Practical Assignment 1

As per example 1 interactions, the program seems fine.

As per example 2 interactions,

Incorrect behaviour:

The program does not display the total number of each kind of tickets sold. The values for adult_tickets, child_tickets, and concession_tickets only represent the values from the last transaction.

Offending lines:

print(f"Adult tickets sold: {adult_tickets}")

print(f"Child tickets sold: {child_tickets}")

print(f"Concession tickets sold: {concession_tickets}")

As per example 3 interactions,

Incorrect behaviour:

The program doesn't check for invalid input when the user is prompted to enter “Is there a new ticket sale? (yes/no):”. If the user enters anything other than "no", the program will accept it and proceed with the next input. This can cause unexpected behaviour and make the program difficult to use.

Offending line:

The code does not have a check for invalid input when the user is prompted to enter “Is there a new ticket sale? (yes/no):”

As per example 4 interactions,

Incorrect behaviour 1:

The program enforces that the input for the number of tickets sold is a positive integer or 0. If the user enters a non-integer value, the program will raise a ValueError, but it does not give the user an informative error message, making it difficult for the user to know what went wrong.

Offending line:

The following line of code raises a ValueError when a non-integer value is entered, but it does not give an informative error message:

except ValueError:

print("Invalid input. Please enter a non-negative integer.")

Incorrect behaviour 2:

The program enforces that the input for the number of tickets sold is a positive integer or 0. If the user enters a integer value less than 0, the program will raise same ValueError, but it does not give the user an informative error message, making it difficult for the user to know what went wrong.

Offending line:

The following line of code raises same ValueError when a integer value less than 0 is entered, but it does not give an informative error message:

```
if adult_tickets < 0 or child_tickets < 0 or concession_tickets < 0:

        raise ValueError
```

and

```
except ValueError:

print("Invalid input. Please enter a non-negative integer.")
```

Incorrect behaviour 3:

The program enforces that the input for the number of tickets sold is a positive integer or 0. If the user enters 0 for all ticket types, the program will raise same ValueError, rather than displaying something like "There must be at least one ticket in the transaction!" and it does not give the user an informative error message, making it difficult for the user to know what went wrong.

Offending line:

The following line of code raises same ValueError when a user enters 0 for all ticket types, rather than displaying something like "There must be at least one ticket in the transaction!":

```
if adult_tickets == 0 and child_tickets == 0 and concession_tickets == 0:

        raise ValueErrorand

except ValueError:

print("Invalid input. Please enter a non-negative integer.")
```

Incorrect behaviour 4:

The program enforces that the input for the number of tickets sold is a positive integer or 0. If the user enters valid value, the program runs without error but calculates only the total_sales and not each kind of total ticket types.

Offending line:

The code does not have the calculation of each kind of total ticket types.