

Homework 3

Assigned Laboratory 4

Due start of Laboratory 5

This is a pledged assignment; you may receive help only from the class teaching assistants, ITC consultants, and the class professors. You may work in groups of up to two for this assignment. Members of a group must come from the same lab section.

Assignments Files

A self-extracting archive `hw03.exe` has been created for this assignment. The archive will produce two files `hw03.ide` and `hw03.cpp`. Archive `hw03.exe` can be acquired by accessing the course website <http://www.cs.virginia.edu/cs101>. If you are copying this file using a web browser, it is recommended that you first copy the file to a hard drive and then, if necessary, copy the file to a floppy. Our browsers seem to miscopy when downloading binary files directly to a floppy.

Objectives

Practice using a graphical application programmer interface. Practice processing data sets of unknown size using a looping construct. Practice rudimentary fault tolerance by modifying errant data set values. These practices are the norm in modern computing environments and software engineering.

Problem

Suppose we are given the following data set that represents 15 observations of some phenomenon. What statements can we make?

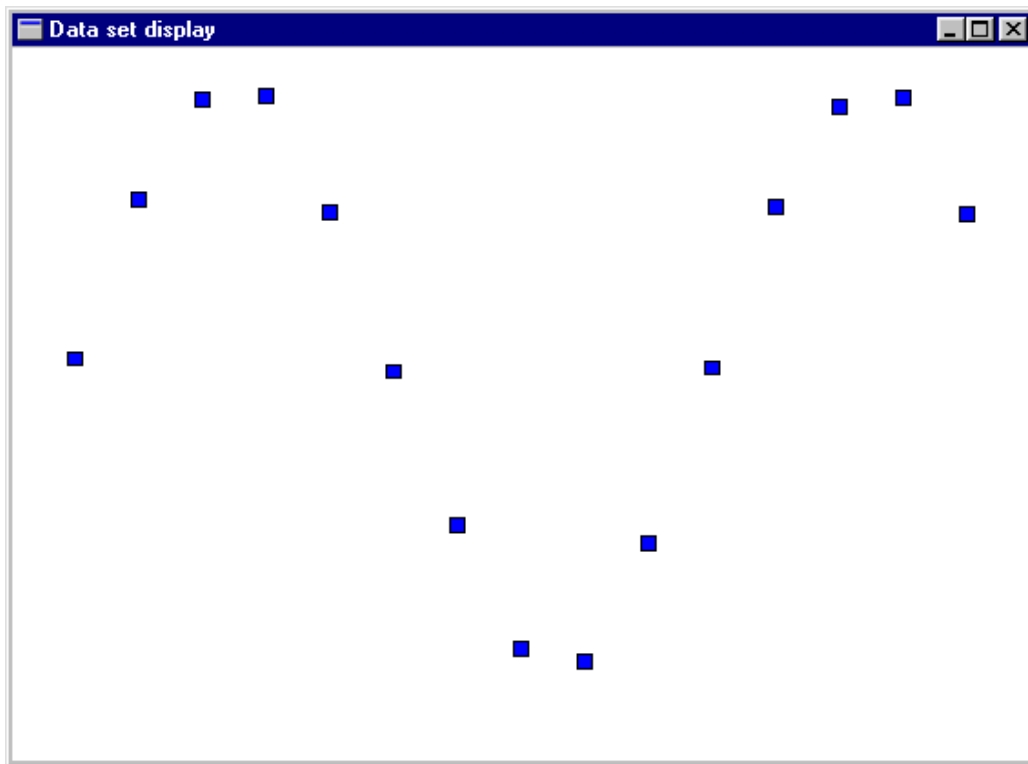
```
4.90 2.41 0.82 0.77 2.60 5.10 7.52 9.45 9.65 7.81 5.04 2.51 0.95
0.80 2.62
```

One statement we can make by performing statistical analysis is that the phenomenon appears to come from the interval $0 \dots 10$ with an average value of 4.97. We could perform further statistical analysis and report for example, the standard deviation 2.95 of the observations.

