ROBERT GORDON UNIVERSITY ABERDEEN TEF Gold

INFORMATICS INSTITUTE OF TECHNOLOGY

BSc. Artificial Intelligence & Data Science
Level 04

CM 1601
PROGRAMMING FUNDAMENTALS

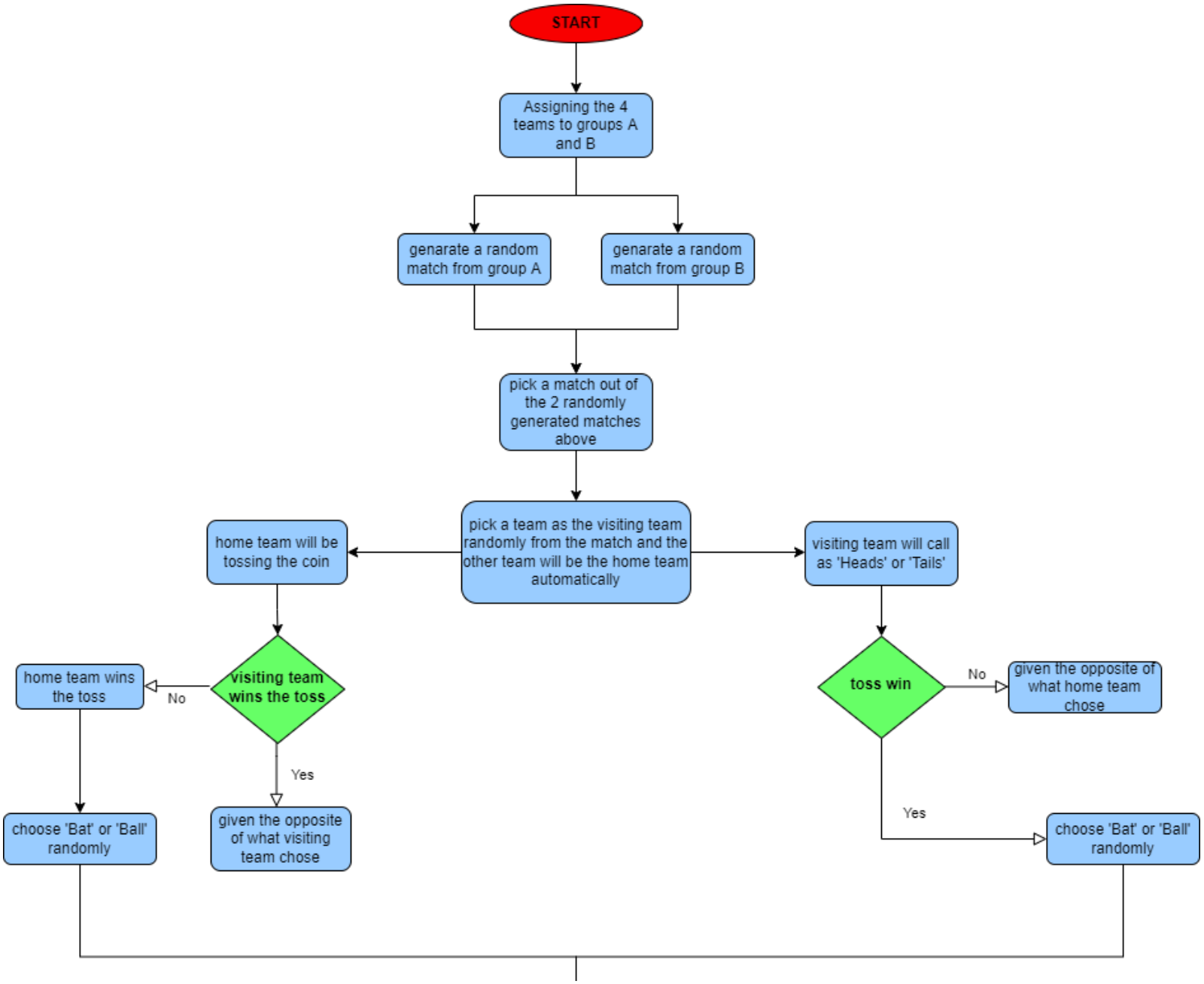
T20 Cricket Tournament

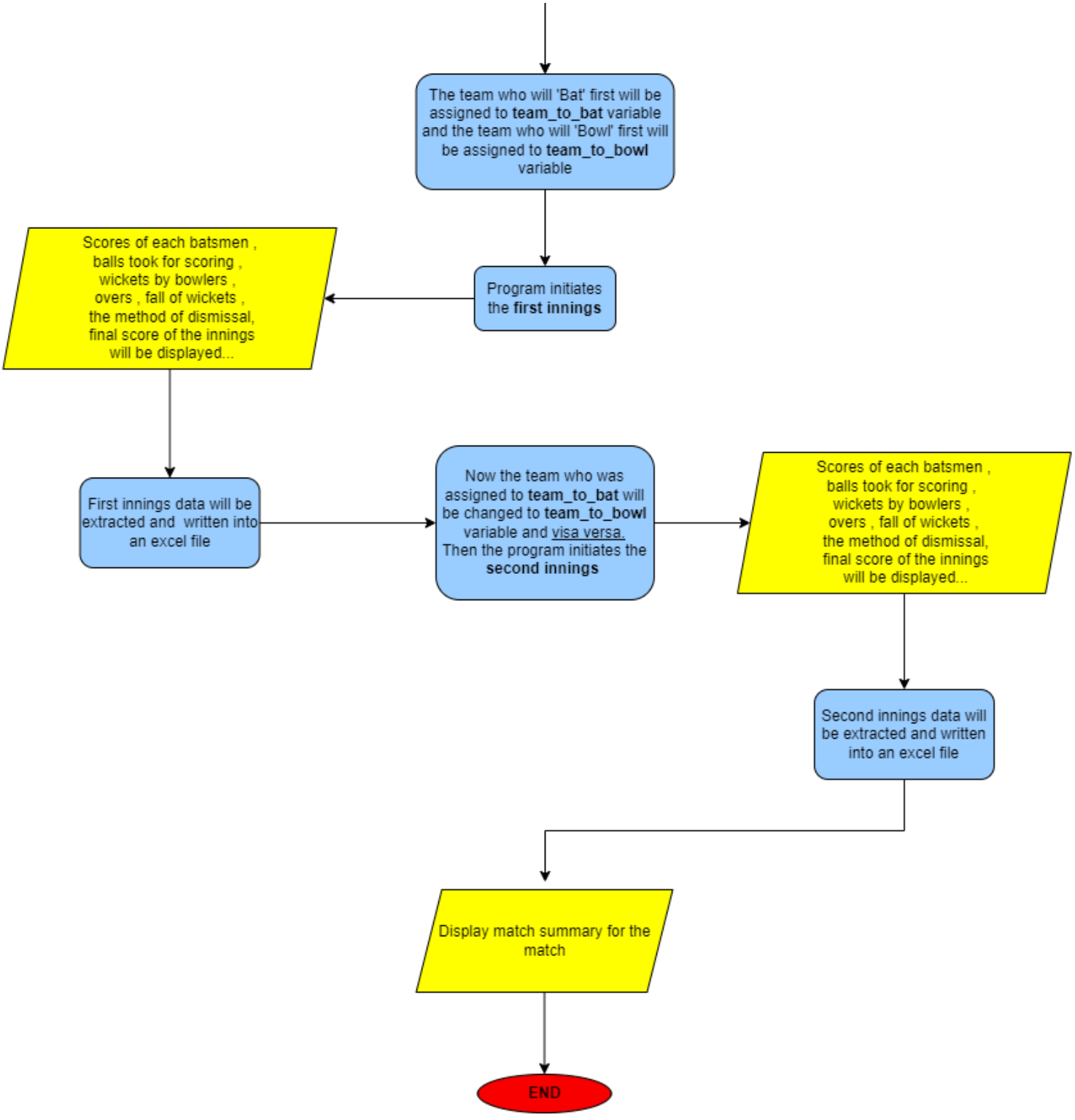
COURSEWORK-I REPORT

NADUN SHAMIKA SENARATHNE
IIT ID: 20210488
RGU ID: 2117538

Table of Contents

Generate random match Flowchart




The team who will 'Bat' first will be assigned to team_to_bat variable and the team who will 'Bowl' first will be assigned to team_to_bowl variable
Program initiates the first innings
Scores of each batsmen , balls took for scoring , wickets by bowlers , overs , fall of wickets , the method of dismissal, final score of the innings will be displayed...
First innings data will be extracted and written into an excel file
Now the team who was assigned to team_to_bat will be changed to team_to_bowl variable and visa versa. Then the program initiates the second innings
Scores of each batsmen , balls took for scoring , wickets by bowlers , overs , fall of wickets , the method of dismissal, final score of the innings will be displayed...
Second innings data will be extracted and written into an excel file
Display match summary for the match
END

Source Code

Assumptions

- User can only edit player names.
- There are no past records of players prior to this tournament.
- Only ways of getting out is by ‘Bowled’,’LBW’ or ‘Caught’ by the bowler himself.
- The hand cricket strategy is used for scoring and dismissals. (When the **batter_score** is equal to the **bolwer_score** the batsman is declared out otherwise the batsman will get runs)
- Extras are not considered as a method of scoring in this tournament.

