S.No: 6	Exp. Name: ***Dynamic method dispatch***	Date:2025-09-16

Aim:

You are tasked with creating a Java program to demonstrate the concept of runtime polymorphism using method overriding. The program should consist of the following classes:

1. Rectangle2 (Parent Class):
- Contains a method named **area()** to calculate and display the area of a rectangle.
- Takes input for the length and width of the rectangle from the user.

2. Square (Child Class of Rectangle2):
- Overrides the **area()** method to calculate and display the area of a square.
- Takes input for the side length of the square from the user.

3. Triangle2 (Child Class of Rectangle2):
- Overrides the **area()** method to calculate and display the area of a triangle.
- Takes input for the base and height of the triangle from the user.

4. Calculation (Main Class):
- Contains the **main** method, where objects of the **Rectangle2, Square**, and **Triangle2** classes are created.
- Demonstrates **runtime polymorphism** by using a parent class reference (**Rectangle2**) to invoke the **area()** method on objects of different classes.

Input Format:

1. The program first prompts the user to enter the length and width of a rectangle:
   - The first integer represents the length of the rectangle.
   - The second integer represents the width of the rectangle.
2. Next, the program prompts the user to enter the side length of a square:
   - The integer represents the side of the square.
3. The program then prompts the user to enter the base and height of a triangle.:
   - The first integer represents the base of the triangle.
   - The second integer represents the height of the triangle.

Output Format:

1. The program prints the area of the rectangle in the format: **Area of rectangle: <calculated_area>**
2. The program prints the area of the square in the format: **Area of square: <calculated_area>**
3. The program prints the area of the triangle in the format: **Area of triangle: <calculated_area>**

Note:

1. Refer to the sample test cases for a better understanding of the input-output format.
2. Implement the missing method bodies as per the given requirements.
3. The **print statements** for prompting user inputs have already been provided.

Source Code:

Calculation.java

```
import java.util.Scanner;
class Rectangle2 {

// Method to calculate the area of a rectangle
public void area(){
```

```
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter length and width of Rectangle:");
    int length = sc.nextInt();
    int width = sc.nextInt();

    int area = length * width;
    System.out.println("Area of rectangle:" + area);
    }
}
class Square extends Rectangle2 {
public void area(){
    // Overridden method for square area calculation
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter side of square:");
    int side = sc.nextInt();
    int area = side*side;
    System.out.println("Area of square:"+ area);
    }
}
class Triangle2 extends Rectangle2 {
    public void area(){
       // Overridden method for triangle area calculation
       Scanner sc = new Scanner(System.in);
       System.out.println("Enter base and height of Traingle:");
       int base = sc.nextInt();
       int height = sc.nextInt();
       int area = (base * height)/2;
       System.out.println("Area of triangle:"+area);
     }
}
class Calculation {
     public static void main(String args[]) {

         Rectangle2 r = new Rectangle2();
         r.area();

         r = new Square();
         r.area();

         r = new Triangle2();
         r.area();
     }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter length and width of Rectangle: 10
7
Area of rectangle:70 8
Enter side of square: 8
Area of square:64 15
Enter base and height of Traingle: 15

9
Area of triangle:67

Test Case - 2
User Output
Enter length and width of Rectangle: 10
15
Area of rectangle:150 30
Enter side of square: 30
Area of square:900 32
Enter base and height of Traingle: 32
21
Area of triangle:336