Another approach for attempting to understand the nature of the phenomenon is to try to detect whether our sequence of observations represents a pattern. For example, are the numbers arranged in descending or Fibonacci-type order? If no patterns are recognized, data visualization is sometimes attempted. For example, a natural way to plot the data set values in a two-dimensional manner would be to have the y-axis corresponding to data set values and the x-axis corresponding to positions in the sequence (e.g, the first two data set values correspond to points (1, 4.9) and (2, 2.41)). Such a plot for the data set is given Figure 1. The plot indicates that the data set does have structure—the values appear to be generated in a sine-line manner. (If you find the plot counterintuitive with respect to data set values, remember that in graphical displays the origin is the upper left corner).

Your assignment is to develop a simple data visualization tool that translates a numeric data set into a set of plotted points for analysis. Your program should be able to produce the plot of Figure 1. The interface to your program will be via a console window. Figure 2 displays the user interac-

Figure 1: Visualization of data values.



tion that produced the plot of Figure 1.

Solution method

Use the EzWindows graphical library and its classes `SimpleWindow` and `RectangleShape` to produce the visualization. In particular, your program should accomplish its task by iteratively drawing squares using the EzWindows `RectangleShape` class, where the i^{th} square is centered at the position whose x-coordinate is i and whose y-coordinate is the i^{th} value in the data set. Each square should be 0.25 centimeters on a side.

The data set is to come from standard input. The data values are floating point and should be represented using the type `float`. The size of the data set you are to process is unknown, but in no case should you plot more than 15 data values. The data values should come from the inclusive interval $0 \dots 10$. Data values less than 0 should be displayed as if their value is 0 along with an error message; data values greater than 10 should be displayed as if their value is 10 along with an error message. If the user provides too many points, error messages should be generated for data values that are not displayed. Note that end-of-file is signaled in a Windows command window by entering `CTL-Z` at the end of a data set.

Based on the preceding discussion, an appropriate size for the display window would be 16 centimeters wide (1 unit greater than the maximum number of data points and 11 centimeters high (1 unit greater than the largest possible data value).

Notes

- Graphical programs require a project file when using the Borland IDE. The project file for