Cricket.py

```
# ------------------------------------------------------------------------------
----------------------Importing Modules-----------------------------------------
------------------------------------------------------
import openpyxl
from openpyxl import load_workbook
from openpyxl.drawing.image import Image
import random
import pandas as pd
from operator import itemgetter
import os
import matplotlib.pyplot as plt
from PIL import Image
# ------------------------------------------------------------------------------
-----------Store information about teams and players.---------------------------
------------------------------------------------------
Mumbai_India = ['Mumbai_India',
                r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\team_data\\Mumbai_India\\Mumbai_India.xlsx']
Chennai_SouthAfrica = ['Chennai_SouthAfrica',
                       r'E:\\IIT\\1st Year\\1st Trimester\\CM1601
[PRO]  Programming Fundamentals\\Course
Work\\team_data\\Chennai_SouthAfrica\\Chennai_SouthAfrica.xlsx']
Delhi_NewZealand = ['Delhi_NewZealand',
                    r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\team_data\\Delhi_NewZealand\\Delhi_NewZealand.xlsx']
RoyalChallengers_Bangladesh = ['RoyalChallengers_Bangladesh',
                               r'E:\\IIT\\1st Year\\1st Trimester\\CM1601
[PRO]  Programming Fundamentals\\Course
Work\\team_data\\RoyalChallengers_Bangladesh\\RoyalChallengers_Bangladesh.xlsx']
Rajastan_Australia = ['Rajastan_Australia',
                      r'E:\\IIT\\1st Year\\1st Trimester\\CM1601
[PRO]  Programming Fundamentals\\Course
Work\\team_data\\Rajastan_Australia\\Rajastan_Australia.xlsx']
Kolkata_England = ['Kolkata_England',
                   r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\team_data\\Kolkata_England\\Kolkata_England.xlsx']
Punjab_Pakistan = ['Punjab_Pakistan',
                   r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\team_data\\Punjab_Pakistan\\Punjab_Pakistan.xlsx']
Sunrisers_SriLanka = ['Sunrisers_SriLanka',
```

```
                    r'E:\\IIT\\1st Year\\1st Trimester\\CM1601
[PRO]  Programming Fundamentals\\Course
Work\\team_data\\Sunrisers_SriLanka\\Sunrisers_SriLanka.xlsx']

# ------------------------------------------------------------------------------
----------------- Assigning teams to groups-------------------------------------
------------------------------------------------------
Group_A = [Mumbai_India, Chennai_SouthAfrica,
           Delhi_NewZealand, RoyalChallengers_Bangladesh]
Group_B = [Rajastan_Australia, Kolkata_England,
           Punjab_Pakistan, Sunrisers_SriLanka]
# ------------------------------------------------------------------------------
--------Global variables which are used in the functions-----------------------
------------------------------------------------------
user_input = ''
global_exit = ''
TOTAL_WICKETS = 10
TOTAL_BALLS = 120
first_ing_total = 0
first_ing_wickets = 0
second_ing_total = 0
second_ing_wickets = 0
match_between = []
team_to_bat = []
team_to_bowl = []
visiting_team = []
home_team = []
filename_match = ''
selection = ''
toss = ''
choose = ''
df_score_card_first_ing_without_index = []
df_bowler_list_first_ing_without_index = []
df_score_card_second_ing_without_index = []
df_bowler_list_second_ing_without_index = []
winning_team = []
losing_team = []
graph_first_ing_balls = []
graph_first_ing_total = []
graph_first_ing_fow_balls = []
graph_first_ing_fow_total = []
graph_second_ing_balls = []
graph_second_ing_total = []
graph_second_ing_fow_balls = []
graph_second_ing_fow_total = []
```

```
# ------------------------------------------------------------------------------
-------------------Defining the functions-----------------------------------
-------------------------------------------------------


def GetTeam(teamName):
    print(GetTeam)
    df = pd.read_excel(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\team_data\\'+teamName+'\\'+teamName+'.xlsx')
    editable_options = df.iloc[:, :1]
    print(editable_options)


def EditTeam(teamName, row, col, edited_name):
    df = pd.read_excel(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\team_data\\'+teamName+'\\'+teamName+'.xlsx')
    df.at[row, col] = edited_name
    print(df.iloc[:, 0])
    df.to_excel(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\team_data\\'+teamName+'\\'+teamName+'.xlsx',
index=False)


def editPlayer(teamName):
    GetTeam(teamName)
    editTeam = input(
        f"Do you want to make any changes on team {teamName}? \n1 - yes \n0 - no
")
    if editTeam == '0':
        global global_exit
        global_exit = 'y'
    while editTeam == '1':
        row = int(input(
            "Which player do you want to edit? \nSelect the corresponding row
number :  "))
        col = 'PLAYER NAME'
        val = input("What should be the change then? ")

        # ------------------------------------------------------------Update
Player name in player_standings when user edit name----------------------------
-------------------------------------
```

```
        team = r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\team_data\\'+teamName+'\\'+teamName+'.xlsx'

        player_standings = pd.read_excel(
            r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\player_standings.xlsx')
        df_player_standings = pd.DataFrame(player_standings)

        wb_obj = openpyxl.load_workbook(team)
        sheet_obj = wb_obj.active

        player_name = sheet_obj.cell(row=row+2, column=1).value

        find_player_standing_index =
df_player_standings.index[df_player_standings['PLAYER NAME'] ==
player_name].tolist(
        )
        df_player_standings.at[find_player_standing_index, 'PLAYER NAME'] = val

        writer = pd.ExcelWriter(
            r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\player_standings.xlsx',
engine='xlsxwriter')
        workbook = writer.book
        worksheet = workbook.add_worksheet('Sheet1')
        writer.sheets['Sheet1'] = worksheet
        df_player_standings.to_excel(
            writer, sheet_name='Sheet1', startrow=0, startcol=0, index=False)
        writer.save()
        writer.close()

        EditTeam(teamName, row, col, val)
        print("your changes have been saved successfully !!!")
        editTeam = input(
            f"Do you want to make any more changes on team {teamName} again?\n1 -
yes \n0 - no ")
        if editTeam == '0':
            global_exit = 'y'
        else:
            global_exit = ''


def team_profile_edit(getData):
    global global_exit
    while (getData == '2') and (global_exit != 'y'):
        getGroup = input(
```

```
            "Which group do you want to see? \n1 - group A \n2 - group B \nOr
press 'x' to exit...     ")
        if getGroup == 'x':
            break

        elif getGroup == '1':
            getTeam = input("Which team do you want to see? \n1 - Mumbai India
\n2 - Chennai SouthAfrica \n3 - Delhi NewZealand \n4 - RoyalChallengers
Bangladesh \n Select a number from 1 to 4 \nOr press 'x' to exit...       ")
            if getTeam == 'x':
                break
            if getTeam == '1':
                editPlayer('Mumbai_India')

            elif getTeam == '2':
                editPlayer('Chennai_SouthAfrica')

            elif getTeam == '3':
                editPlayer('Delhi_NewZealand')

            elif getTeam == '4':
                editPlayer('RoyalChallengers_Bangladesh')

        elif getGroup == '2':
            getTeam = input(
                "Which team do you want to see? \n1 - Rajastan Australia \n2 -
Kolkata England \n3 - Punjab Pakistan \n4 - Sunrisers SriLanka \n Select a number
from 1 to 4")

            if getTeam == '1':
                editPlayer('Rajastan_Australia')

            elif getTeam == '2':
                editPlayer('Kolkata_England')

            elif getTeam == '3':
                editPlayer('Punjab_Pakistan')

            elif getTeam == '4':
                editPlayer('Sunrisers_SriLanka')
    else:
        global_exit = ''


def generate_random_match():
```

```
    match_list = os.listdir(
        r'E:\\IIT\\1st Year\\1st Trimester\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\matches')
    match_list_count = len(match_list)
    match_list_count = match_list_count if match_list_count > 0 else 1

    while match_list_count != 12:
        match_between_A = random.sample(Group_A, 2)
        match_between_B = random.sample(Group_B, 2)

        chosen_match = [match_between_A, match_between_B]
        global match_between
        temp_match_between = random.choice(chosen_match)

        temp1 = str(temp_match_between[0][0]) + '_vs_' + \
            str(temp_match_between[1][0]+'.xlsx')
        temp2 = str(temp_match_between[1][0]) + '_vs_' + \
            str(temp_match_between[0][0]+'.xlsx')

        if (temp1 not in match_list) and (temp2 not in match_list):
            match_between = temp_match_between
            break

    else:
        match_between = []
        raise IndexError('A very specific bad thing happened.')


def points_table():
    points_table = pd.read_excel(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\points_table.xlsx')
    df_points_table = pd.DataFrame(points_table)

    for team in match_between:

        if(team in Group_A):
            group = "Group A"
        else:
            group = "Group B"

        find_team_index = df_points_table.index[df_points_table[group] ==
team[0]].tolist(
        )
```

```
        if(group == "Group A"):
            current_match_count = df_points_table.at[find_team_index[0],
'Matches_A']
            df_points_table.at[find_team_index,
                               'Matches_A'] = current_match_count+1
        else:
            current_match_count = df_points_table.at[find_team_index[0],
'Matches_B']
            df_points_table.at[find_team_index,
                               'Matches_B'] = current_match_count+1
        print('\n\n')

    # ------------------------------------------------------------------------
Write data to excel file by creating Excel Writer Object from Pandas-------------
-----------------------------------------------------

    writer = pd.ExcelWriter(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\points_table.xlsx', engine='xlsxwriter')
    workbook = writer.book
    worksheet = workbook.add_worksheet('Match Summary')
    writer.sheets['Match Summary'] = worksheet
    df_points_table.to_excel(writer, sheet_name='Match Summary',
                             startrow=0, startcol=0, index=False)
    writer.save()
    writer.close()


def toss_coin():
    coin = ["heads", "tails"]
    options = ['bat', 'bowl']
    global team_to_bat
    global team_to_bowl
    global visiting_team
    global home_team
    global selection
    global toss
    global choose

    visiting_team = random.choice(match_between)

    if visiting_team in match_between:
        match_between.remove(visiting_team)

    home_team = match_between[0]
```

```
    # This simulates the coin being tossed
    toss = random.choice(coin)
    # This simulates the visiting team choose head or tails
    selection = random.choice(coin)
    # This simulates the visiting team choose bat or bowl
    choose = random.choice(options)

    print('\n\n')
    print('Home Team - ', home_team[0])
    print('Visiting Team - ', visiting_team[0])
    print('\n\n')

    if selection == toss:
        print(visiting_team[0], 'won the toss and chose to', choose)
        if choose == options[0]:
            team_to_bat = visiting_team
            team_to_bowl = home_team
        else:
            team_to_bat = home_team
            team_to_bowl = visiting_team

    else:
        print(home_team[0], 'won the toss and chose to', choose)
        if choose == options[0]:
            team_to_bat = home_team
            team_to_bowl = visiting_team
        else:
            team_to_bat = visiting_team
            team_to_bowl = home_team

    print('\n\n')
    print('\n\nteam_to_bat', team_to_bat[0])
    print('team_to_bowl', team_to_bowl[0])
    print('\n\n')


def player_standings(batting, bowling):
    player_standings = pd.read_excel(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\player_standings.xlsx')
    df_player_standings = pd.DataFrame(player_standings)

    for player in batting:
```

```
        find_player_index = df_player_standings.index[df_player_standings['PLAYER
NAME'] == player[0]].tolist(
        )
        if not(len(find_player_index) > 0):
            raise Exception('\nPLAYER NOT FOUND!!!!!!!!!')
        current_player_runs = df_player_standings.at[find_player_index[0], 'TOTAL
RUNS']
        df_player_standings.at[find_player_index,
                               'TOTAL RUNS'] = current_player_runs+player[1]

    writer = pd.ExcelWriter(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\player_standings.xlsx',
engine='xlsxwriter')
    workbook = writer.book
    worksheet = workbook.add_worksheet('Sheet1')
    writer.sheets['Sheet1'] = worksheet
    df_player_standings.to_excel(
        writer, sheet_name='Sheet1', startrow=0, startcol=0, index=False)

    writer.save()
    writer.close()

    for player in bowling:
        find_player_index = df_player_standings.index[df_player_standings['PLAYER
NAME'] == player[0]].tolist(
        )
        current_player_wickets = df_player_standings.at[find_player_index[0],
'WICKETS']
        df_player_standings.at[find_player_index,
                               'WICKETS'] = current_player_wickets+player[3]

    writer = pd.ExcelWriter(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\player_standings.xlsx',
engine='xlsxwriter')
    workbook = writer.book
    worksheet = workbook.add_worksheet('Sheet1')
    writer.sheets['Sheet1'] = worksheet
    df_player_standings.to_excel(
        writer, sheet_name='Sheet1', startrow=0, startcol=0, index=False)

    writer.save()
    writer.close()
```

```
def display_player_standings():
    player_standings = pd.read_excel(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\player_standings.xlsx')
    df_player_standings = pd.DataFrame(player_standings)
    print('\n\nTop 5 run scores of the tournament')
    print(df_player_standings[['PLAYER NAME', 'TOTAL RUNS']].nlargest(
        5, 'TOTAL RUNS').to_string(index=False))
    print('\n\nTop 5 wicket takers of the tournament')
    print(df_player_standings[['PLAYER NAME', 'WICKETS']].nlargest(
        5, 'WICKETS').to_string(index=False))


def first_innings():
    global filename_match
    global first_ing_total
    global first_ing_wickets
    global df_score_card_first_ing_without_index
    global df_bowler_list_first_ing_without_index
    global graph_first_ing_balls
    global graph_first_ing_total
    global graph_first_ing_fow_balls
    global graph_first_ing_fow_total
    graph_first_ing_balls = []
    graph_first_ing_total = []
    graph_first_ing_fow_balls = []
    graph_first_ing_fow_total = []

    first_ing_total = 0
    first_ing_wickets = 0
    first_ing_balls = 1
    score_card_first_ing = []

    batsman_onstrike = [['name', 0, 0], True]
    batsman_offstrike = [['name', 0, 0], False]

    bowler_onstrike = []

    # batsman_name , runs , balls_faced , method of dismissal , bowler
    # importing batting team
