

```
public static void computeStats (int [ ] numbers)
{
    int length = numbers.length;
    double med, var, sd, mean, sum, varsum;

    sum = 0;
    for (int i = 0; i < length; i++)
    {
        sum += numbers [ i ];
    }
    med    = numbers [ length / 2];
    mean = sum / (double) length;

    varsum = 0;
    for (int i = 0; i < length; i++)
    {
        varsum = varsum + ((numbers[i]- mean) * (numbers[i] - mean));
    }
    var = varsum / ( length - 1.0 );
    sd  = Math.sqrt ( var );

    System.out.println ("length:           " + length);
    System.out.println ("mean:           " + mean);
    System.out.println ("median:         " + med);
    System.out.println ("variance:       " + var);
    System.out.println ("standard deviation: " + sd);
}
```

```
// Introduction to Software Testing
// Authors: Paul Ammann & Jeff Offutt
// Chapter 3, section 3.3, page 121

// Jeff Offutt--Java version Feb 2003
// Classify triangles
import java.io.*;

class trityp
{
    private static String[] triTypes = { "", // Ignore 0.
        "scalene", "isosceles", "equilateral", "not a valid triangle"};
    private static String instructions = "This is the ancient TriTyp program.\nEnter
three integers that represent the lengths of the sides of a triangle.\nThe triangle
will be categorized as either scalene, isosceles, equilateral\nor invalid.\n";

public static void main (String[] argv)
{ // Driver program for trityp
    int A, B, C;
    int T;

    System.out.println (instructions);
    System.out.println ("Enter side 1: ");
    A = getN();
    System.out.println ("Enter side 2: ");
    B = getN();
    System.out.println ("Enter side 3: ");
    C = getN();
    T = Triang (A, B, C);

    System.out.println ("Result is: " + triTypes[T]);
}

// =====
// Read (or choose) an integer
private static int getN ()
{
    int inputInt = 1;
    BufferedReader in = new BufferedReader (new InputStreamReader (System.in));
    String inStr;

    try
    {
        inStr    = in.readLine ();
        inputInt = Integer.parseInt(inStr);
    }
    catch (IOException e)
    {
        System.out.println ("Could not read input, choosing 1.");
    }
    catch (NumberFormatException e)
    {
        System.out.println ("Entry must be a number, choosing 1.");
    }

    return (inputInt);
} // end getN

} // end trityp class
```

```
// =====  
// The main triangle classification method  
private static int Triang (int Side1, int Side2, int Side3)  
{  
    int triOut;  
  
    // triOut is output from the routine:  
    //   Triang = 1 if triangle is scalene  
    //   Triang = 2 if triangle is isosceles  
    //   Triang = 3 if triangle is equilateral  
    //   Triang = 4 if not a triangle  
  
    // After a quick confirmation that it's a valid  
    // triangle, detect any sides of equal length  
    if (Side1 <= 0 || Side2 <= 0 || Side3 <= 0)  
    {  
        triOut = 4;  
        return (triOut);  
    }  
  
    triOut = 0;  
    if (Side1 == Side2)  
        triOut = triOut + 1;  
    if (Side1 == Side3)  
        triOut = triOut + 2;  
    if (Side2 == Side3)  
        triOut = triOut + 3;  
    if (triOut == 0)  
    { // Confirm it's a valid triangle before declaring  
        // it to be scalene  
  
        if (Side1+Side2 <= Side3 || Side2+Side3 <= Side1 ||  
            Side1+Side3 <= Side2)  
            triOut = 4;  
        else  
            triOut = 1;  
        return (triOut);  
    }  
  
    // Confirm it's a valid triangle before declaring  
    // it to be isosceles or equilateral  
  
    if (triOut > 3)  
        triOut = 3;  
    else if (triOut == 1 && Side1+Side2 > Side3)  
        triOut = 2;  
    else if (triOut == 2 && Side1+Side3 > Side2)  
        triOut = 2;  
    else if (triOut == 3 && Side2+Side3 > Side1)  
        triOut = 2;  
    else  
        triOut = 4;  
    return (triOut);  
} // end Triang
```