

***Name: Muhammad Zain***

***Roll No: SU92-BSAIM-F24-092***

**Mini Project 2**

**Step 1:**

class MovieBudget:

This creates a class named MovieBudget. A class is like a blueprint that groups data and methods together.

**Step 2:**

def \_\_init\_\_(self):

self.movies = [

("Dangal", 80000000),

("Memento", 150000000),

("PK", 85000000),

("Pushpa", 55000000),

("Sholay", 30000000),

("3 ideots", 215000000),

("Jawan", 100000000)

]

This is the constructor. It runs when you create a new object.  
It makes a list named self.movies that stores movies and their budgets. Each movie is stored as a pair (tuple): (name, budget).

**Step 3:**

def add\_movies(self):

numbers = int(input("How many movies to add: "))

for i in range(numbers):

name = input("Enter movie name: ")

budget = int(input("Enter movie budget: "))

self.movies.append((name, budget))

This method asks the user how many new movies they want to add.  
Then it loops that many times, asking for movie name and budget each time.  
Finally, it adds the new movie into the self.movies list.

**Step 4:**

def average\_budgets(self):

total\_budget = 0

for movie in self.movies:

total\_budget += movie[1]

return total\_budget / len(self.movies)

This method calculates the average budget of all movies.  
It starts with total\_budget = 0.  
For each movie, it adds the budget (movie[1]) to the total.  
Then it divides the total by the number of movies and returns the average.

**Step 5:**

def find\_budget(self, average\_budget):

print("Movies budget higher than average:")

count = 0

for movie in self.movies:

name, budget = movie

if budget > average\_budget:

print(f"{name} — higher by {budget - average\_budget}")

count += 1

print("Total movies with budget higher than average:", count)

This method checks which movies have a budget greater than the average.  
For each movie, it compares its budget to the given average.  
If the budget is higher, it prints the movie name and how much higher the budget is.  
It also increases the count of movies above average.  
At the end, it prints how many such movies there are.

**Step 6:**

total = MovieBudget()

total.add\_movies()

average = total.average\_budgets()

print("Average Budget:", average)

total.find\_budget(average)

* total = MovieBudget() makes a new object.
* total.add\_movies() allows the user to add more movies.
* average = total.average\_budgets() calculates the average budget.
* It prints the average.
* total.find\_budget(average) shows which movies are above average.