    batting_url = team_to_bat[1]
    batting_team = pd.read_excel(batting_url)
    # converting excel to python list
    yet_to_bat = batting_team.values.tolist()
```

```
    # bowlers_name , first_ing_balls , runs , wickets , economy
    # importing bowling team
    bowling_url = team_to_bowl[1]
    bowling_team = pd.read_excel(bowling_url)
    # converting excel to python list
    bowling_team_list = bowling_team.values.tolist()

    yet_to_bowl = []

    for i in reversed(range(len(bowling_team_list))):
        if len(yet_to_bowl) < 5:
            yet_to_bowl.append([bowling_team_list[i][0], 0, 0, 0, 0])

    dismissed_batsmen = []
    batsman_list = []

    # method of dismissal
    method_of_dismissal = ['Bowled', 'Caught', 'LBW']

    bowler_score = 0  # score counting variable for bowler
    batter_score = 0  # score counting variable for batsman

    # opening batsmen coming to the field
    batsman_onstrike[0] = yet_to_bat.pop(0)
    batsman_offstrike[0] = yet_to_bat.pop(0)

    # opening bowler
    bowler_onstrike = yet_to_bowl.pop(0)

    while (first_ing_balls < (TOTAL_BALLS+1)):
        if first_ing_wickets == TOTAL_WICKETS:
            break
        else:
            if ((first_ing_balls-1) > 0 and (first_ing_balls-1) % 6 == 0) and
(len(yet_to_bowl)) > 0:
                yet_to_bowl.append(bowler_onstrike)
                bowler_onstrike = yet_to_bowl.pop(0)

            # get random scores for bowler and batsman
            bowler_score = random.randint(1, 6)
            batter_score = random.randint(0, 6)

            if bowler_score == batter_score:
                # adding wickets to bowler
                current_bowler_onstrike_first_ing_wickets = 0
```

```
                current_bowler_onstrike_first_ing_wickets = bowler_onstrike[3]
                bowler_onstrike[3] = current_bowler_onstrike_first_ing_wickets +
1
                # adding first_ing_balls to batsman
                current_batsman_onstrike_balls = batsman_onstrike[0][2]
                batsman_onstrike[0][2] = current_batsman_onstrike_balls + 1

                # moving the dismissed_batsmen to dismissed_batsmen array
                dismissed_batsmen.append(batsman_onstrike[0])

                # adding method of dismissal to batsman
                current_batsman_method_of_dismissal = random.choices(
                    method_of_dismissal)
                batsman_onstrike[0][3] = current_batsman_method_of_dismissal[0]

                # adding dismissed bowler name to batsman
                current_batsman_bowler_dismissed = batsman_onstrike[0][4]
                batsman_onstrike[0][4] = current_batsman_bowler_dismissed + \
                    str(bowler_onstrike[0])

                # fall of wickets
                print('FOW at', first_ing_total, ' --> ', first_ing_wickets+1,
                      ' on over -', int(first_ing_balls/6), '.',
(first_ing_balls) % 6, batsman_onstrike[0][0])

                # appending the FOW data to the graph
                graph_first_ing_fow_balls.append(first_ing_balls)
                graph_first_ing_fow_total.append(first_ing_total)

                # bring new batsman to the crease (batsman_onstrike)
                if len(yet_to_bat) > 0:
                    batsman_onstrike[0] = []
                    batsman_onstrike[0] = yet_to_bat.pop(0)

                # out - add wicket to wickets
                first_ing_wickets += 1

            else:
                # adding batter_score to current_batsman
                current_batsman_onstrike_score = 0
                current_batsman_onstrike_score = batsman_onstrike[0][1]
                batsman_onstrike[0][1] = current_batsman_onstrike_score +
batter_score

                # adding first_ing_balls to current_batsman
```

```
                current_batsman_onstrike_balls = batsman_onstrike[0][2]
                batsman_onstrike[0][2] = current_batsman_onstrike_balls + 1

                # adding batter_score to current_bowler
                current_bowler_onstrike_runs = 0
                current_bowler_onstrike_runs = bowler_onstrike[2]
                bowler_onstrike[2] = current_bowler_onstrike_runs + \
                    batter_score

                # swapping onstrike batsman when strike rotates
                if batter_score == 1 or batter_score == 3:
                    current_batsman = batsman_onstrike[0]
                    # swapping onstrike batsman
                    batsman_onstrike[0] = batsman_offstrike[0]
                    batsman_offstrike[0] = current_batsman
                else:
                    pass  # when batter_score is not swapping

                # add batter score to first_ing_total
                first_ing_total += batter_score

        # adding first_ing_balls to bowler
        current_bowler_onstrike_balls = 0
        current_bowler_onstrike_balls = bowler_onstrike[1]
        bowler_onstrike[1] = current_bowler_onstrike_balls + 1

        # adding first_ing_balls to first_ing ball count
        first_ing_balls += 1

        # adding first_ing_total to graph_first_ing_total
        graph_first_ing_total.append(first_ing_total)

    # assinging first innings balls to graph
    graph_first_ing_balls = range(1, first_ing_balls)

    # last dismissed batsman
    last_dismissal = dismissed_batsmen[-1]

    # add dismissed_batsmen to batsman_list
    batsman_list = dismissed_batsmen

    # add each batsman in yet_to_bat to batsman_list array for displaying
purposes
    if len(yet_to_bat) > 0:
        for i in range(len(yet_to_bat)):
```

```
            batsman_list.append(yet_to_bat[i])

    # add on and off strike batsmen to batsman_list
    if first_ing_wickets != TOTAL_WICKETS:
        batsman_onstrike[0][3] = '* NOT OUT'
        batsman_list.append(batsman_onstrike[0])

    batsman_offstrike[0][3] = 'NOT OUT'
    batsman_list.append(batsman_offstrike[0])

    # add batsman_list to score_card_first_ing
    score_card_first_ing = batsman_list

    # add bowlers to bowler_list_first_ing
    bowler_list_first_ing = yet_to_bowl
    bowler_list_first_ing.append(bowler_onstrike)

    # sort score_card_first_ing to the original batting order
    sorted_list = sorted(score_card_first_ing, key=itemgetter(5))

    # convert score_card_first_ing to a data frame for displaying
    df_score_card_first_ing = pd.DataFrame(sorted_list)

    # converting bowler first_ing_balls to overs
    for bowler_overs_first_ing in bowler_list_first_ing:
        bowler_overs_first_ing[1] = str(
            int((bowler_overs_first_ing[1])/6)) + '.' +
str((bowler_overs_first_ing[1]) % 6)

    # additing the economy for bowler
    for bowler_economy_first_ing in bowler_list_first_ing:
        bowler_economy_first_ing[4] = round(
            bowler_economy_first_ing[2]/float(bowler_economy_first_ing[1]), 2)

    # convert df_bowler_list_first_ing to a data frame for displaying
    df_bowler_list_first_ing = pd.DataFrame(bowler_list_first_ing)

    print('\n\n-------------------------------------------------1st Innings
Summary-----------------------------------------')
    print('\n')
    print('\nTotal-', first_ing_total, '\nwickets -', first_ing_wickets,
          '\novers -', int((first_ing_balls-1)/6), '.', (first_ing_balls-1) % 6,
'\nballs', (first_ing_balls-1))
    # print('Extras - ',extras_first_ing)
    print('\nLast dismissal', last_dismissal)
```

```
    print('\n\n----------------------------------------------1st Innings
Scorecard-----------------------------------------')
    new_headers = ['Batting', 'Runs', 'Balls Faced',
                   'MOD', 'Bowler', 'Batting No']
    df_score_card_first_ing.columns = new_headers
    df_score_card_first_ing_without_index = df_score_card_first_ing.set_index(
        'Batting')
    print('\n')
    print(df_score_card_first_ing_without_index)

    # convert match summary to a dataframe
    overs = str(int((first_ing_balls-1)/6)) + \
        '.' + str((first_ing_balls-1) % 6)
    first_ing_summary = [
        [first_ing_total, first_ing_wickets, overs, (first_ing_balls-1)]]

    df_first_ing_summary = pd.DataFrame(first_ing_summary, columns=[
                                        'Total', 'Wickets', 'Overs', 'Balls'])

    print('\n\n----------------------------------------------1st Innings Bowling
figures-----------------------------------------')
    new_headers = ['Bowling', 'Overs', 'Runs', 'Wickets', 'Economy']
    df_bowler_list_first_ing.columns = new_headers
    df_bowler_list_first_ing_without_index = df_bowler_list_first_ing.set_index(
        'Bowling')
    print('\n')
    print(df_bowler_list_first_ing_without_index)
    print('\n\n')

    # ---------------------------------------------------------------------Write
data to excel file by creating Excel Writer Object from Pandas-------------------
-----------------------------------------------------

    filename_match = str(visiting_team[0]) + '_vs_' + str(home_team[0])
    match_file_path = r'E:\\IIT\\1st Year\\1st Trimester\\CM1601
[PRO]  Programming Fundamentals\\Course Work\\tournament\\matches\\' +
filename_match + '.xlsx'

    writer = pd.ExcelWriter(match_file_path, engine='xlsxwriter')
    workbook = writer.book
    worksheet = workbook.add_worksheet('Match Summary')
    writer.sheets['Match Summary'] = worksheet

    df_score_card_first_ing.to_excel(
```

```
        writer, sheet_name='Match Summary', startrow=0, startcol=0, index=False)

    df_first_ing_summary.to_excel(
        writer, sheet_name='Match Summary', startrow=14, startcol=0, index=False)

    df_bowler_list_first_ing.to_excel(
        writer, sheet_name='Match Summary', startrow=19, startcol=0, index=False)

    writer.save()
    writer.close()
    print('\n\n')
    # ------------------------------------------------------------------------
-----------------Update player standings-----------------------------------
-------------------------------------------------------
    player_standings(score_card_first_ing, bowler_list_first_ing)


def second_innings():
    global second_ing_total
    global second_ing_wickets
    global df_score_card_second_ing_without_index
    global df_bowler_list_second_ing_without_index
    global graph_second_ing_balls
    global graph_second_ing_total
    global graph_second_ing_fow_balls
    global graph_second_ing_fow_total
    graph_second_ing_balls = []
    graph_second_ing_total = []
    graph_second_ing_fow_balls = []
    graph_second_ing_fow_total = []
    second_ing_total = 0
    second_ing_wickets = 0
    second_ing_balls = 1
    score_card_second_ing = []

    batsman_onstrike = [['name', 0, 0], True]
    batsman_offstrike = [['name', 0, 0], False]

    bowler_onstrike = []

    # batsman_name , runs , balls_faced , method of dismissal , bowler
    # importing batting team
    batting_url = team_to_bowl[1]
    batting_team = pd.read_excel(batting_url)
    # converting excel to python list
```

```
    yet_to_bat = batting_team.values.tolist()

    # bowlers_name , second_ing_balls , runs , wickets
    # importing bowling team
    bowling_url = team_to_bat[1]
    bowling_team = pd.read_excel(bowling_url)
    # converting excel to python list
    bowling_team_list = bowling_team.values.tolist()

    yet_to_bowl = []

    for i in reversed(range(len(bowling_team_list))):
        if len(yet_to_bowl) < 5:
            yet_to_bowl.append([bowling_team_list[i][0], 0, 0, 0, 0])

    dismissed_batsmen = []
    batsman_list = []

    # method of dismissal
    method_of_dismissal = ['Bowled', 'Caught', 'LBW']

    bowler_score = 0  # score counting variable for bowler
    batter_score = 0  # score counting variable for batsman

    # opening batsmen coming to the field
    batsman_onstrike[0] = yet_to_bat.pop(0)
    batsman_offstrike[0] = yet_to_bat.pop(0)

    # opening bowler
    bowler_onstrike = yet_to_bowl.pop(0)

    while (second_ing_balls < (TOTAL_BALLS+1)):
        if ((second_ing_wickets == TOTAL_WICKETS) or (second_ing_total >
first_ing_total)):
            break
        else:
            if ((second_ing_balls-1) > 0 and (second_ing_balls-1) % 6 == 0) and
(len(yet_to_bowl)) > 0:
                yet_to_bowl.append(bowler_onstrike)
                bowler_onstrike = yet_to_bowl.pop(0)

            # get random scores for bowler and batsman
            bowler_score = random.randint(1, 6)
            batter_score = random.randint(0, 6)
```

```
            if bowler_score == batter_score:
                # adding wickets to bowler
                current_bowler_onstrike_second_ing_wickets = 0
                current_bowler_onstrike_second_ing_wickets = bowler_onstrike[3]
                bowler_onstrike[3] = current_bowler_onstrike_second_ing_wickets +
1
                # adding second_ing_balls to batsman
                current_batsman_onstrike_balls = batsman_onstrike[0][2]
                batsman_onstrike[0][2] = current_batsman_onstrike_balls + 1

                # moving the dismissed_batsmen to dismissed_batsmen array
                dismissed_batsmen.append(batsman_onstrike[0])

                # adding method of dismissal to batsman
                current_batsman_method_of_dismissal = random.choices(
                    method_of_dismissal)
                batsman_onstrike[0][3] = current_batsman_method_of_dismissal[0]

                # adding dismissed bowler name to batsman
                current_batsman_bowler_dismissed = batsman_onstrike[0][4]
                batsman_onstrike[0][4] = current_batsman_bowler_dismissed + \
                    str(bowler_onstrike[0])

                # fall of wickets
                print('FOW', second_ing_total, ' --> ', second_ing_wickets+1,
                      ' on over -', int(second_ing_balls/6), '.',
(second_ing_balls) % 6, batsman_onstrike[0][0])

                # appendind the FOW data to the graph
                graph_second_ing_fow_balls.append(second_ing_balls)
                graph_second_ing_fow_total.append(second_ing_total)

                # bring new batsman to the crease (batsman_onstrike)
                if len(yet_to_bat) > 0:
                    batsman_onstrike[0] = []
                    batsman_onstrike[0] = yet_to_bat.pop(0)

                # out - add wicket to wickets
                second_ing_wickets += 1

            else:
                # adding batter_score to current_batsman
                current_batsman_onstrike_score = 0
                current_batsman_onstrike_score = batsman_onstrike[0][1]
```

```
                batsman_onstrike[0][1] = current_batsman_onstrike_score +
batter_score

                # adding second_ing_balls to current_batsman
                current_batsman_onstrike_balls = batsman_onstrike[0][2]
                batsman_onstrike[0][2] = current_batsman_onstrike_balls + 1

                # adding batter_score to current_bowler
                current_bowler_onstrike_runs = 0
                current_bowler_onstrike_runs = bowler_onstrike[2]
                bowler_onstrike[2] = current_bowler_onstrike_runs + \
                    batter_score

                # swapping onstrike batsman when strike rotates
                if batter_score == 1 or batter_score == 3:
                    current_batsman = batsman_onstrike[0]
                    # swapping onstrike batsman
