Project 1 Design Document

**Project Requirements:**

In this program, we are generating revenue report a local cinema in a format of **Invoice**. The 80% of **Gross Total** would send to the movie distributor, by this program we can calculate the gross total, distributor revenue & the net profit of the cinema of particular day. We would pass **Movie Name, Adult Tickets & Children Tickets** as inputs**.** The price of **Adult** ticket is **$7.50** & for **Children’s** it’s **$4.75**. As the number(amount) of tickets are constants, so we can take **constants** in program. If they weren’t so, we can also take that as an input in the starting of the program. But as they are fixed, we’d **hard code** it.

**Program Inputs:**

* Number of adult tickets sold
  + - adultTicketsSold
    - **Positive integer** only acceptable in Input Type (adultTicketsSold >= 0)
* Number of children tickets sold
  + - childTicketsSold
    - **Positive integer** only acceptable in Input Type (childTicketsSold >= 0)
* Movie Name
  + - cstring (data\_type char[]) movieName
    - Any **string of characters** is allowed/acceptable here

**Program Output:**

* Price of adult Ticket
  + - const double adultPrice
    - We are considering adultPrice const because it would not change though out the program & double for the point/floating values in the input.
    - adultPrice = 7.50
* Price of children’s ticket
  + - const double childPrice
    - We are considering adultPrice const because it would not change though out the program & double for the point/floating values in the input.
    - childPrice = 4.75
* Gross box office revenue
  + - double grossBoxRevenue
    - The output would be represented till 2 points after decimal. Ex: 120.30
    - This is the total revenue of the cinema of a particular day.
    - grossBoxRevenue = (adultPrice \* adultTicketsSold) + (childPrice \* childTicketsSold)
    - **Note**: Could not exceed more than 2 points after decimal.
* Distributor Amount
* double distributorAmount
* The output would be represented till 2 points after decimal. Ex: 120.30
* The 80% of revenue would be sent to the Movie Distributor.
* distributorAmount = grossBoxRevenue \* 0.80
* Net Profit
* double netProfit
* The output would be represented till 2 points after decimal. Ex: 120.30
* After subtracting 80% revenue of distributors the final profit/out would be the profit of the cinema
* netProfit = grossBoxRevenue – distributorAmount

**Test Plan:**

Program inputs

First input will be a string asking name of the movie in character, second input would be the total adult tickets sold in a day in Integer, further on third input it’d ask to enter the total child tickets sold in a day in Integer.

**Integer limit (num>0)**

**Input Cases:**

1. Freaky 211 73
2. The Invisible Man 80 50
3. Hunter Hunter 101.5 33
4. 123 12 5

**Note:** adultTicketsSold, childTicketsSold, adultPrice, grossBoxRevenue, distributorAmount, netProfit can’t be display over 2 points after decimal.

**Test Output:**

1. **Variables state**

movieName = Freaky

adultTicketsSold = 211

childTicketsSold = 73

grossBoxRevenue = (221 \* 7.50) + (73 \* 4.75)

distributorAmount = grossBoxRevenue \* 0.8

netProfit = grossBoxRevenue – distributorAmount

**output should be:**

Movie Name: Freaky

Gross Box Office Revenue: $ 1929.25

Distributor Amount: $ 1543 .40

Net Profit: $ 385.85

Test Case Pass

1. **Variables state**

movieName = The Invisible Man

adultTicketsSold = 80

childTicketsSold = 50

grossBoxRevenue = (80 \* 7.50) + (50 \* 4.75)

distributorAmount = grossBoxRevenue \* 0.8

netProfit = grossBoxRevenue – distributorAmount

**output should be:**

Movie Name: The Invisible Man

Gross Box Office Revenue: $ 837.50

Distributor Amount: $ 670

Net Profit: $ 167.50

Test Case Pass

1. **Variables state**

movieName = Hunter Hunter

adultTicketsSold = 101.5 [It’d read input as integer because you can’t sell ticket in decimals]

childTicketsSold = 33

grossBoxRevenue = (101 \* 7.50) + (33 \* 4.75)

distributorAmount = grossBoxRevenue \* 0.8

netProfit = grossBoxRevenue – distributorAmount

**output should be:**

Movie Name: Hunter Hunter

Gross Box Office Revenue: $ 914.25

Distributor Amount: $ 731.40

Net Profit: $ 182.25

Test Case Pass with some bug

1. **Variable State**

movieName = 123 [The movie name is in numeric form but as we are reading string input so it’d be fine]

adultTicketsSold = 12

childTicketsSold = 3

grossBoxRevenue = (12 \* 7.50) + (3 \* 4.75)

distributorAmount = grossBoxRevenue \* 0.8

netProfit = grossBoxRevenue – distributorAmount

**output should be:**

Movie Name: 123

Gross Box Office Revenue: $ 104.25

Distributor Amount: $ 83.40

Net Profit: $ 20.85

Test Case Pass

**Solution Overview:**

Algorithm steps for the Program.

1. Start program with declaring Variables for user input & data manipulations

adultPrice is 7.30, childPrice is 4.75

adultTicketsSold, childTicketsSold

movieName

Declaring Variables for output

grossBoxRevenue, distributorAmount, netProfit

1. Input from user
   * + What is the name of the movie? <String>
     + How many adult tickets sold? <Integer>
     + How many children tickets sold? <Integer>

After reading input in movieName, adultTicketsSold, childTicketsSold respectively.

1. After input the calculation would be process on the input data

Find grossBoxRevenue **(**adultTicketsSold **x** adultPrice**)** **+ (**childTicketsSold **x** childPrice**)**

1. Now, calculate distributorAmount

distributorAmount **(**grossBoxRevenue **x** 0.8**)**

1. At last, calculating netProfit

netProfit **(**grossBoxRevenue **-** distributorAmount**)**

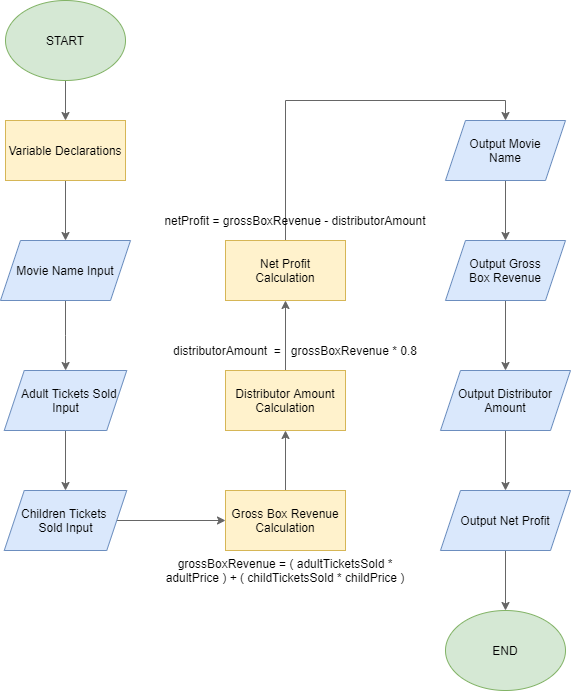
1. So, we have calculated grossBoxRevenue, distributorAmount, netProfit

Now we can print the desired output on the screen up to 2 decimal points.

* + - Print **movieName**
    - Print **grossBoxRevenue**
    - Print **distributorAmount**
    - Print **netProfit**

1. End of Program

**Algorithm Flowchart:**

****