1)Function with parameters - with return types

Using rest parameters

```
function add(...nums:number[]) : number{
var i:number;
var sum:number=0;
        for(i=0;i<nums.length;i++){
        sum+=nums[i]
        }
        //console.log("Total sum is :",sum);
        return sum;
}
var total:number=add(2,3,4);
console.log("Total sum is :",total);
```

2)Function with parameters - without return types

Using optional parameters:

```
function details(name :string,rollNo: number,age?:number) {
    console.log("Name is:",name);
    console.log("roll number is :",rollNo);
    if(age!=undefined)
      console.log("Age is",age);

}
console.log("Details of first student :");
details("vamsi",100,23);
console.log("Details of second student :");
details("harshit",101);
```

3)Function without parameters - without return types

```
function details() {
   console.log("sample type script program using functions :")

}
details();
```

4)Function without parameters - with return types

```
function details() : string {
```

```
    return "sample type script program using functions :";

}
var example : string = details();
console.log(example);
```

5)Function with parameters - with return types using anonymous classes

```
var example  = function(a :number) {
    return "sample type script program using functions :"+a;

};
console.log(example(1));
```

6) Create a function to check if a given year is a leap year or not that returns string “Leap Year” or “Not a Leap Year”

```
var leapYear  = function(year :number) {
   year=year%400;
   if(year%4==0){
   return "Year is leap year";
   }
   else{
   return "year is not leap year :";
   }

};
console.log(leapYear(2002));
```

Arrow Functons :

7)Function with parameters - with return types

```
var total=(...nums:number[])=>{
var i:number;
var sum:number=0;
        for(i=0;i<nums.length;i++){
        sum+=nums[i]
        }
        return sum;

}
console.log(total(2,3,4));
```

8) Function with parameters - without return types

```
var details=(name :string,rollNo: number,age?:number)=>{
         console.log("Name is:",name);
```

```
        console.log("roll number is :",rollNo);
        if(age!=undefined)
           console.log("Age is",age);
}
console.log("Details of first student :");
details("vamsi",100,23);
console.log("Details of second student :");
details("harshit",101);
```

9) Function without parameters - without return types

```
var details=()=>{
            console.log( "sample type script program using functions :");
}
details();
```

10)
Function without parameters - with return types

```
var details=()=>{
            return "sample type script program using functions :";
}
console.log(details());
```

11)- Define some arrays and iterate using different loops.

```
var num : number[]=[1,2,3,4,5];
console.log("Using for loop :");
for(let i=0;i<num.length;i++){
            console.log(num[i]);
}
console.log("Using for in  loop:");
for(let i in num){
            console.log(i);
}
console.log("Using for of  loop:");
for(let i of num){
            console.log(i);
}
console.log("Using for Each loop :");
num.forEach((value)=>{
            console.log(value);
})
```

12)- Define an empty string array and then add values and remove values from the array.

```
var str : string[]=new Array();
str.push("a");
```

```
console.log("After pushing one element into array :")
console.log(str);
str.pop();
console.log("After removing last element from array :");
console.log(str);
```

13)List different types in typescript and their declarations

```
let sample : boolean=true;
console.log ("boolean value is : " + sample);
let str: String = "hello"
console.log ("string : " + str);
let num: number[]=[3,4,5];
console.log ("number array is : " + num);
let arr2:Array<String>=["vamsi","Guggilam"];
console.log ("string array: " + arr2);
let variable: number = 5
console.log ("number is : " + variable);
```

14)