Brandon Tonge
ID - 201358937

COMP101 – Assignment 06

Python Code –

```
# 201358937 Tonge_Brandon-CA06.py
# November 2018
# This program will accept multiple user inputs regarding the production of a stage show.
# It will then calculate how long it will take to make a profit if the show has a full
# house every night. It also give the option to generate 5 days where a random number of
# seats have been sold (over 40%) and then use the average to again predict the number of
# days until a profit is made.

import random
import math

# Main Function
def main():

    # Main Menu
    print("\n---Main Menu---")
    print("A - Theatre")
    print("E - Extend")
    print("X - Exit Program")
    print("")
    choice = str.upper(input("Please select an option from the menu: "))

    # TEST
    # print(choice)

    # Function Selection
    if(choice == "A"):
        theatre()

    elif(choice == "E"):
        extended()

    elif(choice == "X"):
        exit()

    else:
        print("\nPlease enter a valid choice!\n")
        main()


# Theatre Function
def theatre():

    # Accept the users inputs
    while True:
        prod_cost = input("\nPlease enter the overall cost of the production: £")
        try:
```

```
            prod_cost = float(prod_cost)
            break
        except:
            print("\nPlease enter a valid number!")
            continue

    print("\nWhat is the cost of tickets in -")
    while True:
        bandA = input("Band A: £")
        try:
            bandA = float(bandA)
            break
        except:
            print("\nPlease enter a valid number!")
            continue

    while True:
        bandB = input("Band B: £")
        try:
            bandB = float(bandB)
            break
        except:
            print("\nPlease enter a valid number!")
            continue

    while True:
        bandC = input("Band C: £")
        try:
            bandC = float(bandC)
            break
        except:
            print("\nPlease enter a valid number!")
            continue

    while True:
        drinkws = input("\nPlease enter the wholesale cost of drinks: £")
        try:
            drinkws = float(drinkws)
            break
        except:
            print("\nPlease enter a valid number!")
            continue

    while True:
        programws = input("\nPleas enter the wholesale cost of programs: £")
        try:
            programws = float(programws)
            break
        except:
            print("\nPlease enter a valid number!")
            continue
```

Brandon Tonge
ID - 201358937

```
    # User choice in which function to run
    while True:
        print("\n- Show Turnout -")
        print("Assume full house   =  1")
        print("Assume random house =  2")
        choice = input("Please choose an option: ")
        if(choice == "1" or choice == "2"):
            try:
                choice = int(choice)
                break
            except:
                print("Please enter one of the options below!")
                continue
        else:
            print("Please enter one of the options below!")
            continue


    if(choice == 1):
        fullhouse(prod_cost, bandA, bandB, bandC, drinkws, programws)

    else:
        randomhouse(prod_cost, bandA, bandB, bandC)

    main()


# Full House Function
def fullhouse(prod_cost, bandA, bandB, bandC, drinkws, programws):

    # Create list with a full house
    numc = 5
    numr = 5
    seats = [[1 for row in range(numr)] for col in range(numc)]

    # Test
    print()
    for i in range(numr):
        print(seats[i])

    # Count how many seats are filled for each reference in seats
    seats_filled = sum(i.count(1) for i in seats)

    # Money made per each band and consumables
    prof_banda = float((seats[0].count(1) + (seats[1].count(1))) * bandA)
    prof_bandb = float((seats[3].count(1) + (seats[3].count(1))) * bandB)
    prof_bandc = float((seats[4].count(1)) * bandC)
    prof_drinks = float(drinkws + (drinkws / 2))
    prof_programs = float(programws + (programws / 4))
    overall_drinks = float(seats_filled * prof_drinks)
    overall_programs = float(math.ceil(float(seats_filled / 2)) * prof_programs)
```

