□Right Triangle Height:

```
import java.util.Scanner;
public class MathPractise{
public static void main(string[] args){
System.out.println("Calculate the height of a
                              triangle");
Scanner input = new scanner(System.in);
double base;
System.out.println(" Enter the value of base: ");
base = input.nextDouble();
System.out.println(" Enter angle(theta in degree): ");
double theta;
theta = input.nextDouble();
double height = base * Math.tan(Math.toRadians(theta)
System.out.println(" The Ultimate height is: " +
                              height);
    }
}
```

Compound Interest Calculation:

```
import java.util.Scanner;
public class InterestCalculation{
public static void main(string[] args){
 scanner input= new Scanner(system.in);

System.out.println("Enter principal P: ");
double p;
p= input.nextDouble();
System.out.println(" Enter Annual Interest Rate
                        (r as a decimal): ");
double r;
r=input.nextDouble();
System.out.println(" Enter Number of compounds
                        per year(n): ");
int n;
n=input.nextInt();
System.out.println("Enter time in Years(t):");
int t;
t = input.nextInt();
double A = p* Math.pow(1+r/n, n*t);
System.out.println(" Total Amount: ",+A);
  }
}
```

Convert Cartesian to Polar Coordinates:

```
import java.util.Scanner;
public class Convert{
public static void main(String[] args){
  Scanner input= new Scanner(System.in);
System.out.print(" Enter the value of x: ");
double x;
x= input.nextDouble();
System.out.println(" Enter the value of y: ");
double y;
y=input.nextDouble();
double r = Math.sqrt(Math.pow(x,2)+Math.pow(y,2));
double theta= Math.todegrees(Math.atan2(y,x));
System.out.println(" r is: "+r);
System.out.println(" theta is: "+ theta);
  }
}
```

Calculate distance between two points:

```
import java.util. Scanner;
public class distance{
public static void main(String [] args){
  Scanner input = new Scanner(System.in);
System.out.print("Enter the value of x1: ");
double x1;
x1 = input.nextDouble();
System.out.print("Enter the value of y1: ");
double y1;
y1 = input.nextDouble();
System.out.print("Enter the value of x2: ");
double ~~y~~ x2;
x2 = input.nextDouble();
System.out.print("Enter the value of y2: ");
double y2;
y2 = input.nextDouble;
double d = Math.sqrt(Math.pow(x2-x1,2) +
                     Math.pow(y2-y1,2));
System.out.print("Distance: "+d);
   }
}
```

Find the smallest positive Root of a Quardratic Equation

```
import java.util.Scanner;
public class QuardraticSolver{
public static void main(String[] args){
 Scanner input = new Scanner(System.in);
System.out.print(" Enter co-efficient a: ");
double a = input.nextDouble();

System.out.print(" Enter co-efficient b: ");
double b= input.nextDouble();

System.out.print(" Enter co-efficient c: ");
double c = input.nextDouble();

double discriminant = Math.pow(b,2)-4*a*c;

if (discriminant >=0){
double root1 = -b+Math.sqrt(discriminant))/(2*a);
double root2 = -b-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 ");
    }
  }
   else {
   System.out.println ("No real roots");
  }
  input.close ();
    }
 }
```