# Lab 6 - 2 Programs - (Lec. 6)

**Lab 6** -  **Program #1 -** Write one number to a text file.

// This program writes one number two times to a text file. (see output below)

// The program uses the setprecision and fixed manipulators to format file output.

// The first output simply writes the number to file, without formatting decimal places.

// The second output formats the number to 2 places to the right of the decimal point.

**Fill in the blanks** - then enter the code and run the program and see if correct.

#include <iostream>

(1.) Include other libraries that may

be needed.

using namespace std;

int main()

{

fstream dataFile;

double number = 17.816392;

(2.) Write one statement to open

in output mode.

The file name is: **values.txt**

(3.) Format for fixed-point notation.

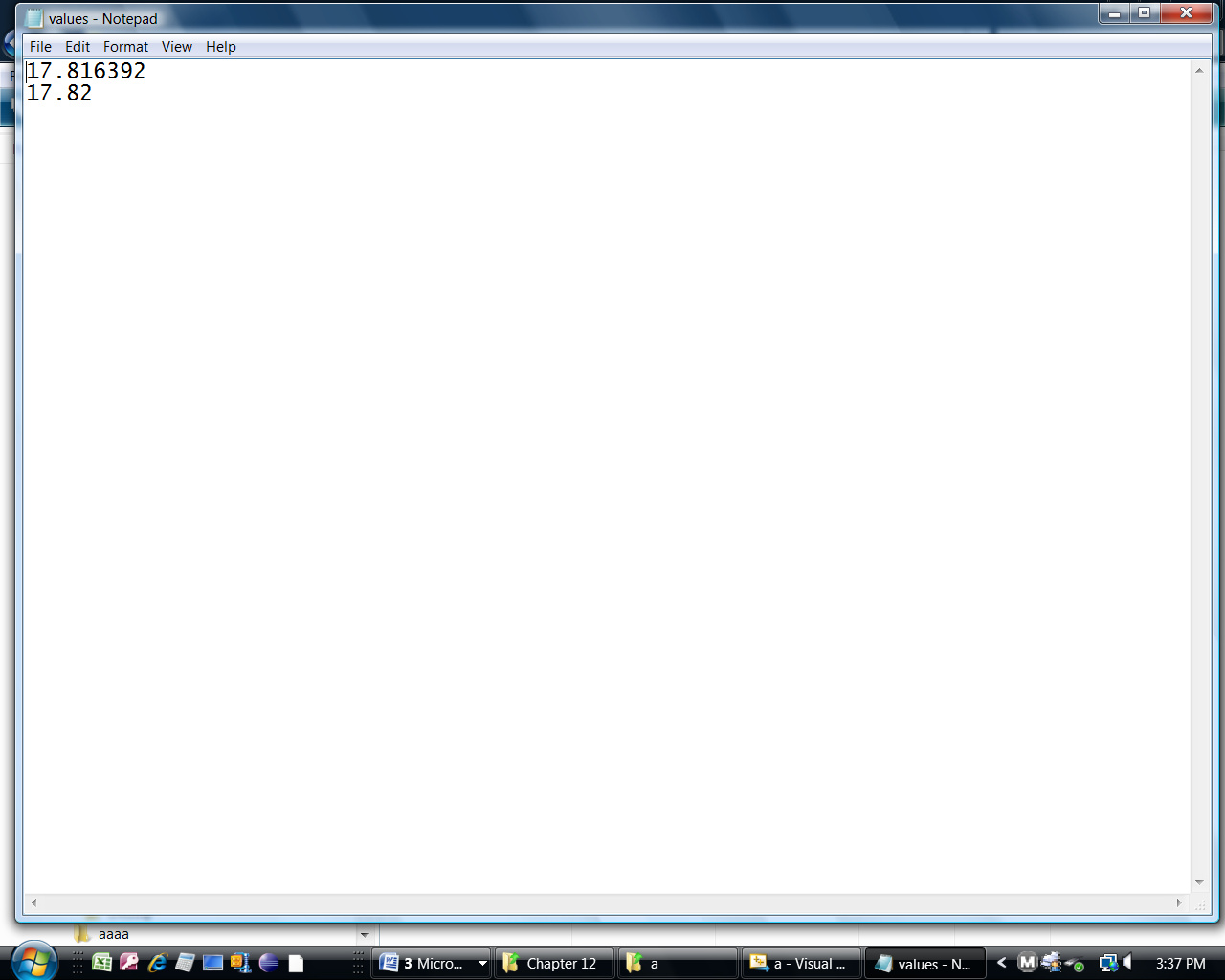
(4.) Write **number** to the file.

(5.) Format output to 2 decimal places.

(6.) Write **number** to the file.

cout << "Data has been written to file.\n";

(7.) Close the file.



return 0;

}

(8.) Open the text file to see the output.

**Lab 6** -  **Program #2 -** Write one number to a binary file. Use the write() and read() functions

with binary data, where the data is not char type.

(Typecasting is required) - Name the file: *test.bin*

Fill in the blanks, then enter the code and run the program.

#include <iostream>

Note: The data is **int** type, so typecasting is

required in the **write()** and **read()** functions.

#include <fstream>

using namespace std;

int main()

{

const int SIZE = 10;

int numbers[SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

fstream file;

(1.) Write one statement to open

the file for output in binary mode.

The file name is: **numbers.dat**

cout << "Writing the data to the file.\n";

(2.) Write one statement to write the

contents of the array to the file.

file.close();

(3.) Write one statement to open

the file for input in binary mode.

The file name is: **numbers.dat**

// Read the contents of the file into the array.

cout << "Now reading the data back into memory.\n";

(4.) Write one statement to read the

contents of the file into the array.

// Display the contents of the array.

for (int count = 0; count < SIZE; count++)

cout << numbers[count] << " ";

cout << endl;

file.close();

/\* **OUTPUT**

Writing the data to the file.

Now reading the data back into memory.

1 2 3 4 5 6 7 8 9 10

return 0;

}