

p6.22

Henry Hsu

October 19, 2011

1 Specification

Write a function

```
void bar_chart(vector<double> data)
```

that displays a bar chart of the values in data, using asterisks, like this:

```
*****  
*****  
*****
```

You may assume that all values in data are positive. First figure out the maximum value in data. That value's bar should be drawn with 40 asterisks. Shorter bars should use proportionally fewer asterisks.

2 Analysis/Design

since specification does not say where the data comes from. Let's randomly generate the data values. we can write a function to fill the vector with the data. No limits are est on the range of data values except that they are positive for p6.21 (any values for p6.22)

Set

- randomly generate data values into a vector of doubles
- calculate bar lengths
- display the bars

3 Implementation

Since there may be large amounts of data in the vector, we can pass it in from main by refer

We can generate the values using the built-in function `\verb'rand()'`. This generates pseud

the vector has a function built-in that knows the number of elements it holds named `\verb'si`

"p6_22.cpp" 2≡

```
⟨ include files 4c ⟩  
⟨ get input 3a ⟩  
⟨ fill captions 3b ⟩  
⟨ create bar data 4a ⟩  
⟨ bar chart 4b ⟩
```

```
int main()  
{  
    vector <double> data = get_input ();  
    vector <string> captions = fill_captions ();  
  
    /* The following performs very basic input validation */  
    if (data.size() == 0)  
    {  
        cout << "Program requires at least one valid input to continue." << endl;  
    }  
    else if (data.size() > 5)  
    {  
        cout << "Maximum of only 5 data entry allowed." << endl;  
    }  
    else  
    {  
        /* This code block checks the vector for the element with the largest value */  
        double max_value = data[0];  
        for (int counter = 1; counter < data.size (); counter++)  
        {  
            if (data [counter] > max_value)  
            {  
                max_value = data [counter];  
            }  
        }  
  
        /* The below function creates a vector of % that matches the data vector element  
        vector<double> bar_data = create_bar_data(data, max_value);  
  
        /* The below function outputs the user input along with the appropriate bars */  
        bar_chart (data, bar_data, captions);  
    }  
}  
◇
```

$\langle \text{get input 3a} \rangle \equiv$

```
vector <double> get_input()
{
    vector <double> data;
    bool continue_input = true;
    while (continue_input)
    {
        cout << "Please input up to 5 values no greater than 100, use any letter
        double input;
        cin >> input;                // input validation
        if (cin.fail() || input > 100 || input < -100)
        {
            continue_input = false;
        }
        else
        {
            data.push_back(abs(input));    // writes input to vector element
        }
    }
    return data;                      // returns user input as a vector
}
```

◇

Fragment referenced in 2.

$\langle \text{fill captions 3b} \rangle \equiv$

```
vector <string> fill_captions()
{
    vector <string> captions(5);
    captions[0] = "Egypt";
    captions[1] = "France";
    captions[2] = "Japan";
    captions[3] = "Uruguay";
    captions[4] = "Switzerland";
    return captions;
}
```

◇

Fragment referenced in 2.

⟨ create bar data 4a ⟩ ≡

```
vector <double> create_bar_data(vector <double> database, double max_value)
{
    vector <double> bar_data;           // creates matching vector
    for (int counter = 0; counter < database.size(); counter++)
    {
        bar_data.push_back (((database[counter] / max_value) * 100));
    }

    return bar_data;
}
```

◇

Fragment referenced in 2.

⟨ bar chart 4b ⟩ ≡

```
void bar_chart(vector <double> data, vector <double> bar_data, vector <string> captions)
{
    for (int counter = 0; counter < bar_data.size (); counter++)
    {
        cout << right << setw(15) << captions[counter] << " ";
        for (double bar_counter = 0; bar_counter < bar_data[counter]; bar_counter++)
        {
            cout << "*";
        }
        cout << endl;
    }
}
```

◇

Fragment referenced in 2.

These are the include files needed for library function calls

⟨ include files 4c ⟩ ≡

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <vector>
```

```
using namespace std;
```

◇

Fragment referenced in 2.

4 Test

e.g. if the data generated was: 10, 20, 40 Expect: ***** (10) *****
(20) ***** (40)
e.g. if the data generated was: 5, 10, 20 Expect: ***** (5) *****
(10) ***** (20)

```
C:\Users\Echo\Desktop\cs102>a
Please input up to 5 values no greater than 100, use any letter key to quit: 10
Please input up to 5 values no greater than 100, use any letter key to quit: 20
Please input up to 5 values no greater than 100, use any letter key to quit: 40
Please input up to 5 values no greater than 100, use any letter key to quit: a
    Egypt *****
    France *****
    Japan *****
```

```
C:\Users\Echo\Desktop\cs102>a
Please input up to 5 values no greater than 100, use any letter key to quit: 5
Please input up to 5 values no greater than 100, use any letter key to quit: 10
Please input up to 5 values no greater than 100, use any letter key to quit: 20
Please input up to 5 values no greater than 100, use any letter key to quit: q
    Egypt *****
    France *****
    Japan *****
```