MONTE\_CARLO\_SYSTEM

Employ Monte Carlo Simulation to compute the area bounded by the x-axis in the interval [mMinX, mMaxX], and the range of the function value inside [mMinY, mMaxY].

mMinX = 0;

mMaxX = 6;

mMinY = -4.0;

mMaxY = 4.0;

There are four functions:

f0(x) = -4 e-x/2.0 cos(x2)

f1(x) = -cos(x)cos(5x) + 4/(1+x2)

f2(x) = -3sin(x)+cos(4x)

f3(x) = (ex –e-x)/( ex +e-x). This is hyperbolic tangent.

where e is the base of the natural log, which is approximated as 2.718281828.

/\*

Show the message to the user.

\*/

void MONTE\_CARLO\_SYSTEM::askForInput( )

{

cout << "MONTE\_CARLO\_SYSTEM::askForInput( )" << endl;

cout << "Use keys to control......" << endl;

cout << "1: FUNCTION\_EXPONENTIAL\_COSINE" << endl;

cout << "2: FUNCTION\_COSINE" << endl;

cout << "3: FUNCTION\_SINE\_COSINE" << endl;

cout << "4: FUNCTION\_HYPERBOLIC\_TANGENT" << endl;

cout << "m: minimum number of sample points" << endl;

cout << ",: decrease the number of sample points" << endl;

cout << ".: increase the number of sample points" << endl;

cout << "/: maximum number of sample points" << endl;

}

/\*

Handle the key events based on the key usage.

\*/

bool MONTE\_CARLO\_SYSTEM::handleKeyPressedEvent( int key )

{

……

computeSamples( );

computeArea( );

……

}

/\*

Reset the system state.

Steps:

compute the samples

compute the area

\*/

void MONTE\_CARLO\_SYSTEM::reset( )

/\*

Return the range [mMinX, mMaxX] of x to [minX, maxX].

\*/

void MONTE\_CARLO\_SYSTEM::getRangeOfX( double &minX, double &maxX ) const

/\*

Uniformly generate all the random sample points

inside [mMinX, mMaxX] x [mMinY, mMaxY].

The number of sample points is mNumSamples.

\*/

void MONTE\_CARLO\_SYSTEM::computeSamples( )

/\*

Compute the area based on the Monte Carlo Simulation.

Steps:

1. compute the number of samples inside the area

2. compute the ratio which is equal to

the number of samples inside the area

divided

by the total number of samples

3. Based on the ratio, compute the area

4. Show the area

5. return the area value

\*/

double MONTE\_CARLO\_SYSTEM::computeArea( ) const

{

……

cout << "Area:" << area << endl;

return area;

}

/\*

Return the number of sample points.

\*/

int MONTE\_CARLO\_SYSTEM::getNumSamples( ) const

/\*

Return the sample point (x,y)

\*/

void MONTE\_CARLO\_SYSTEM::getSample(int sampleIndex, double &x, double &y ) const

/\*

Based on the function type mFunctionType,

compute the function value for a given value x.

Return the function value.

\*/

double MONTE\_CARLO\_SYSTEM::getValue(double x) const

/\*

Given a point (x,y).

Check if the point lies inside the area.

Return true if it lies inside the area.

Return false otherwise.

The area is defined as the region bounded by

the x-axis and the curve of the function f(x).

Assume that (x,y) lies inside area, then

one of the following conditions must be satisfied.

- If f(x) >= 0 and y >=0, then y <= f(x).

- If f(x) <= 0 and y <=0, then y >= f(x).

Otherwise, (x,y) does not lay inside the area.

\*/

bool MONTE\_CARLO\_SYSTEM::isInsideArea( double x, double y ) const