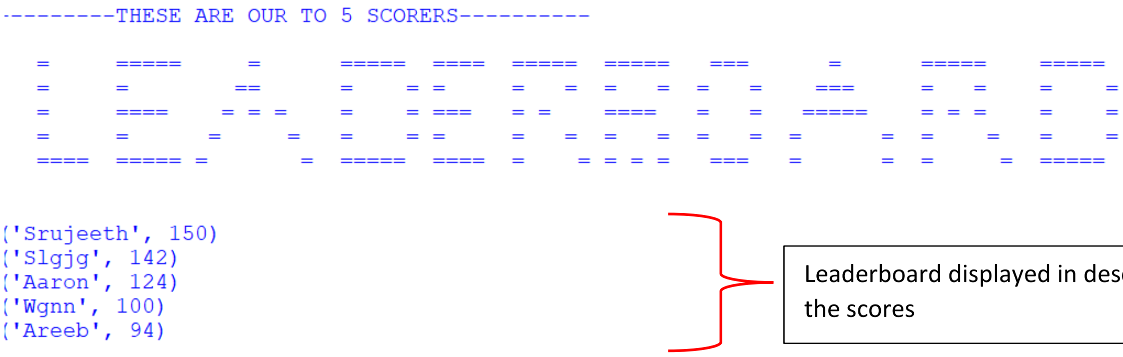


```
mysql> use dropdead;
Database changed
mysql> show tables;
+--------------------+
| Tables_in_dropdead |
+--------------------+
| leaderboard        |
+--------------------+
1 row in set (0.00 sec)

mysql> select * from leaderboard;
+----------+-------+
| Name     | Score |
+----------+-------+
| Slgjg    |   142 |
| Srujeeth |   150 |
| Areeb    |    94 |
| Wgnn     |   100 |
| Alex     |    94 |
| Aaron    |   124 |
+----------+-------+
6 rows in set (0.00 sec)

mysql>
```

Inserts the record of the new winner's score and name in the Leaderboard table.

The table is unordered with respect to the scores

Output (Leader-board):



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

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


('Srujeeth', 150)
('Slgjg', 142)
('Aaron', 124)
('Wgnn', 100)
('Areeb', 94)
```

Leaderboard displayed in descending order of the scores

5. 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.

6. FUTURE SCOPE

3D graphics have become an important part of every aspect of design nowadays. From game development, to web development, to animations, to data representation, it can be found everywhere. Because of this, it would be great to have an graphics engine in this game. The extension of this project would be integrating graphics and animation to the dice drop game. This could be effectively done through various python libraries, one of them being the Arcade library. The Arcade library is a modern Python Module used widely for developing 2D video games with compelling graphics and sound. Arcade is an object-oriented library. It can be installed like any other Python Package.

7. BIBILOGRAPH

- ✓ The block diagram has been made in Draw.io
  https://www.draw.io/

- ✓ Reference taken from Sumita Arora Computer Science with Python Class XII book