                    batsman_onstrike[0] = batsman_offstrike[0]
                    batsman_offstrike[0] = current_batsman
                else:
                    pass  # when batter_score is not swapping

                # add batter score to second_ing_total
                second_ing_total += batter_score

        # adding second_ing_balls to bowler
        current_bowler_onstrike_balls = 0
        current_bowler_onstrike_balls = bowler_onstrike[1]
        bowler_onstrike[1] = current_bowler_onstrike_balls + 1
        second_ing_balls += 1

        # adding second_ing_total to graph_second_ing_total
        graph_second_ing_total.append(second_ing_total)

    # assinging second innings balls to graph
    graph_second_ing_balls = range(1, second_ing_balls)

    # plotting balls and total graph
    plt.plot(graph_first_ing_balls, graph_first_ing_total)
    plt.plot(graph_second_ing_balls, graph_second_ing_total)

    # plotting fow graph
    plt.plot(graph_first_ing_fow_balls, graph_first_ing_fow_total, linestyle='',
linewidth=3,
             marker='o', markerfacecolor='green', markersize=6)
```

```
    plt.plot(graph_second_ing_fow_balls, graph_second_ing_fow_total,
linestyle='', linewidth=3,
             marker='o', markerfacecolor='red', markersize=6)

    # add legend
    plt.legend([team_to_bat[0], team_to_bowl[0],
                team_to_bat[0]+' Wickets', team_to_bowl[0]+' Wickets'])

    # giving a title to the graph
    plt.title('--Innings Progression Graph--')

    # naming the x axis
    plt.xlabel('Balls')
    # naming the y axis
    plt.ylabel('Runs')

    # saving the graph into an image
    plt.savefig(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\myplot.png', format='png')

    # clearing the plt image to avoid overwriting
    plt.clf()

    # last dismissed batsman
    last_dismissal = dismissed_batsmen[-1]

    # add dismissed_batsmen to batsman_list
    batsman_list = dismissed_batsmen

    # add each batsman in yet_to_bat to batsman_list array for displaying
purposes
    if len(yet_to_bat) > 0:
        for i in range(len(yet_to_bat)):
            batsman_list.append(yet_to_bat[i])

    # add on and off strike batsmen to batsman_list
    if second_ing_wickets != TOTAL_WICKETS:
        batsman_onstrike[0][3] = '* NOT OUT'
        batsman_list.append(batsman_onstrike[0])

    batsman_offstrike[0][3] = 'NOT OUT'
    batsman_list.append(batsman_offstrike[0])

    # add batsman_list to score_card_second_ing
```

```
    score_card_second_ing = batsman_list

    # sort score_card_second_ing to the original batting order
    sorted_list = sorted(score_card_second_ing, key=itemgetter(5))

    # add bowlers to bowler_list_second_ing
    bowler_list_second_ing = yet_to_bowl
    bowler_list_second_ing.append(bowler_onstrike)

    # convert df_score_card_second_ing to a data frame for displaying
    df_score_card_second_ing = pd.DataFrame(sorted_list)

    # converting bowler second_ing_balls to overs
    for bowler_overs_second_ing in bowler_list_second_ing:
        bowler_overs_second_ing[1] = str(
            int((bowler_overs_second_ing[1])/6)) + '.' +
str((bowler_overs_second_ing[1]) % 6)

    # adding the economy for bowler
    for bowler_economy_second_ing in bowler_list_second_ing:
        if not(float(bowler_economy_second_ing[1]) > 0):
            raise Exception('BOWLER ECONOMY NOT FOUND!!!!!!!!!!!')
        bowler_economy_second_ing[4] = round(
            bowler_economy_second_ing[2]/float(bowler_economy_second_ing[1]), 2)

    # convert df_bowler_list_second_ing to a data frame for displaying
    df_bowler_list_second_ing = pd.DataFrame(bowler_list_second_ing)

    print('\n\n-----------------------------------------------------2nd Innings
Summary-----------------------------------')
    print('\nTotal-', second_ing_total, '\nwickets -', second_ing_wickets,
          '\novers -', int((second_ing_balls-1)/6), '.', (second_ing_balls-1) %
6, '\nballs', (second_ing_balls-1))
    # print('Extras',extras_second_ing)
    print('\nLast dismissal', last_dismissal)

    print('\n\n-----------------------------------------------------2nd Innings
Scorecard---------------------------------')
    new_headers = ['Batting', 'Runs', 'Balls Faced',
                   'MOD', 'Bowler', 'Batting No']
    df_score_card_second_ing.columns = new_headers
    df_score_card_second_ing_without_index = df_score_card_second_ing.set_index(
        'Batting')
    print('\n')
    print(df_score_card_second_ing_without_index)
```

```
    # convert match summary to a dataframe
    overs = str(int((second_ing_balls-1)/6)) + \
        '.' + str((second_ing_balls-1) % 6)
    second_ing_summary = [
        [second_ing_total, second_ing_wickets, overs, (second_ing_balls-1)]]

    df_second_ing_summary = pd.DataFrame(second_ing_summary, columns=[
        'Total', 'Wickets', 'Overs', 'Balls'])

    print('\n\n-----------------------------------------------------2nd Innings
Bowling figures------------------------------')
    new_headers = ['Bowling', 'Overs', 'Runs', 'Wickets', 'Economy']
    df_bowler_list_second_ing.columns = new_headers
    df_bowler_list_second_ing_without_index =
df_bowler_list_second_ing.set_index(
        'Bowling')
    print('\n')
    print(df_bowler_list_second_ing_without_index)
    print('\n\n')

    # ---------------------------------------------------------------------
Write data to excel file by creating Excel Writer Object from Pandas-------------
-------------------------------------------------------

    book = load_workbook(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\matches\\' + filename_match + '.xlsx')
    writer = pd.ExcelWriter(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\matches\\' + filename_match + '.xlsx',
engine='openpyxl')
    writer.book = book

    writer.sheets = dict((ws.title, ws) for ws in book.worksheets)

    df_score_card_second_ing.to_excel(
        writer, sheet_name='Match Summary', startrow=0, startcol=9, index=False)

    df_second_ing_summary.to_excel(
        writer, sheet_name='Match Summary', startrow=14, startcol=9, index=False)

    df_bowler_list_second_ing.to_excel(
        writer, sheet_name='Match Summary', startrow=19, startcol=9, index=False)
```

```
    worksheet = book.worksheets[0]
    img = openpyxl.drawing.image.Image(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\myplot.png')
    img.anchor = 'D30'
    worksheet.add_image(img)

    writer.save()
    writer.close()
    # ------------------------------------------------------------------------
-----------------Update player standings-----------------------------------------
--------------------------------------------------
    player_standings(score_card_second_ing, bowler_list_second_ing)


def update_points_table(winning_team, losing_team, is_drawn):
    update_points_table = pd.read_excel(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\points_table.xlsx')
    df_update_points_table = pd.DataFrame(update_points_table)

    if is_drawn == 0:
        if(winning_team in Group_A):
            group = "Group A"
            find_points_table_index =
df_update_points_table.index[df_update_points_table[group] ==
winning_team[0]].tolist(
            )
            current_wins_count =
df_update_points_table.at[find_points_table_index[0], 'Won']
            df_update_points_table.at[find_points_table_index,
                                      'Won'] = current_wins_count+1

            find_points_table_index =
df_update_points_table.index[df_update_points_table[group] ==
losing_team[0]].tolist(
            )
            current_loss_count =
df_update_points_table.at[find_points_table_index[0], 'Lost']
            df_update_points_table.at[find_points_table_index,
                                      'Lost'] = current_loss_count+1

            find_points_table_index =
df_update_points_table.index[df_update_points_table[group] ==
winning_team[0]].tolist(
```

```
            )
            current_points_count =
df_update_points_table.at[find_points_table_index[0], 'Points']
            df_update_points_table.at[find_points_table_index,
                                      'Points'] = current_points_count+2

        elif(winning_team in Group_B):
            group = "Group B"

            find_points_table_index =
df_update_points_table.index[df_update_points_table[group] ==
winning_team[0]].tolist(
            )
            current_wins_count =
df_update_points_table.at[find_points_table_index[0], 'Won_B']
            df_update_points_table.at[find_points_table_index,
                                      'Won_B'] = current_wins_count+1

            find_points_table_index =
df_update_points_table.index[df_update_points_table[group] ==
losing_team[0]].tolist(
            )
            current_loss_count =
df_update_points_table.at[find_points_table_index[0], 'Lost_B']
            df_update_points_table.at[find_points_table_index,
                                      'Lost_B'] = current_loss_count+1

            find_points_table_index =
df_update_points_table.index[df_update_points_table[group] ==
winning_team[0]].tolist(
            )
            current_points_count =
df_update_points_table.at[find_points_table_index[0], 'Points_B']
            df_update_points_table.at[find_points_table_index,
                                      'Points_B'] = current_points_count+2

    elif is_drawn == 1:

        if(winning_team in Group_A):
            group = "Group A"

            find_points_table_index_1 =
df_update_points_table.index[df_update_points_table[group] ==
winning_team[0]].tolist(
            )
```

```
            find_points_table_index_2 =
df_update_points_table.index[df_update_points_table[group] ==
losing_team[0]].tolist(
            )
            current_points_count_1 =
df_update_points_table.at[find_points_table_index_1[0], 'Points']
            df_update_points_table.at[find_points_table_index_1,
                                      'Points'] = current_points_count_1+1

            current_points_count_2 =
df_update_points_table.at[find_points_table_index_2[0], 'Points']
            df_update_points_table.at[find_points_table_index_2,
                                      'Points'] = current_points_count_2+1

        elif(winning_team in Group_B):
            group = "Group B"

            find_points_table_index_1 =
df_update_points_table.index[df_update_points_table[group] ==
winning_team[0]].tolist(
            )
            find_points_table_index_2 =
df_update_points_table.index[df_update_points_table[group] ==
losing_team[0]].tolist(
            )
            current_points_count_1 =
df_update_points_table.at[find_points_table_index_1[0], 'Points']
            df_update_points_table.at[find_points_table_index_1,
                                      'Points'] = current_points_count_1+1

            current_points_count_2 =
df_update_points_table.at[find_points_table_index_2[0], 'Points']
            df_update_points_table.at[find_points_table_index_2,
                                      'Points'] = current_points_count_2+1

    writer = pd.ExcelWriter(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\points_table.xlsx', engine='xlsxwriter')
    workbook = writer.book
    worksheet = workbook.add_worksheet('Sheet1')
    writer.sheets['Sheet1'] = worksheet
    df_update_points_table.to_excel(
        writer, sheet_name='Sheet1', startrow=0, startcol=0, index=False)

    writer.save()
```

```
    writer.close()


def display_points_table():
    update_points_table = pd.read_excel(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\points_table.xlsx')
    df_update_points_table = pd.DataFrame(update_points_table)
    print('\n')
    print(df_update_points_table.to_string(index=False))


def match_summary():
    # Toss
    print('\n\n----------------------------------------------------------Match
Summary-----------------------------------------\n')
    match_toss = ''
    match_result = ''

    if selection == toss:
        match_toss = str(visiting_team[0].replace(
            '_', ' ')+' Won the toss and chose to ' + choose)
        print(match_toss)
        print('\n')
    else:
        match_toss = str(home_team[0].replace(
            '_', ' ')+' Won the toss and chose to ' + choose)
        print(match_toss)
        print('\n')

    print('----------------------------------------------------------First Innings
Top Performers----------------------------------\n\n')
    print(team_to_bat[0].replace('_', ' '))
    print(df_score_card_first_ing_without_index.nlargest(4, 'Runs'))

    print('\n', team_to_bowl[0].replace('_', ' '))
    print(df_bowler_list_first_ing_without_index.nlargest(3, 'Wickets'))

    print('\n\nTotal', first_ing_total, '/', first_ing_wickets)
    print('\n')

    print('----------------------------------------------------------Second Innings
Top Performers----------------------------------\n\n')
    print(team_to_bowl[0].replace('_', ' '))
    print(df_score_card_second_ing_without_index.nlargest(4, 'Runs'))
```

```
    print('\n', team_to_bat[0].replace('_', ' '))
    print(df_bowler_list_second_ing_without_index.nlargest(3, 'Wickets'))

    print('\n\nTarget', first_ing_total+1)
    print('Total', second_ing_total, '/', second_ing_wickets)
    print('\n\n')

    # Match result
    global winning_team
    global losing_team
    is_drawn = 0

    print('\n----------------------------------------------------------------Match
Result----------------------------------------------\n')

    if (second_ing_total > first_ing_total):
        match_result = str(team_to_bowl[0].replace('_', ' ')+' Won by ' +
                           str(TOTAL_WICKETS-second_ing_wickets)+' wickets')
        print(match_result)
        winning_team = team_to_bowl
        losing_team = team_to_bat

    elif (second_ing_total < first_ing_total):
        match_result = str(team_to_bat[0].replace(
            '_', ' ')+' Won by ' + str(first_ing_total-second_ing_total)+' runs')
        print(match_result)
        winning_team = team_to_bat
        losing_team = team_to_bowl

    elif (second_ing_total == first_ing_total):
        winning_team = team_to_bat
        losing_team = team_to_bowl
        is_drawn = 1
        print('\n\n----------------------------------------------------------Match
drawn-----------------------------------------------\n')

    book = load_workbook(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\matches\\' + filename_match + '.xlsx')
    writer = pd.ExcelWriter(
        r'E:\\IIT\\1st Year\\1st Trimester\\CM1601 [PRO]  Programming
Fundamentals\\Course Work\\tournament\\matches\\' + filename_match + '.xlsx',
engine='openpyxl')
    writer.book = book
```

```
    worksheet = book.worksheets[0]
    worksheet['F27'] = match_toss
    worksheet['F28'] = match_result

    writer.save()
    writer.close()
    print('\n-------------------------------------------------------------------------------
-------------------------------------------------\n')
    # -------------------------------------------------------------------------------
-----------------------Update Points table-------------------------------------------
-------------------------------------------------------
    update_points_table(winning_team, losing_team, is_drawn)
```

