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)

```
<!DOCTYPE html>
<html>
    <head>
        <title>demo</title>
    </head>
    <body>
        <h3>Promises</h3>
        <script src=Assignment3.ts></script>
        </script>
    </body>
</html>
```

```
var pro1 = new Promise(function(resolve,reject)
{
    if(1>0)
    resolve(x=15);
    else
    reject(4);
});
var pro2 = new Promise(function(resolve,reject)
{
    if(1>0)
    resolve(y =10);
    else
    reject(8);
});
Promise.all([pro1, pro2]).then(values =>{console.log(x + y);});
```

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.

```
<!DOCTYPE html>
<html>
    <head>
        <title>demo</title>
    </head>
    <body>
        <h3>TypeScript classes & types</h3>
        <script src=Assignment3.ts></script>
        </script>
```

```
    </body>
</html>
```

```
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 = () => { this.balance += this.interest;
        console.log(this.balance);};
}
class CurrentAccount extends Account{
    constructor(id,name,balance,cashCredit)
    {
        super(id,name,balance);
        this.cashCredit = cashCredit;
    }
    totalBalance = () => { this.balance += this.cashCredit;
        console.log(this.balance);};
}

let S1 = new SavingsAccount(101,'Rakshita',11,27);
let S2 = new CurrentAccount(103, 'Udita', 13,27);
S1.totalBalance();
S2.totalBalance();
```

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.

```
<!DOCTYPE html>
<html>
    <head>
        <title>demo</title>
    </head>
    <body>
        <h3>TypeScript Interfaces</h3>
        <script src=Assignment3.ts></script>
        </script>
```

```
    </body>
</html>
```

```
interface Printable{
    fname?:string;
    id?: number;
    area?:Number;
};

let circle :Printable ={
   area: 100,
};

let employee :Printable = {
    fname: "rakshita",
    id:101,
};

function printAll(circle, employee){
    console.log(circle);
    console.log(employee);
}

printAll(circle,employee);
```