Marvellous Infosystems: Angular Assignment No: 3

Name: Shrirang Jagdish Nikam

Enrollment No: 396AM_Shrirang

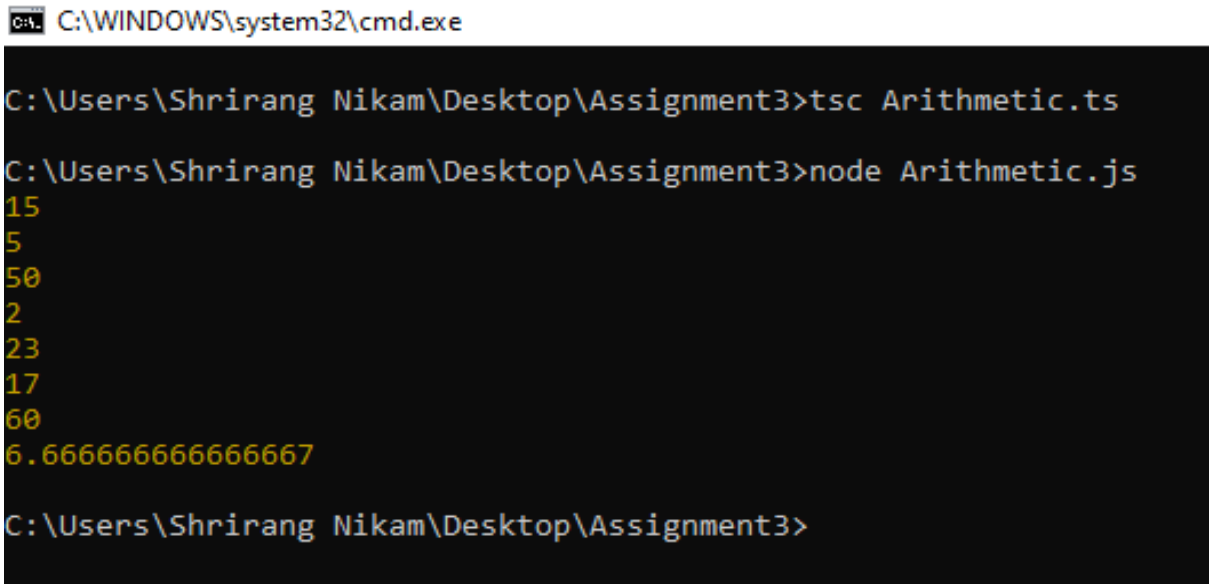
1. **Create one typescript application which contains one class named as Arithmetic. Arithmetic class contains three characteristics (Class data members) as Number1, Number2. Create one parametrised constructor which accept two values and assign it to Number1 and Number2.**
   In Arithmetic class we have to write four methods (Behaviours) as Addition, Subtraction, Multiplication and Division.
   Addition method will add Number1, Number2 & return result.
   Subtraction method will subtract Number1, Number2 & return result.
   Multiplication method will multiply Number1, Number2 & return result.
   Division method will divide Number1, Number2 & return result.
   After designing the class create two objects of that class by providing some hardcoded value.
   Call all the methods by using both the objects.

Answer:

```
class Arithmetic {

private number1: number;
private number2: number;

constructor(number1: number, number2: number) {
this.number1 = number1;
this.number2 = number2;
}

public addition(): number {
return this.number1 + this.number2;
}

public subtraction(): number {
return this.number1 - this.number2;
```

```
}

public multiplication(): number {
return this.number1 * this.number2;
}

public division(): number {
return this.number1 / this.number2;
}
}
const arithmeticObj1 = new Arithmetic(10, 5);
const arithmeticObj2 = new Arithmetic(20, 3);
console.log(arithmeticObj1.addition());
console.log(arithmeticObj1.subtraction());
console.log(arithmeticObj1.multiplication());
console.log(arithmeticObj1.division());
console.log(arithmeticObj2.addition());
console.log(arithmeticObj2.subtraction());
console.log(arithmeticObj2.multiplication());
console.log(arithmeticObj2.division());
```