Coursework.py

```
      # ------------------------------------------------------------------------
------------------------------Importing Modules-----------------------------------
------------------------------------------------------------

import cricket
from openpyxl import load_workbook
import random
import pandas as pd
from operator import itemgetter
# ------------------------------------------------------------------------------
--Global variables which are used when accessing the functions-------------------
------------------------------------------------------

user_input = ''
global_exit = ''
winning_team = []
losing_team = []

print("\n\n------------------------------------------------------------------------
---------Welcome to IIT Cricket Premier League 2021-----------------------------
-------------------------------------------------------")

try:
    while user_input != 'x':
        user_input = input(
            "\n\nPress the desired number for your action... \n\nPlay a new match
- 1 \nView/edit team/player profile - 2 \nView Player Standings - 3 \nView
Tournament Standings - 4 \nPress 'x' to exit...  ")

        if user_input == '1':
            cricket.generate_random_match()
            cricket.points_table()
            cricket.toss_coin()
            cricket.first_innings()
            cricket.second_innings()
            cricket.match_summary()
        elif user_input == '2':
            cricket.team_profile_edit(user_input)
        elif user_input == '3':
            cricket.display_player_standings()
        elif user_input == '4':
            cricket.display_points_table()
        elif user_input == 'x':
```

```
            break
        else:
            print('Input value incorrect \nTry again!!!\n')
except IndexError as e:
    print('\n-------------------------------------------------------------------------
----------------------TOURNAMENT OVER-----------------------------------------------
-------------------------------------------------\n')
except Exception as e:
    print(e)

print("\n\n-------------------------------------------------------------------------
-------------------Thank you for playing!!!--------------------------------------
-------------------------------------------------")
```