Figure 2: Example user interaction in console window.

```

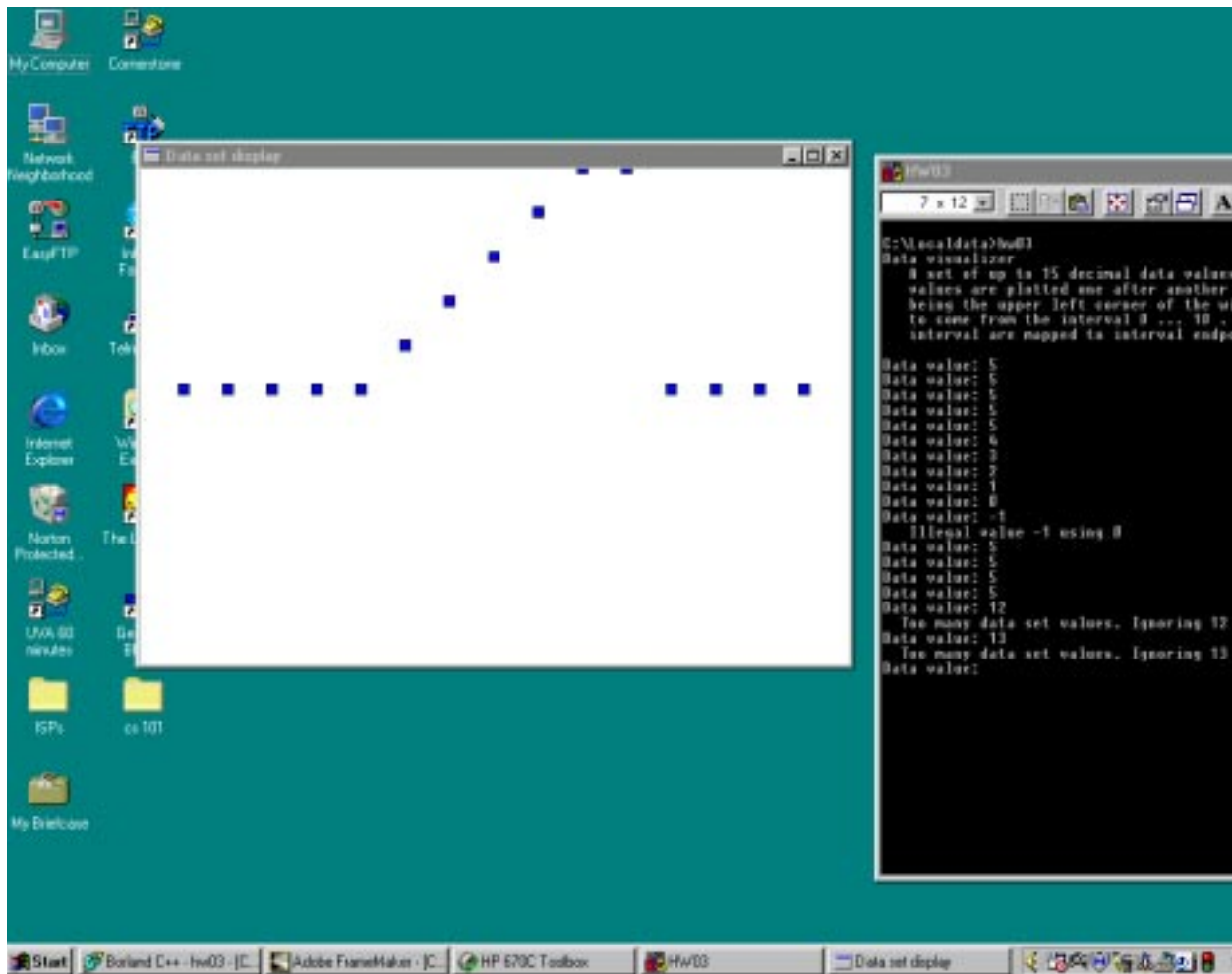
MS-DOS HW03
7 x 12
C:\Localdata>hw03
Data visualizer
  A set of up to 15 decimal data values are processed.  The data
  values are plotted one after another with the origin being
  being the upper left corner of the window.  Data values are
  to come from the interval 0 ... 10 .  Data values outside this
  interval are mapped to interval endpoints.

Data value: 4.90
Data value: 2.41
Data value: 0.82
Data value: 0.77
Data value: 2.60
Data value: 5.10
Data value: 7.52
Data value: 9.45
Data value: 9.65
Data value: 7.81
Data value: 5.04
Data value: 2.51
Data value: 0.95
Data value: 0.80
Data value: 2.62
Data value:
  
```

this assignment, `hw03.ide`, was created when archive `hw03.exe` is decompressed. Your `ApiMain()` is to be placed in `hw03.cpp`. A skeleton `hw03.cpp` file is created for you during the decompression of `hw03.exe`. The skeleton file contains a blank standard program header and a function `ApiMain()` with an empty body.

- When working on a project, the project file must be opened *before* the source file(s) are opened. Therefore, when working on this assignment, open `hw03.ide` *before* opening `hw03.cpp`. To open an `ide` file, select the item `Open Project` under Borland's `Project` menu.
- To open `hw03.cpp` for editing, double click on its filename in the project window (this window is typically displayed in the bottom of the IDE). *Remember to save your program frequently.*
- You should run your program from a command window. When your program runs, it will create the graphical display window. Note with this release of `EzWindows` the graphical display is not automatically redrawn when a covering window is moved away. Therefore, before you run your program, resize the command window and adjust its position so that it is on the left side of the screen. This situation is depicted in Figure 3.
- The project file for this assignment was developed for use in one of the on-grounds ITC computer laboratories such as the Catlin Laboratory (The Stacks). In particular, as required by the Borland compiler, the project file specifies the locations of local and standard libraries. Our specification is their location on the ITC servers. If you decide to work on this assignment elsewhere, please refer to the handout available in the documents section of the class web site.

Figure 3: Screen capture of graphical and command windows.



- File `hw03.cpp` should be submitted electronically before the laboratory in which it is due. You should also turn in a hardcopy at the beginning of the laboratory in which it is due.
- The program should follow course programming style guidelines. The guideline is available on the CS 101 website. In particular named constants should be used to represent the number of data values and the data value interval. Be sure that you consistently use the named constants rather than literal values in your program.
- The grading criteria for this assignment can also be found on the CS 101 website.