Experiment – 1 a: TypeScript

Name of Student	**Sneha Patra**
Class Roll No	**D15A_40**
D.O.P.	**23/01/2025**
D.O.S.	**30/01/2025**
Sign and Grade	

Aim: Write a simple TypeScript program using basic data types (number, string, boolean) and operators.

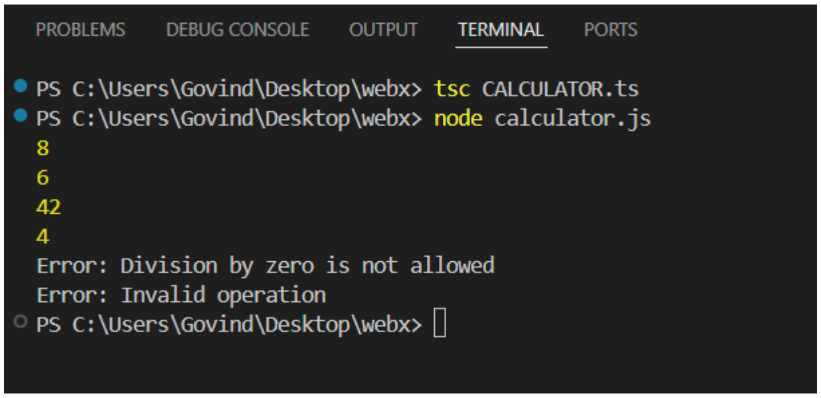
Overview of Task Performed:

This experiment delved into key TypeScript features, focusing on object-oriented principles. We constructed a hierarchy of student classes, illustrating inheritance and method specialization. We also explored composition by integrating a library account with student data. Furthermore, an employee management system was built using interfaces to define a contract for different employee roles, ensuring consistent structure and type safety across manager and developer classes.

GitHub Link: https://github.com/Sneha0321/WebX_Exp1

Output:

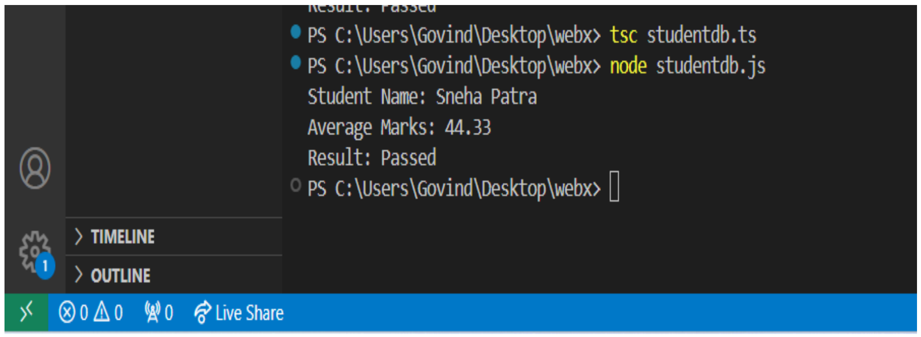
a) Create a calculator in TypeScript that uses basic operations like addition, subtraction, multiplication, and division. It also gracefully handles invalid operations and division by zero..



```
PS C:\Users\Govind\Desktop\webx> tsc CALCULATOR.ts
PS C:\Users\Govind\Desktop\webx> node calculator.js
8
6
42
4
Error: Division by zero is not allowed
Error: Invalid operation
PS C:\Users\Govind\Desktop\webx>
```

The student-related output displayed detailed information about individual students, including any specific attributes for graduate students, and confirmed the successful integration of library account details with student records. This output validated the correct application of inheritance and composition in organizing student data.

b) Design a Student Result database management system using TypeScript.



```
PS C:\Users\Govind\Desktop\webx> tsc studentdb.ts
PS C:\Users\Govind\Desktop\webx> node studentdb.js
Student Name: Sneha Patra
Average Marks: 44.33
Result: Passed
PS C:\Users\Govind\Desktop\webx>
```

The employee management system's output presented information for both managers and developers, with managers showing their assigned department and developers listing their programming languages. This output confirmed that the interface-based design successfully enforced a uniform structure, ensuring type safety and consistency across different employee roles.

Conclusion:

In this experiment, we successfully implemented a simple calculator using TypeScript with proper error handling for invalid operations and division by zero. Additionally, we developed a student result management system that calculates total marks, average marks, and evaluates the result based on the average score. The experiment demonstrated the effectiveness of TypeScript's type safety, object-oriented features, and error-handling capabilities.