

Java Program: MathPractice.java

This Java program demonstrates five mathematical equations using user input and standard Java Math functions.

```
// FIXED: The Scanner class is in the
'java.util' package, not 'java.lang' import
java.util.Scanner;

public class MathPractice {
    public static void
main(String[] args)
{
    Scanner sc = new
Scanner(System.in);

// Equation 1: Right Triangle Height

        System.out.println("Equation 1:
Calculate the height of a right triangle.");
        System.out.print("Enter base (b) : ");
        double b = sc.nextDouble();
        System.out.print("Enter angle (theta in degrees) :
");        double theta = sc.nextDouble();
        double height = b *
Math.tan(Math.toRadians(theta));
        System.out.println("Height: " + height);
```

Sample Program Output:

--- Sample Execution Output ---

```
Equation 1: Calculate the height of a right
triangle.
Enter base (b): 5
Enter angle (theta in degrees): 30
Height: 2.886751345948129
```

// Equation 2: Compound Interest

```
System.out.println("\nEquation 2:  
Compound Interest Calculation.");  
System.out.print("Enter Principal (P): ");  
double P = sc.nextDouble();  
System.out.print("Enter Annual Interest  
Rate (r as a decimal): "); double  
r = sc.nextDouble();  
System.out.print("Enter Number of  
Compounds per Year (n): "); int n  
= sc.nextInt();  
System.out.print("Enter Time in Years (t): ");  
double t = sc.nextDouble();  
double A = P * Math.pow(1 + r / n,  
n * t); System.out.println("Total  
Amount: " + A);
```

Sample Program Output:

```
Equation 2: Compound Interest Calculation.  
Enter Principal (P): 10000  
Enter Annual Interest Rate (r as a decimal):  
0.05  
Enter Number of Compounds per Year (n): 4  
Enter Time in Years (t): 5  
Total Amount: 12833.590
```

```
// Equation 3: Cartesian to Polar

        System.out.println("\nEquation 3:
Convert Cartesian to Polar Coordinates.");
        System.out.print("Enter x: ");           double x
= sc.nextDouble();
        System.out.print("Enter y: ");           double y
= sc.nextDouble();           double rPolar =
Math.sqrt(Math.pow(x, 2) + Math.pow(y, 2));
        double thetaPolar = Math.toDegrees(Math.atan(y /
x));           System.out.println("Radius: " +
rPolar + ", Angle: " + thetaPolar + "°");
```

Sample Program Output:

```
Equation 3: Convert Cartesian to Polar
Coordinates.
Enter x: 3
Enter y: 4
Radius: 5.0, Angle: 53.1301°
```

```
// Equation 4: Distance Between Two Points

        System.out.println("\nEquation 4:
Calculate Distance Between Two Points.");
        System.out.print("Enter x1: ");           double
x1 = sc.nextDouble();
        System.out.print("Enter y1: ");           double
y1 = sc.nextDouble();
        System.out.print("Enter x2: ");           double
x2 = sc.nextDouble();
        System.out.print("Enter y2: ");           double
y2 = sc.nextDouble();
        double distance = Math.sqrt(Math.pow(x2
- x1, 2) + Math.pow(y2 - y1, 2));
        System.out.println("Distance: " + distance);
```

Sample Program Output:

```
Equation 4: Calculate Distance Between Two
Points.
Enter x1: 2
Enter y1: 3
Enter x2: 7
Enter y2: 8
Distance: 7.0710678118654755
```

// Equation 5: Quadratic Equation

```
System.out.println("\nEquation 5: Solve
Quadratic Equation.");
System.out.print("Enter coefficient a: ");
double a = sc.nextDouble();
System.out.print("Enter coefficient b: ");
double bQuad = sc.nextDouble();
System.out.print("Enter coefficient c: ");
double c = sc.nextDouble();           double
discriminant = Math.pow(bQuad, 2) - 4 * a * c;
if (discriminant >= 0) {             double root1
= (-bQuad + Math.sqrt(discriminant)) / (2 * a);
double root2 = (-bQuad - Math.sqrt(discriminant))
/ (2 * a);           System.out.println("Roots:
" + root1 + ", " + root2);           if
(root1 >= 0 && root2 >= 0)
{
    System.out.println("Smallest
positive root: " + Math.min(root1, root2));
} else if (root1 >= 0) {
    System.out.println("Smallest
positive root: " + root1);
} else if (root2 >= 0) {
    System.out.println("Smallest
positive root: " + root2);
} else {
    System.out.println("No positive
roots.");}
}

System.out.println("No real
roots.");}
```

```
sc.close();  
}  
}
```

Sample Program Output:

--- Sample Execution Output ---

```
Equation 5: Solve Quadratic Equation.  
Enter coefficient a: 1  
Enter coefficient b: -3  
Enter coefficient c: 2  
Roots: 2.0, 1.0  
Smallest positive root: 1.0
```