Brandon Tonge
ID - 201358937

```
    # Test
    # print(prof_banda, prof_bandb, prof_bandc)

    print("\n- Seat Sales Alone - ")
    # Overall price of seats for the night
    overall_band = float(prof_banda + prof_bandb + prof_bandc)
    print(f"The total for seat sales is = £{overall_band}")
    print(f"- Band A = £{prof_banda}")
    print(f"- Band B = £{prof_bandb}")
    print(f"- Band C = £{prof_bandc}")

    # Days until a profit is made
    days_profit = math.ceil(float(prod_cost / overall_band))
    print(f"With production costing £{prod_cost} it will take {days_profit} days to make a
profit.")

    # Print seats and consumables for the night
    print("\n- Seat and Consumables -")
    print(f"The total for seat sales is = £{overall_band}")
    print(f"The drinks sales are = £{overall_drinks}")
    print(f"The program sale are = £{overall_programs}")

    # Days until a profit is made
    days_profit2 = math.ceil((float(prod_cost / (overall_band + overall_drinks +
overall_programs))))
    print(f"With production costing £{prod_cost} it will take {days_profit2} days to make a
profit.")
    main()


# Random House Function
def randomhouse(prod_cost, bandA, bandB, bandC):

    prof_banda = 0
    prof_bandb = 0
    prof_bandc = 0

    # Create list and randomly populate it
    for i in range(5):
        numc = 5
        numr = 5

        seats = [[0 for row in range(numr)] for col in range(numc)]

        # Populate the list with 10 - 25 "1" in randomly selected positions
        for pos in random.sample(range(25), random.randint(10, 25)):
            seats[pos // 5][pos % 5] = 1

        # Test print
        print()
        print(f"   - Day {i + 1} -")
        for i in range(numr):
```

```
            print(seats[i])


        # Money made per each band
        prof_banda = prof_banda + float(((seats[0].count(1)) + (seats[1].count(1))) * bandA)
        prof_bandb = prof_bandb + float(((seats[2].count(1)) + (seats[3].count(1))) * bandB)
        prof_bandc = prof_bandc + float((seats[4].count(1)) * bandC)


    # Test
    # print("\n", prof_banda, prof_bandb, prof_bandc)

    # Overall price of seats for the night
    average_band = (float(prof_banda + prof_bandb + prof_bandc)) / 5
    print(f"The average profit over five nights is = £{average_band}")

    # Days until a profit is made
    days_profit = math.ceil(float(prod_cost / average_band))
    print(f"With production costing £{prod_cost} it will take {days_profit} days to make a
profit.")

    main()


# Extended Function
def extended():

    print("Extended")
    main()


main()
```

Testing Table –

In this testing table I am going to test both the full house and the random house functions with the same data, this way I can check to see if the maths in each one is working correctly. As for the theatre function, I have repeated the same validation for each input, so I don't need to extensively test each one individually. I will include validated testing for the first one, but this can apply to all the subsequent ones also.

Inputs	Expected Output	Actual Output	Comments
Production Cost – "Ten" 5000	I expect the first input to cause and error message and the second one to be allowed.	As I predicted the first one didn't crash the program and produced an error message, the second input passed through fine.	This is a test of the try and except loop I have put in place on each input. This will only allow the user to continue if they enter a value that can be cast to a float otherwise it will produce an error message. Here I can see that it works as I expected and that lets me know it will also work for all the other inputs needed from the user.

