1. Declare a variable age of type number and assign it a value. Print it.

```
let age: number = 25;
console.log(age);
```

2. Create a variable username of type string and log "Hello, <username>".

```
let username: string = "Raksha";
console.log("Hello, " + username);
```

3. Declare a boolean variable isActive and assign it true. Print its type.

```
let isActive: boolean = true;
console.log(typeof isActive);
```

4. Create two number variables x and y, assign values, and print their sum.

```
let x: number = 10;
let y: number = 20;
console.log(x + y);
```

5. Declare a variable colors as an array of strings with three colors. Print the second one.

```
let colors: string[] = ["red", "blue", "green"];
console.log(colors[1]);
```

6. Create a constant PI with value 3.14 and try reassigning it (observe the error).

```
const PI: number = 3.14;
```

7. Write a function greet that takes a name (string) and returns "Hello, <name>".

```
function greet(name: string): string {
   return "Hello, " + name;
}
console.log(greet("Rahul"));
```

8. Write a function addNumbers that takes two numbers and returns their sum.

```
function addNumbers(a: number, b: number): number {
   return a + b;
}
console.log(addNumbers(5, 7));
```

9. Create a function isEven that takes a number and returns true if it’s even, else false.

```
function isEven(num: number): boolean {
   return num % 2 === 0;
}
console.log(isEven(8));
```

10. Write a function multiply with default parameter b = 5 that multiplies a * b.

```
function multiply(a: number, b: number = 5): number {
   return a * b;
}
console.log(multiply(4));
```

11. Create an arrow function square that takes a number and returns its square.

```
const square = (n: number): number => n * n;

console.log(square(6));
```

12. Write a function printDetails that accepts a name (string) and age (number) and prints:

"Name: <name>, Age: <age>".

```
function printDetails(name: string, age: number): void {
   console.log(`Name: ${name}, Age: ${age}`);
}
printDetails("Neha", 22);
```

13. Create a class Person with name and age properties, and a method introduce() that logs "Hi, I'm <name> and I'm <age> years old.".

```
class Person {
  name: string;
  age: number;
  constructor(name: string, age: number) {
    this.name = name;
    this.age = age;
  }
  introduce(): void {
    console.log(`Hi, I'm ${this.name} and I'm ${this.age} years old.`);
  }
}
const person1 = new Person("Alice", 25);
person1.introduce();
```

14. Add a constructor to Person that initializes name and age.

```
class Person {
  name: string;
  age: number;
  constructor(name: string, age: number) {
    this.name = name;
    this.age = age;
  }
  introduce(): void {
    console.log(`Hi, I'm ${this.name} and I'm ${this.age} years old.`);
  }
}
const person1 = new Person("Alice", 25);
person1.introduce();
```

15. Create a class Car with properties brand and year, and a method displayInfo() that logs "Car: <brand>, Year: <year>".

```
class Car {
    brand: string;
    year: number;
    constructor(brand: string, year: number) {
        this.brand = brand;
        this.year = year;
    }
    displayInfo(): void {
```

```
        console.log(`Car: ${this.brand}, Year: ${this.year}`);
    }
}
let car1 = new Car("Toyota", 2022);
car1.displayInfo();
```

16. Create a class Rectangle with properties width and height and a method getArea() that returns area.

```
class Rectangle {
    width: number;
    height: number;
    constructor(width: number, height: number) {
        this.width = width;
        this.height = height;
    }
    getArea(): number {
        return this.width * this.height;
    }
}
let rect1 = new Rectangle(5, 10);
console.log(rect1.getArea());
```

17. Create a class Student that has name and grade, and a method displayGrade() that logs "Student <name> has grade <grade>".

```
class Student {
    name: string;
```

```
    grade: string;
    constructor(name: string, grade: string) {
        this.name = name;
        this.grade = grade;
    }
    displayGrade(): void {
        console.log(`Student ${this.name} has grade ${this.grade}`);
    }
}
let s1 = new Student("Priya", "A");
s1.displayGrade();
```

18. Create a class BankAccount with accountNumber and balance, and a method deposit(amount) that adds to balance and logs the new balance.

```
class BankAccount {
    accountNumber: number;
    balance: number;
    constructor(accountNumber: number, balance: number) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }
    deposit(amount: number): void {
        this.balance += amount;
        console.log(`New Balance: ${this.balance}`);
    }
}
let acc1 = new BankAccount(12345, 5000);
acc1.deposit(2000);
```