

## EXERCISE3:

### Question1:

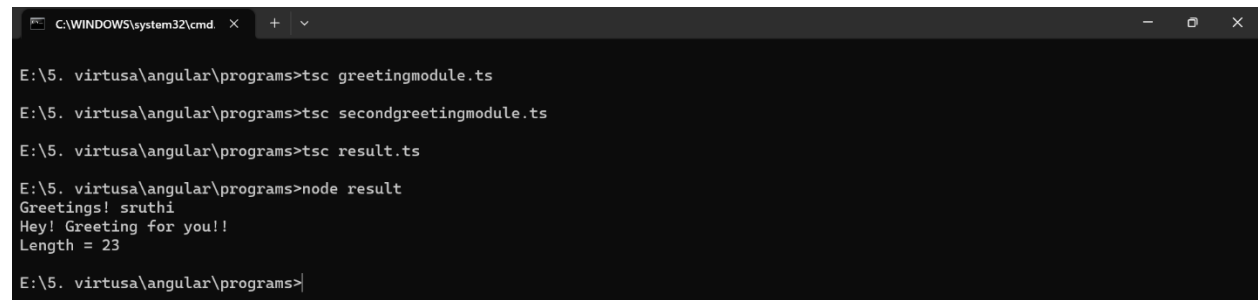
```
export function returnGreeting(greeting:string){
    console.log("Greetings! "+greeting);
}
```

### Question2:

```
export function returnGreeting():string{
    return "Hey! Greeting for you!!";
}
export function getLength(n:number):string{
    return "Length = "+n;
}
```

### Question3:

```
import * as g1 from "./greetingmodule";
import * as g2 from "./secondgreetingmodule";
g1.returnGreeting("sruthi");
console.log(g2.returnGreeting());
console.log(g2.getLength(23));
```



The screenshot shows a terminal window with the following commands and output:

```
C:\WINDOWS\system32\cmd. X + v
E:\5. virtusa\angular\programs>tsc greetingmodule.ts
E:\5. virtusa\angular\programs>tsc secondgreetingmodule.ts
E:\5. virtusa\angular\programs>tsc result.ts
E:\5. virtusa\angular\programs>node result
Greetings! sruthi
Hey! Greeting for you!!
Length = 23
E:\5. virtusa\angular\programs>
```

### Question4:

```
export namespace first{
    export class Train{
        trainID:number;
        trainName:string;
        speed:number;
        constructor(trainID:number,trainName:string,speed:number){
            this.trainID=trainID;
            this.trainName=trainName;
            this.speed=speed;
        }
        coverDistance(hour:number){
```

```

        console.log("The train "+this.trainName+" of id "+this.trainID+" has
covered a distance of "+hour*this.speed);
    }
}
export class Driver{
    driver(){
        console.log("Driver with Train");
    }
}
}

```

### Question5:

```

import {first} from "./firstNs";
let t=new first.Train(12078,"Jana Shatabdi",30);
let d=new first.Driver();
t.coverDistance(6);
d.driver();

```

```

C:\WINDOWS\system32\cmd. X + v
E:\5. virtusa\angular\programs>tsc firstNs.ts
E:\5. virtusa\angular\programs>tsc firstNs1.ts
E:\5. virtusa\angular\programs>node firstNs1
The train Jana Shatabdi of id 12078 has covered a distance of 180
Driver with Train
E:\5. virtusa\angular\programs>

```

### Question6:

```

namespace second{
    export class stadium{
        stadiumName:string;
        city:string;
        capacity:number;
        constructor(stadiumName:string,city:string,capacity:number){
            this.stadiumName=stadiumName;
            this.city=city;
            this.capacity=capacity;
        }
        matchResult(team1Score:number,team2Score:number){
            console.log("The match is in "+this.stadiumName+" stadium situated in
a city "+this.city+" with a capacity of "+this.capacity);
            if(team1Score>team2Score){
                console.log("Team A won the match!");
            }
            else if(team2Score>team1Score){
                console.log("Team B won the match!");
            }
        }
    }
}

```

```

    }
    else{
        console.log("Both Teams with simillar score");
    }
}
}
}
}

```

### Question7:

```

///<reference path="./secondNs.ts"/>
let s=new second.stadium("ntr","guntur",200);
s.matchResult(30,50);

```

```

C:\WINDOWS\system32\cmd. x + v

E:\5. virtusa\angular\programs>tsc secondNs.ts

E:\5. virtusa\angular\programs>tsc -target es6 secondNs1.ts --outfile secondns.js

E:\5. virtusa\angular\programs>node secondns
The match is in ntr stadium situated in a city guntur with a capacity of 200
Team B won the match!

E:\5. virtusa\angular\programs>

```

### Question8:

```

interface ProcessIdentity<T,U>{
    value:T;
    message:U;
    property():T;
}

```

### Question9:

```

interface ProcessIdentity<T,U>{
    value:T;
    message:U;
    property():T;
}
class callIdentity<X,Y> implements ProcessIdentity<X,Y>{
    value:X;
    message:Y;
    constructor(value:X,message:Y){
        this.value=value;
        this.message=message;
    }
    property():X{
        return this.value;
    }
    message_return():Y{

```

```
        return this.message;
    }
}
let g=new callIdentity<number,string>(6,"sruthi");
console.log("The message is: "+g.message_return());
console.log("The value is: "+g.property());
```

JS .D.TS Errors Logs Plugins

[LOG]: "The message is: sruthi"

[LOG]: "The value is: 6"