Brandon Tonge
ID - 201358937

Production Cost – 10000 Band A – 15 Band B – 10 Band C – 5 Drinks – 4 Program – 2 Option – 1	I expect the output to match the example given in the specification sheet.	As I predicted the output was exactly the same as the one shown in the specification sheet. This includes the days to make a profit as well as the overall seats costs.	This test was to see if the maths behind the full house function was working. Both the seat sales alone matched as well as the seats plus the consumables. Each of the outputs is exactly as predicted in the specification sheet. I can also see that the "math.ceil" is working in the program sales output. This is show by the output being for 13 people and not 12.5.
Production Cost – 10000 Band A – 15 Band B – 10 Band C – 5 Drinks – 4 Program – 2 Option – 2	Each output will be different as the program will create 5 random lists but I have included a print of each list, so I can work it out once it has been printed.	In this test I found that the average profit over five nights and the days needed to make a profit to be accurate.	Here my program prints out five randomly filled lists. From here I was able to count each filled seat on the bands and work out if the program was calculating the correct answers. To aid in the I have included a commented-out test print that will show the sum of each band across every list created. I then divided the sum of these by 5 to get the average. Again, I used the "math.ceil" when calculating the days to make a profit and found it to be accurate as well.
Production Cost – 5000 Band A – 20 Band B – 15 Band C – 10 Drinks – 5 Program – 2 Option – 2	Each output will be different as the program will create 5 random lists, but I have included a print of each list, so I can work it out once it has been printed.	In this test I found that the average profit over five nights and the days needed to make a profit to be accurate.	I ran the random function again with my own figures just to double check that each part was working correctly. Again, I used the printed lists to manually calculate the average profit as well as the days needed to make a profit overall.
Production Cost – 7000 Band A – 20 Band B – 17.50 Band C –12.50 Drinks – 3.50 Program – 2.50 Option – 1	I expect the output to be accurate and will calculate it manually once the program has run.	As I expected the program ran fine and the outputs were accurate.	I used this test to not only double check the maths in the full house function but also make sure the program can handle floats. As I expected the program worked out fine and each of the outputs were representative of the inputs I used.

Pseudocode –

OUTPUT "enter cost of production"
INPUT cost of production
STORE in "cost" variable

OUTPUT "enter cost of Band A"
INPUT cost of Band A
STORE in "BandA" variable

OUTPUT "enter cost of Band B"

Brandon Tonge
ID - 201358937

INPUT cost of Band B
STORE in “BandB” variable

OUTPUT “enter cost of Band C”
INPUT cost of Band C
STORE in “BandC” variable

OUTPUT “enter cost of drinks”
INPUT cost of drinks
STORE in “drinks” variable

OUTPUT “enter cost of programs”
INPUT cost of programs
STORE in “programs” variable

OUTPUT “enter user choice”
INPUT choice
STORE in “choice” variable

Full House –

CREATE 5 x 5 list and populate with 1’s

COUNT the first two rows of the list
STORE in band A variable

COUNT rows three and four of the list
STORE in band B variable

COUNT the last row of the list
STORE in band C variable

CALCULATE the profit of band A, B and C by multiplying the number of 1’s by the inputted prices per seat
STORE the sum of these in the profit variable

CALCULATE the profit from drinks by multiplying the number of seats by the cost of drinks plus 50%
STORE in the drinks profit variable

CALCULATE the profit from the programs by multiplying the number of seats by the cost of programs plus 25%
STORE in the program profit variable

CALCULATE the days needed to make a profit by dividing the production cost by the profit variable (use formatting to store a relevant answer)
STORE in the days to profit variable

PRINT the overall profit of the seats
PRINT the drinks profit variable
PRINT the program profit variable

Brandon Tonge
ID - 201358937

PRINT the production cost
PRINT the days needed to make a profit

Random House –

LOOP
    CREATE a 5 x 5 list and populate with 0's
    RANDOM use random function to distribute 1's throughout the list (between 10 and 25)
    PRINT the randomly populate list

    CALCULATE the profit of band A, B and C by multiplying the number of 1's by the inputted prices per seat
    STORE the sum of these in the relevant band (A,B,C) variable

CALCULATE the average profit per each band by adding the five variables created from the five lists and then dividing them by five.
STORE in the average profit variable

CALCULATE the days needed to make a profit by dividing the production cost variable by the average profit variable
STORE in the days profit variable

PRINT the average days variable
PRINT the production cost variable
PRINT the days until profit variable