Output

Main Menu

```
Press the desired number for your action...

Play a new match - 1
View/edit team/player profile - 2
View Player Standings - 3
View Tournament Standings - 4
Press 'x' to exit...  2
```

Editing menu

```
 Which group do you want to see?
 1 - group A
 2 - group B
 Or press 'x' to exit...     2
 Which team do you want to see?
 1 - Rajastan Australia
 2 - Kolkata England
 3 - Punjab Pakistan
 4 - Sunrisers SriLanka
  Select a number from 1 to 44
```

```
             PLAYER NAME
0       Dinesh Chandimal
1        Pathum Nissanka
2       Charith Asalanka
3    Dhananjaya de Silva
4      Bhanuka Rajapakse
5        Dasun Shanka(C)
6      Wanindu Hasaranga
7   Chamika Karunarathne
8    Dushmantha Chameera
9     Maheesh Theekshana
10         Nuwan Pradeep
Do you want to make any changes on team Sunrisers_SriLanka?
1 - yes
0 - no 1
```

```
Which player do you want to edit?
Select the corresponding row number :  3
What should be the change then? Angelo Mathews
```

```
0          Dinesh Chandimal
1           Pathum Nissanka
2          Charith Asalanka
3            Angelo Mathews
4         Bhanuka Rajapakse
5           Dasun Shanka(C)
6         Wanindu Hasaranga
7      Chamika Karunarathne
8       Dushmantha Chameera
9        Maheesh Theekshana
10             Nuwan Pradeep
Name: PLAYER NAME, dtype: object
your changes have been saved successfully !!!
Do you want to make any more changes on team Sunrisers_SriLanka again?
1 - yes
0 - no 0
```

Viewing Tournament Standing before the tournament begins

```
Play a new match - 1
View/edit team/player profile - 2
View Player Standings - 3
View Tournament Standings - 4
Press 'x' to exit...  4


                 Group A  Matches_A  Won  Lost  Points  |           Group B  Matches_B  Won_B  Lost_B  Points_B
             Mumbai_India          0    0     0       0  | Rajastan_Australia          0      0       0         0
       Chennai_SouthAfrica         0    0     0       0  |    Kolkata_England          0      0       0         0
          Delhi_NewZealand         0    0     0       0  |    Punjab_Pakistan          0      0       0         0
RoyalChallengers_Bangladesh        0    0     0       0  | Sunrisers_SriLanka          0      0       0         0
```

Playing a match

```
Press the desired number for your action...

Play a new match - 1
View/edit team/player profile - 2
View Player Standings - 3
View Tournament Standings - 4
Press 'x' to exit...  1
```

First Innings

```
Home Team -  RoyalChallengers_Bangladesh
Visiting Team -  Mumbai_India



Mumbai_India won the toss and chose to bat




team_to_bat Mumbai_India
team_to_bowl RoyalChallengers_Bangladesh



FOW at 34  -->  1  on over - 1 . 4 KL Rahul
FOW at 70  -->  2  on over - 3 . 3 Virat Kohli(C)
FOW at 100  -->  3  on over - 5 . 0 Rohit Sharma
FOW at 105  -->  4  on over - 5 . 2 Rishabh Pant
FOW at 117  -->  5  on over - 6 . 4 Ravindra Jadeja
FOW at 131  -->  6  on over - 7 . 4 Suryakumar Yadav
FOW at 173  -->  7  on over - 10 . 2 Hardik Pandya
FOW at 183  -->  8  on over - 11 . 0 Mohommad Shami
FOW at 187  -->  9  on over - 12 . 0 Jasprit Bumrah
FOW at 237  -->  10  on over - 14 . 4 Bhuvneshwar Kumar
```

```
--------------------------------------------------1st Innings Summary-----------------------------------------



Total- 237
wickets - 10
overs - 14 . 4
balls 88

Last dismissal ['Bhuvneshwar Kumar', 71, 25, 'Caught', ' Shamim Hossain', 8]
```

----------1st Innings Scorecard----------

Batting	Runs	Balls Faced	MOD	Bowler	Batting No
KL Rahul	34	10	Caught	Mustafizur Rahman	1
Rohit Sharma	38	13	LBW	Shamim Hossain	2
Virat Kohli(C)	19	5	Caught	Mahedi Hasan	3
Suryakumar Yadav	16	8	LBW	Taskin Ahmed	4
Rishabh Pant	5	2	Caught	Shoriful Islam	5
Ravindra Jadeja	5	3	Bowled	Mustafizur Rahman	6
Hardik Pandya	25	11	Bowled	Shoriful Islam	7
Bhuvneshwar Kumar	71	25	Caught	Shamim Hossain	8
Mohommad Shami	10	4	Bowled	Shoriful Islam	9
Jasprit Bumrah	1	2	Caught	Mustafizur Rahman	10
Yuzvendra Chahal	13	5	NOT OUT		11

----------1st Innings Bowling figures----------

Bowling	Overs	Runs	Wickets	Economy
Shoriful Islam	3.0	47	3	15.67
Mustafizur Rahman	3.0	37	3	12.33
Taskin Ahmed	3.0	60	1	20.00
Mahedi Hasan	3.0	49	1	16.33
Shamim Hossain	2.4	44	2	18.33

