```java
import CSCI.*;
import java.util.*;
public class MathPlay
{
    public final static int ERROR = 0; //error constant is declared
    public static void main (String[] args)
    {
        String filename = args[0];
        double average;
        ArrayList<String> input = Reader(filename);
        int[] numbers = decode(input);
        int size = numbers.length;
        average = getAverage(numbers); //call various self explanatory methods for respective calculations
        int max = getMax(numbers);
        int min = getMin(numbers);
        int median = getMedian(numbers);
        System.out.println("Size: "+size+" Average: "+average+" Max: "+max+" Min: "+min+" Median: "+median);
    } //end main

    public static ArrayList<String> Reader(String filename)
    {
       FileIn myFile = new FileIn(filename);
       ArrayList<String> input = new ArrayList<String>();
       String line; //primer read
       line = myFile.Read();
       while (line != null) //while loop until end of file is reached
          {
            input.add(line); //place data into arrayList using add
            line = myFile.Read(); //read next line
          }
       myFile.close(); //close file
       return input; //return arraylist
    }

    public static int[] decode(ArrayList<String> input)
    {
        int size = input.size(); //set size to length of data in input file
        int[] numbers = new int[size]; //create new numbers array of size equal to input file
        String line;
        for(int i = 0; i<size; ++i) //for length of this array
        {
            line = input.get(i); //add value from input file to array at each index
            numbers[i] = CSCIConvert.Parse(line,ERROR); //parsed to ensure it is an int as it does this
        }
        return numbers; //return the final completed array
    }

    public static double getAverage(int[] numbers)
    {
        int size = numbers.length; //set size to size of numbers arrayList
        double sum = 0; //declare sum at zero first
        if(0 == size) return 0; //no dividing by zero
        for(int i=0; i<size; ++i)
        {
            sum = sum + numbers[i]; //sum all data points for length of numbers arrayList
        }
        double average = sum/size; //calculate average
        return average; //return the calculated average double
    }

    public static int getMax(int[] numbers)
    {
        int size = numbers.length; //set size to length of array
        /*if (0 == size) return 0;
        int max = numbers[0];
        for(int i=0; i<size; ++i)
        {
            if(max < numbers[i]) max = numbers[i];
        }
        return max;*///commented out this code because arraysort does it better
        Arrays.sort(numbers); //sort array
        return numbers[size-1]; //return max value
    }

    public static int getMin(int[] numbers)
    {
        int size = numbers.length; //set size to length of array
        Arrays.sort(numbers); //sort array
        return numbers[0]; //return first value in array (min)
    }

    public static int getMedian(int[] numbers)
    {
        Arrays.sort(numbers); //sort array
        int size = numbers.length; //set size to length of array
        if (0 == size) return 0; //avoid division by zero
        int median = numbers[0]; //default median to first value
        if(evenCheck(size)) //check if array has even number of data values
        {
            median = (numbers[size/2] + numbers[(size/2)-1])/2; //formula for even
            number of data values
        }
        else
        {
            median = numbers[(size-1)/2]; //formula for odd number of data values
        }
        return median; //return median value (middle value) of array
    }

    public static boolean evenCheck(int value) //check if integer value is even
    {
        if((value%2)==0) return true; //if even, return true (can divide by zero with
        no remainder)
        else return false;
    }
}
```