Output:

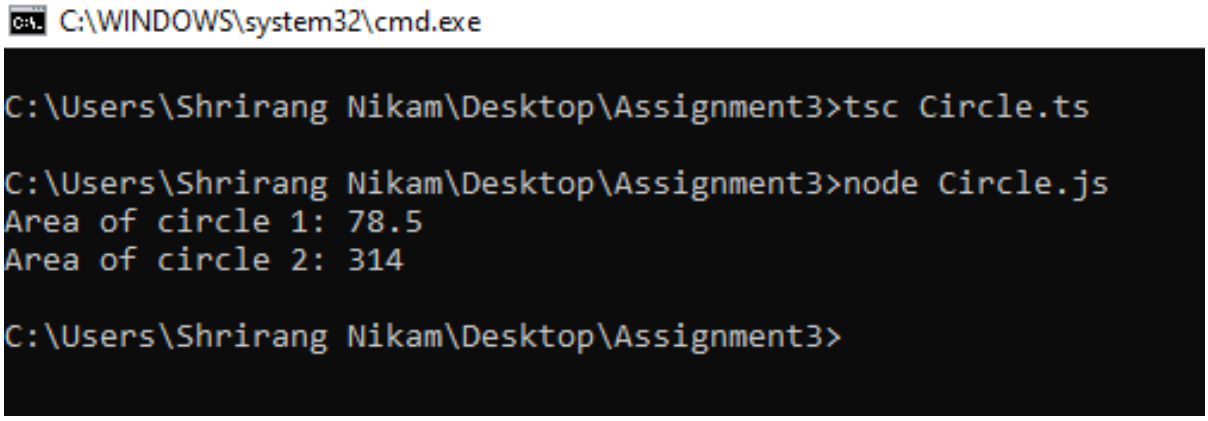


2. **Create one typescript application which contains one class named as Circle.**
   Circle class contains two characteristics (Class data members) as Radius, PI. Create one parametrised constructor which accept one value and assign it to Radius. Value of PI member is set to 3.14.
   In circle class we have to one method (Behaviours) as Area which will return area of Circle. After designing the class create two objects of that class by providing some hardcoded value. Call the method Area by using both the objects.

Answer:

```
class Circle {

radius: number;
readonly PI: number = 3.14;

constructor(radius: number) {
this.radius = radius;
}

area(): number {
return this.PI * this.radius * this.radius;
}
}
const circle1 = new Circle(5);
const circle2 = new Circle(10);
console.log(`Area of circle 1: ${circle1.area()}`);
console.log(`Area of circle 2: ${circle2.area()}`);
```

Output:



3. **Create one typescript application which contains one class named as Circle X which sill inherits above Circle class.**
   In Circle X class we have to write one method (Behaviours) as Circumference which will return circumference of circle.
   After designing the class create two objects of that class by providing some hardcoded value. Call circumference and Area methods by using both the objects.

   Answer:

```
class Circle {
  private readonly radius: number;

  constructor(radius: number) {
    this.radius = radius;
  }

  getArea(): number {
    return Math.PI * this.radius * this.radius;
  }

  getRadius(): number {
    return this.radius;
  }
}

class CircleX extends Circle {
  constructor(radius: number) {
    super(radius);
```

```
  }

  getCircumference(): number {
    return 2 * Math.PI * this.getRadius();
  }
}
const circle1 = new CircleX(5);
const circle2 = new CircleX(10);
console.log(`Circle 1 area: ${circle1.getArea()}`);
console.log(`Circle 1 circumference: ${circle1.getCircumference()}`);
console.log(`Circle 2 area: ${circle2.getArea()}`);
console.log(`Circle 2 circumference: ${circle2.getCircumference()}`);
```

Output: