1. Which members of the Circle class are encapsulated?

- **Encapsulation**: Hides data and provides controlled access via methods like getters and setters.
- If `radius` is private, it is encapsulated. If public, it is not.

2. What name must the constructor of a class have?

- A constructor **must have the same name as the class** and no return type.

Example:
```
public class Circle {
    public Circle() {
        // Constructor code
    }
}
```

-

3. Difference between private and public access modifiers?

Private: Accessible only within the same class.
Example:
```
private int radius;
```

-

Public: Accessible from any other class.
Example:
```
public int radius;
```

-

4. Validity of `Circle dot = new Circle(2); dot.radius = 5;`

- If `radius` is public, the statement is valid.

If private, use a setter method:
dot.setRadius(5);

5. Issues in the Roo class code

a) Code Issues

1. **Constructor Syntax**: Missing parentheses.

Fix:
public Roo() {
  x = 1;
}

     ○

2. **Return Type Mismatch in `factor()`**: Return type mismatch between `double` and `int`.
     ○ Fix by:

Changing the return type to `double`:
private double factor() {
  return 0.12;
}

       ■

Casting to `int`:
private int factor() {
  return (int) 0.12;
}

       ■

b) Explanation of Methods

- `setX(int z)`: Setter for `x`.
- `getX()`: Getter for `x`.
- `calculate()`: Modifies `x` using `factor()`. Throws error if `factor()` type mismatch isn't fixed.
- `factor()`: Returns a constant value (0.12), with a return type mismatch.

7. Class for a Sports Team

a) Object Names:

- Examples: `Lions`, `Tigers`, `Eagles`.

b) Method Members:

- `playGame()`, `practice()`, `addPlayer()`.

c) Data Members:

- `teamName`, `coach`, `players`.

8. Band Festival Simulation

a) Class Name:

- `Band` could represent common properties and methods for all bands.

b) Objects:

- Instances like `TwoToos`, `EggRolls`, `Goop`.

c) Data Member Examples:

- `name`: The band's name.
- `genre`: The type of music they play.

d) Method Members:

- `tuneUp()`: Simulates the band tuning their instruments.
- `playMusic()`: Simulates the band performing.
- `takeABow()`: Simulates the end of their performance.