ES6 & TypeScript Assignments

1. **Constants:** Declare a constant & confirm its value cannot be changed.
2. **Scoping:** Declare a variable inside if condition & make sure that it is not accessible outside if condition.
3. **Enhanced object properties:** Create an ‘Order’ object having data members ‘id’, ‘title’, ‘price’. Add the methods printOrder() &getPrice(). Now, copy the order object using Object.assign().
4. **Arrow functions:** Take an array of strings & convert it into another array of object which has two properties {string, string\_length}. For example:

let names = [‘Tom’, ‘Ivan’, ‘Jerry’]

Output: [ {name: ’Tom’, length: 3}, {name: ’Ivan’, length: 4 }, {name: ’Jerry’, length: 5} ]

1. **Extended parameter handling:**
   1. Write a add() with default values.
   2. Write a function userFriends() that takes 2 arguments username & array of user friends. The function should print username & his list of friends. (Use rest parameters)
   3. Write a function printCapitalNames() that takes five names as argument & prints them in capital letters. Use spread operator in order to call printCapitalNames() function.
2. **Template literals:** Draft a ticket to Sysnet that describes problem with your laptop. Use ‘template literals’ to add value of laptop model, your desk no, your name etc.
3. **De-structuring assignment:**
   1. Suppose there is a javascript array with 4 elements. Print the value of 3rd element using array matching.
   2. Create an organization object having attributes name, address. Write a program to retrieve pin code of an address using object deep matching.
4. **Classes & Modules:** Write a class Account with attributes id, name, balance. Add two sub classes SavingAccount & CurrentAccount having specific attribute interest & cash\_credit respectively. Create multiple saving & current account objects. Write a functionality to find out total balance in the bank.
5. **Promises:** Create 2 promises, one generates value of x & another generates value of y. Write a program to print sum of x & y. (Use Promise.all)
6. **TypeScript classes & types:** Write a class Account with attributes id, name, balance. Add two sub classes SavingAccount & CurrentAccount having specific attribute interest & cash\_credit respectively. Create multiple saving & current account objects. Write a functionality to find out total balance in the bank.
7. **TypeScript Interfaces:** Write an interface Printable. Create 2 objects circle & employee those implement Printable interface. Write a function printAll() that takes all objects as argument & invoke print() method on every object.