TypeScript Assignment

1. **Promises:** Create 2 promises, one generates value of x & another generates value of you. Write a program to print sum of x & y. (Use Promise.all).

Ans:-

```
const sum = (a, b) => a + b;
const first = new Promise((resolve, reject) => {
const x = 5;
if (x) resolve(x);
else reject(x);
});
const second = new Promise((resolve, reject) => {
const y = 10;
if (y) resolve(y);
else reject(y);
});
const allpromises = Promise.all([first, second]);
allpromises.then(success => console.log('sum:', sum(success[0],
success[1])))
.catch(error => console.log('error:', error))
.finally(() => console.log("Executed finally block"));
```

> node "c:\Users\ASUS\Desktop\ES2015\Scripts\tempCodeRunnerFile.js"
sum: 15
Executed finally block

2. 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.

Ans:-

```
class Account {
    constructor(id, name, balance) {
        this.id = id;
        this.name = name;
        this.balance = balance;
    }
}
class SavingsAccount extends Account {
    constructor(id, name, balance, interest) {
        super(id, name, balance);
        this.interest = interest;
    }
```

```
    totalBalance() {
        let newBalance = this.balance * this.interest;
        this.balance = this.balance + newBalance;
        return this.balance
    }
}
class CurrentAccount extends Account {
    constructor(id, name, balance, cash_credit) {
        super(id, name, balance);
        this.cash_credit = cash_credit;
    }
    totalBalance() {
        let newBalance = this.balance * this.cash_credit;
        this.balance = this.balance + newBalance;
        return this.balance;
    }
}
var saving = new SavingsAccount(" 987612345128", "Bhushan", 200000, 1.5);
var current = new CurrentAccount(" 987612345129", "Nikhil", 500000, 0.5);
console.log(saving.totalBalance());
console.log(current.totalBalance());
```

> node "c:\Users\ASUS\Desktop\ES2015\Scripts\Classes & Types TS3(2).js"
500000
750000

3. 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.

Ans:-

```
interface Printable {
    name: string,
    printAll: (string) => string
}
var circle: Printable = {
    name: "Bhushan",
    printAll: (str) => { return "Hi" + str }
}
var employee: Printable = {
    name: " xyz",
    printAll: (str) => { return "Hello" + str }
}
console.log(circle.printAll(circle.name))
console.log(employee.printAll(employee.name))
```

> Hi Bhushan

Hello Ankush