## Cube Root Solution

*Description:*

Calculate the cube root of one number, and the library function is not allowed.

*Interface Illustration:*

* *Prototype:*

public static double getCubeRoot(double input)

* *Input:*

double input

* *Return:*

double the cube root value of the parameter, keep one decimal number.

*Code:*

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

const double EPS = 1e-6;

class DataDecide {

public:

static double getCubeRoot(double input) {

double first = 0.0;

double middle = 0.0;

double end = input;

/\*

\*\* The comparsion between two double variables need the help of

\*\* precision. It can control distance between two variables.

\*/

while ( end - first >= EPS ) {

middle = ( first + end ) / 2.0;

if ( middle \* middle \* middle > input ) {

end = middle;

} else if ( middle \* middle \* middle < input ) {

first = middle;

} else

return middle;

}

return middle;

};

};

int main( ) {

double input;

cin>>input;

double result = DataDecide::getCubeRoot(input);

/\*

\*\* The setprecision function is under the iomanip header file.

\*\* The function can help control the print out byte number.

\*\* setprecision(2) helps narrow the byte number as 2.

\*\* The number output afterwords would only include 2 bytes.

\*/

cout<<setprecision(2);

cout<<result<<endl;

}