

Experiment – 1 a: TypeScript

Name of Student	Spandan Deb
Class Roll No	13
D.O.P.	
D.O.S.	
Sign and Grade	

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

Overview of Tasks Performed:

Created a calculator function that performs basic arithmetic operations and handles errors for invalid operators and division by zero. Implemented a class-based system to manage students and their exam results, including adding students and results to a database, and retrieving results for specific students or all students.

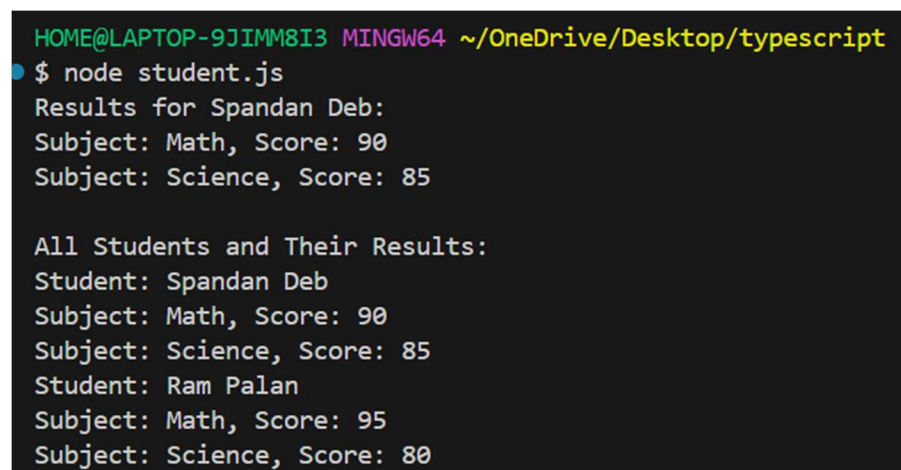
GitHub Link- <https://github.com/spandandeb/WebXEx1a>

a) Calculator

```
HOME@LAPTOP-9JIMM8I3 MINGW64 ~/OneDrive/Desktop/typescript
$ node calculator.js
12
8
20
5
Division by zero is not allowed!
```

This screenshot displays the output of the calculations like it displays `12` for `10+2`, `8` for `10-2`, etc. It also handles errors like division by zero and invalid operators. Whenever a user divides a number by `0` or types an invalid operator, they get a message in the output regarding the error.

b) Student Database system



```
HOME@LAPTOP-9JIMM8I3 MINGW64 ~/OneDrive/Desktop/typescript
$ node student.js
Results for Spandan Deb:
Subject: Math, Score: 90
Subject: Science, Score: 85

All Students and Their Results:
Student: Spandan Deb
Subject: Math, Score: 90
Subject: Science, Score: 85
Student: Ram Palan
Subject: Math, Score: 95
Subject: Science, Score: 80
```

This screenshot displays the output of the student database systems where a new student entries can be added via the db object created from class Database .Results for a specific student can be retrieved as well as the entire student entries with results can be retrieved.

Conclusion:

This experiment demonstrated the fundamental concepts of TypeScript, such as the use of basic data types (number, string, Boolean) and operators through the implementation of a calculator function and a student management system. The calculator function showcased arithmetic operations and error handling, while the student management system illustrated data organization and retrieval using classes. Together, these examples highlighted the practical application of TypeScript in performing computational tasks and managing data efficiently, reinforcing the language's capabilities in maintaining clear and functional code.