Second Innings

```
FOW 64  -->  1  on over - 3 . 5 Tamim Iqbal
FOW 120  -->  2  on over - 6 . 4 Shakib Al Hasan
FOW 146  -->  3  on over - 8 . 2 Liton Das
FOW 167  -->  4  on over - 9 . 5 Mahmudulla(C)
FOW 188  -->  5  on over - 11 . 1 Mushfiqur Rahim
FOW 200  -->  6  on over - 11 . 4 Shamim Hossain
FOW 231  -->  7  on over - 13 . 2 Mahedi Hasan
FOW 236  -->  8  on over - 13 . 5 Taskin Ahmed
```

```
----------------------------------------------------2nd Innings Summary---------------------------------

Total- 242
wickets - 8
overs - 14 . 0
balls 84

Last dismissal ['Taskin Ahmed', 5, 3, 'Bowled', ' Bhuvneshwar Kumar', 9]


----------------------------------------------------2nd Innings Scorecard---------------------------------


                  Runs  Balls Faced        MOD              Bowler  Batting No
Batting
Tamim Iqbal         40           13     Bowled   Bhuvneshwar Kumar           1
Liton Das           82           27        LBW   Bhuvneshwar Kumar           2
Shakib Al Hasan     16            7        LBW      Jasprit Bumrah           3
Mushfiqur Rahim     33           12     Bowled      Jasprit Bumrah           4
Mahmudulla(C)       11            6     Bowled       Hardik Pandya           5
Afif Hossain        23            7    NOT OUT                               6
Shamim Hossain      12            3        LBW      Jasprit Bumrah           7
Mahedi Hasan        14            5     Caught   Bhuvneshwar Kumar           8
Taskin Ahmed         5            3     Bowled   Bhuvneshwar Kumar           9
Mustafizur Rahman    6            1  * NOT OUT                              10
Shoriful Islam       0            0                                         11


--------------------------------------------------2nd Innings Bowling figures-------------------------------


                 Overs  Runs  Wickets  Economy
Bowling
Hardik Pandya      2.0    26        1    13.00
Yuzvendra Chahal   3.0    64        0    21.33
Jasprit Bumrah     3.0    50        3    16.67
Mohommad Shami     3.0    50        0    16.67
Bhuvneshwar Kumar  3.0    52        4    17.33
```

Match Summary

```
----------------------------------------------------Match Summary---------------------------------------

Mumbai India Won the toss and chose to bat
```

```
----------------------------------------------------First Innings Top Performers-------------------------------


Mumbai India
                  Runs  Balls Faced     MOD              Bowler  Batting No
Batting
Bhuvneshwar Kumar    71           25  Caught      Shamim Hossain           8
Rohit Sharma         38           13     LBW      Shamim Hossain           2
KL Rahul             34           10  Caught   Mustafizur Rahman           1
Hardik Pandya        25           11  Bowled      Shoriful Islam           7

 RoyalChallengers Bangladesh
                 Overs  Runs  Wickets  Economy
Bowling
Shoriful Islam     3.0    47        3    15.67
Mustafizur Rahman  3.0    37        3    12.33
Shamim Hossain     2.4    44        2    18.33


Total 237 / 10


----------------------------------------------------Second Innings Top Performers-----------------------------


RoyalChallengers Bangladesh
                Runs  Balls Faced      MOD              Bowler  Batting No
Batting
Liton Das         82           27      LBW   Bhuvneshwar Kumar           2
Tamim Iqbal       40           13   Bowled   Bhuvneshwar Kumar           1
Mushfiqur Rahim   33           12   Bowled      Jasprit Bumrah           4
Afif Hossain      23            7  NOT OUT                               6

 Mumbai India
                 Overs  Runs  Wickets  Economy
Bowling
Bhuvneshwar Kumar  3.0    52        4    17.33
Jasprit Bumrah     3.0    50        3    16.67
Hardik Pandya      2.0    26        1    13.00


Target 238
Total 242 / 8
```

```
-----------------------------------------------------------Match Result-------------------------------------------

RoyalChallengers Bangladesh Won by 2 wickets

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

Viewing Tournament Standing after playing a match

```
Press the desired number for your action...

Play a new match - 1
View/edit team/player profile - 2
View Player Standings - 3
View Tournament Standings - 4
Press 'x' to exit...  4


                  Group A  Matches_A  Won  Lost  Points  |           Group B  Matches_B  Won_B  Lost_B  Points_B
              Mumbai_India          1    0     1       0  | Rajastan_Australia          0      0       0         0
         Chennai_SouthAfrica          0    0     0       0  |    Kolkata_England          0      0       0         0
            Delhi_NewZealand          0    0     0       0  |    Punjab_Pakistan          0      0       0         0
RoyalChallengers_Bangladesh          1    1     0       2  | Sunrisers_SriLanka          0      0       0         0
```

Viewing Player Standing after playing a match

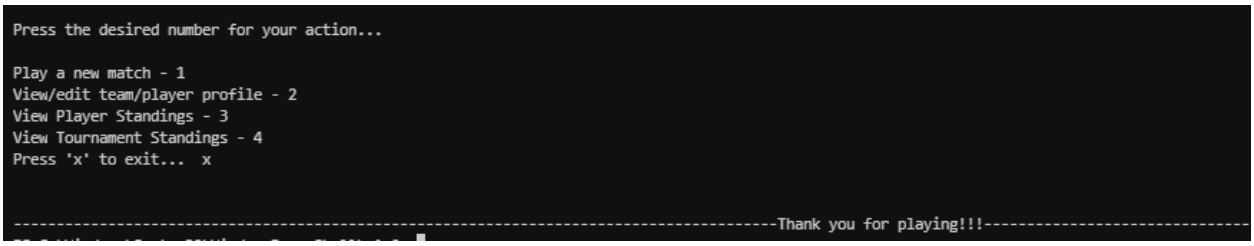
```
Press the desired number for your action...

Play a new match - 1
View/edit team/player profile - 2
View Player Standings - 3
View Tournament Standings - 4
Press 'x' to exit...  3


Top 5 run scores of the tournament
      PLAYER NAME  TOTAL RUNS
        Liton Das          82
Bhuvneshwar Kumar          71
      Tamim Iqbal          40
     Rohit Sharma          38
        KL Rahul           34


Top 5 wicket takers of the tournament
      PLAYER NAME  WICKETS
Bhuvneshwar Kumar        4
   Jasprit Bumrah        3
Mustafizur Rahman        3
   Shoriful Islam        3
   Shamim Hossain        2
```

Exiting from the main menu



Match Summary example

A	B	C	D	E	F
Batting	**Runs**	**Balls Faced**	**MOD**	**Bowler**	**Batting No**
Jason Roy	1	2	LBW	Nuwan Pradeep	1
Jos Buttler	20	7	LBW	Wanindu Hasaranga	2
Dawid Malan	11	6	LBW	Maheesh Theekshana	3
Jonny Bairstow	5	2	LBW	Dushmantha Chameer	4
Eoin Morgan(C)	25	9	NOT OUT		5
Moeen Ali	0	2	LBW	Wanindu Hasaranga	6
Liam Livingstone	24	7	Bowled	Nuwan Pradeep	7
Chris Woakes	0	1	Caught	Nuwan Pradeep	8
Chris Jordan	2	2	LBW	Maheesh Theekshana	9
Adil Rashid	4	2	Caught	Maheesh Theekshana	10
Mark Wood	6	2	Caught	Maheesh Theekshana	11

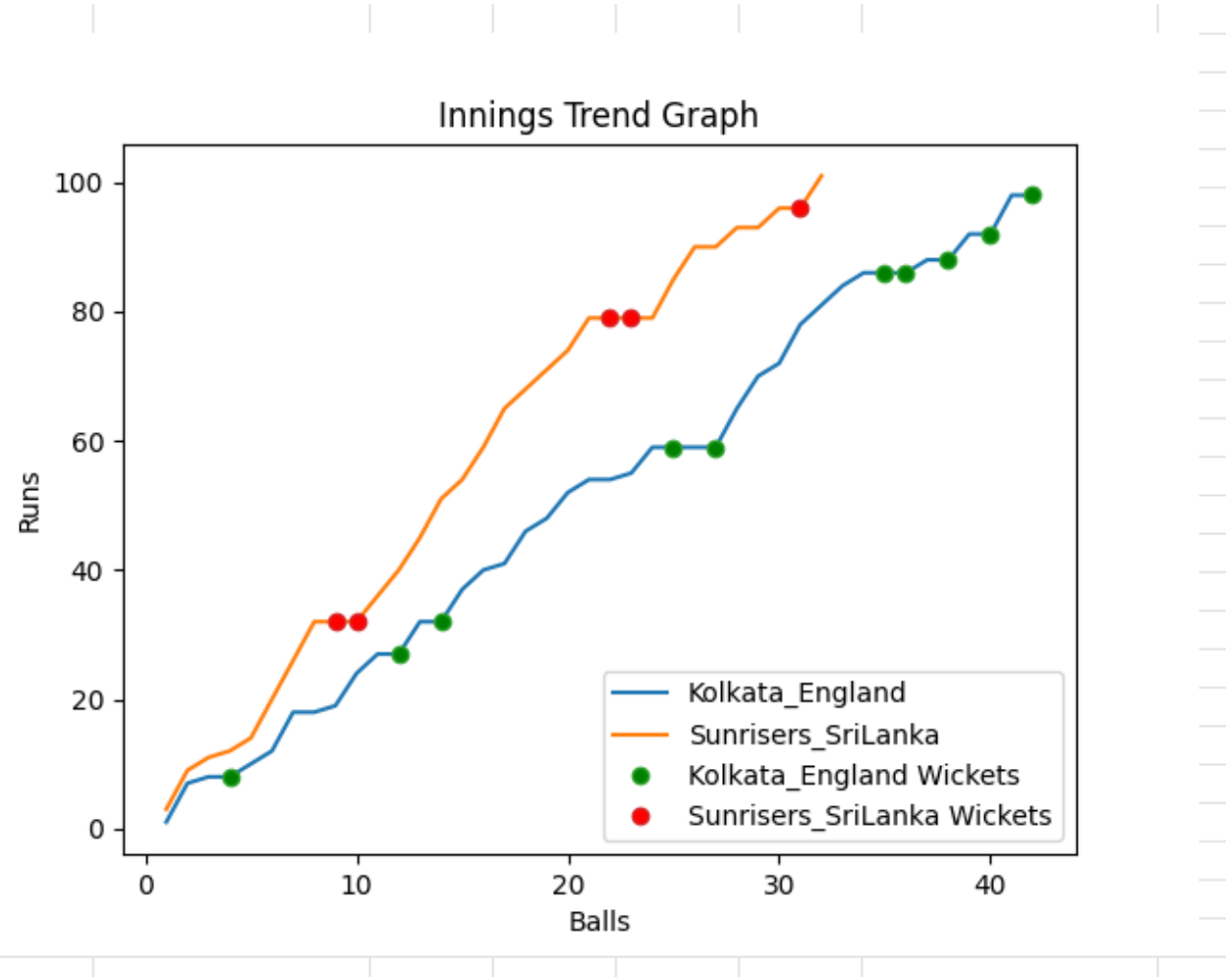
Total	**Wickets**	**Overs**	**Balls**
98	10	7.0	42

Bowling	**Overs**	**Runs**	**Wickets**	**Economy**
Dushmantha Chameera	1.0	19	1	19
Chamika Karunarathne	1.0	13	0	13
Wanindu Hasaranga	1.0	13	2	13
Nuwan Pradeep	2.0	26	3	13
Maheesh Theekshana	2.0	27	4	13.5

J	K	L	M	N	O
Batting	**Runs**	**Balls Faced**	**MOD**	**Bowler**	**Batting No**
Dinesh Chandimal	23	6	Bowled	Adil Rashid	1
Pathum Nissanka	29	9	NOT OUT		2
Charith Asalanka	0	1	Bowled	Adil Rashid	3
Dhananjaya de Silva	30	8	LBW	Chris Woakes	4
Bhanuka Rajapakse	0	1	LBW	Chris Woakes	5
Dasun Shanka(C)	14	6	Caught	Mark Wood	6
Wanindu Hasaranga	5	1	* NOT OUT		7
Chamika Karunarathne	0	0			8
Dushmantha Chameera	0	0			9
Maheesh Theekshana	0	0			10
Nuwan Pradeep	0	0			11

Total	Wickets	Overs	Balls
101	5	5.2	32

Bowling	Overs	Runs	Wickets	Economy
Adil Rashid	1.0	20	2	20
Chris Jordan	1.0	28	0	28
Chris Woakes	1.0	11	2	11
Liam Livingstone	1.0	17	0	17
Mark Wood	1.2	25	1	20.83


Innings Trend Graph
Runs
Balls
0
20
40
60
80
100
0
10
20
30
40
Kolkata_England
Sunrisers_SriLanka
Kolkata_England Wickets
Sunrisers_SriLanka Wickets

Test Plan

01)Generating two teams from the two groups randomly and picking one match

- Input :

```
match_between_A = random.sample(Group_A, 2)
match_between_B = random.sample(Group_B, 2)


chosen_match = [match_between_A,match_between_B]
match_between = random.choice(chosen_match)


print(match_between)
```

- Expected output:

  Ex: Punjab Pakistan vs Kolkata England

- Actual output:

```
[['Punjab_Pakistan', 'E:\\\\IIT\\\\1st Year\\\\1st Trimester\\\\CM1601 [PRO]  Programming
Fundamentals\\\\Course Work\\\\team_data\\\\Punjab_Pakistan\\\\Punjab_Pakistan.xlsx'], ['Kolkata_England',
'E:\\\\IIT\\\\1st Year\\\\1st Trimester\\\\CM1601 [PRO]  Programming Fundamentals\\\\Course
Work\\\\team_data\\\\Kolkata_England\\\\Kolkata_England.xlsx']]
```

02)Assigning the two picked teams from the match as Home team and Visiting team

- Input :

```
coin = ["heads", "tails"]
options = ['bat','bowl']

visiting_team = random.choice(match_between)

if visiting_team in match_between:
    match_between.remove(visiting_team)

home_team = match_between[0]

print('Home Team - ', home_team)
print('Visiting Team - ', visiting_team)
```

- Expected output:

  Home team – Delhi New Zealand
  Visiting team – Chennai South Africa

- Actual output:

```
Home Team -  ['Delhi_NewZealand', 'E:\\\\IIT\\\\1st Year\\\\1st Trimester\\\\CM1601 [PRO]  Programming
Fundamentals\\\\Course Work\\\\team_data\\\\Delhi_NewZealand\\\\Delhi_NewZealand.xlsx']
Visiting Team -  ['Chennai_SouthAfrica', 'E:\\\\IIT\\\\1st Year\\\\1st Trimester\\\\CM1601 [PRO]
Programming Fundamentals\\\\Course Work\\\\team_data\\\\Chennai_SouthAfrica\\\\Chennai_SouthAfrica.xlsx']
```

03)Displaying batting card without index column

- Input :

```
#add batsman_list to score_card_first_ing
score_card_first_ing = batsman_list

# sort score_card_first_ing to the original batting order
sorted_list = sorted(score_card_first_ing, key=itemgetter(5))


# convert score_card_first_ing to a data frame for displaying
df_score_card_first_ing = pd.DataFrame(sorted_list)

df_score_card_first_
```

- Expected output:

Batting	Runs	Balls Faced	MOD	Bowler	Batting No
Jason Roy	61	21	Bowled	Shadab Khan	1
Jos Buttler	7	6	Bowled	Imad Wasim	2
Dawid Malan	6	4	LBW	Hasan Ali	3
Jonny Bairstow	31	10	Bowled	Shaheen Afridi	4
Eoin Morgan(C)	36	11	Bowled	Shadab Khan	5
Moeen Ali	3	2	LBW	Imad Wasim	6
Liam Livingstone	8	4	Bowled	Shadab Khan	7
Chris Woakes	46	13	Bowled	Hasan Ali	8
Chris Jordan	9	5	NOT OUT		9
Adil Rashid	0	1	Bowled	Hasan Ali	10
Mark Wood	5	6	LBW	Imad Wasim	11

- Actual output:

	Batting	Runs	Balls Faced	MOD	Bowler	Batting No
0	Jason Roy	61	21	Bowled	Shadab Khan	1
1	Jos Buttler	7	6	Bowled	Imad Wasim	2
2	Dawid Malan	6	4	LBW	Hasan Ali	3
3	Jonny Bairstow	31	10	Bowled	Shaheen Afridi	4
4	Eoin Morgan(C)	36	11	Bowled	Shadab Khan	5
5	Moeen Ali	3	2	LBW	Imad Wasim	6
6	Liam Livingstone	8	4	Bowled	Shadab Khan	7
7	Chris Woakes	46	13	Bowled	Hasan Ali	8
8	Chris Jordan	9	5	NOT OUT		9
9	Adil Rashid	0	1	Bowled	Hasan Ali	10
10	Mark Wood	5	6	LBW	Imad Wasim	11

04)Showing a summary of an innings' top performers

- Input :

```
print(team_to_bat[0].replace('_', ' '))
print(df_score_card_first_ing_without_index.nlargest(11, 'Runs'))


print('\n',team_to_bowl[0].replace('_', ' '))
print(df_bowler_list_first_ing_without_index.nlargest(5, 'Wickets'))

print('\nTotal',first_ing_total,'/',first_ing_wickets)
```

- Expected output:

```
Kolkata England
                Runs  Balls Faced     MOD           Bowler  Batting No
Batting
Jason Roy         61           21  Bowled      Shadab Khan           1
Chris Woakes      46           13  Bowled        Hasan Ali           8
Eoin Morgan(C)    36           11  Bowled      Shadab Khan           5
Jonny Bairstow    31           10  Bowled   Shaheen Afridi           4

 Punjab Pakistan
             Balls  Runs  Wickets  Economy
Bowling
Shadab Khan     12    22        3       22
Hasan Ali       18    40        3       40
Imad Wasim      17    24        3       24

Total 212 / 10
```

- Actual output:

```
Kolkata England
                 Runs  Balls Faced      MOD          Bowler  Batting No
Batting
Jason Roy          61           21   Bowled      Shadab Khan           1
Chris Woakes       46           13   Bowled        Hasan Ali           8
Eoin Morgan(C)     36           11   Bowled      Shadab Khan           5
Jonny Bairstow     31           10   Bowled   Shaheen Afridi           4
Chris Jordan        9            5  NOT OUT                           9
Liam Livingstone    8            4   Bowled      Shadab Khan           7
Jos Buttler         7            6   Bowled       Imad Wasim           2
Dawid Malan         6            4      LBW        Hasan Ali           3
Mark Wood           5            6      LBW       Imad Wasim          11
Moeen Ali           3            2      LBW       Imad Wasim           6
Adil Rashid         0            1   Bowled        Hasan Ali          10

 Punjab Pakistan
               Balls  Runs  Wickets  Economy
Bowling
Shadab Khan       12    22        3       22
Hasan Ali         18    40        3       40
Imad Wasim        17    24        3       24
Shaheen Afridi    18    63        1       63
Haris Rauf        18    63        0       63

Total 212 / 10
```

05)Converting the random match generation code into a function

- Input :

```
def generate_random_match():
    match_between_A = random.sample(Group_A, 2)
    match_between_B = random.sample(Group_B, 2)

    chosen_match = [match_between_A, match_between_B]
    match_between = random.choice(chosen_match)
    print(match_between)
```

- Expected output:

  Ex: Punjab Pakistan vs Kolkata England

- Actual output:

```
  File "e:\IIT\1st Year\1st Trimester\CM1601 [PRO]  Programming Fundamentals\Course Work\files\coursework.py", line 195, in toss
    visiting_team = random.choice(match_between)
  File "C:\Users\nadun\AppData\Local\Programs\Python\Python39\lib\random.py", line 346, in choice
    return seq[self._randbelow(len(seq))]
IndexError: list index out of range
```