CLASS MATRIX REFERENCE MANUAL

RETURN	METHOD NAME	PARAMETERS	DESCRIPTION
INITIALIZATION			
	Matrix	void	New Matrix
	Matrix	<T>** X int Rows int Cols	Convert two-dimensions array *X* of size *Rows* x *Cols* to a matrix
int	GetLengthRows	void	Get number of rows of the matrix
int	GetLengthCols	void	Get number of columuns of the matrix
SELECTION METHODS			
Vector<T>*	GetRowRef	int Index	Get reference of *Index* row of the matrix
Vector<T>*	GetRowCopy	int Index	Get copy of *Index* row of the matrix
Vector<T>*	GetColCopy	int Index	Get copy of *Index* column of the matrix
T	GetValue	int RowIndex int ColIndex	Get an element of the matrix
ADD / REMOVE METHODS			
void	Clear	void	Remove all the elements from matrix
void	AddRowRef	Vector<T>* V	Add a row reference to matrix
void	AddRowCopy	Vector<T>* V	Add a new row to matrix
void	AddRowCopy	T* V int N	Add a new row to matrix of length *N*
void	AddRowRefAt	Vector<T>* V int Index	Add a row reference to matrix on *Index* position
void	AddRowCopyAt	Vector<T>* V int Index	Add a row to matrix on *Index* position
void	AddRowCopyAt	T* V int N int Index	Add a row to matrix of length *N* on *Index* position
void	AddColCopy	Vector<T>* V	Add a new column to matrix
void	AddColCopy	T* V int N	Add a new column to matrix of length *N*
void	AddColCopyAt	Vector<T>* V int Index	Add a new column to matrix at on *Index* position
void	AddColCopyAt	T* V int N int Index	Add a new column to matrix of length *N* on *Index* position
void	RemoveRow	int Index	Remove row on *Index* position from matrix
void	RemoveCol	int Index	Remove row on *Index* position from matrix
Matrix<T>*	ExtractRows	int FromIndex int ToIndex	Build a new matrix with rows position from *FromIndex* to *ToIndex*
Matrix<T>*	ExtractCols	int FromIndex int ToIndex	Build a new matrix with columns position from *FromIndex* to *ToIndex*
PRE-BUILT VECTORS			
Matrix<double>*	ZeroMatrix	int RowsNumber int ColsNumber	Build a new matrix of zeros with size *RowsNumber* x *ColsNumber*
Matrix<double>*	RandMatrix	int RowsNumber int ColsNumber	Build a new matrix of zeros with size *RowsNumber* x *ColsNumber*

CLASS MATRIX REFERENCE GUIDE

RETURN	METHOD NAME	PARAMETERS	DESCRIPTION
MATHEMATICAL METHODS			
void	SumScalar	<T> X	Sum *X* to each element of the vector
void	ProductScalar	<T> X	Multiply *X* to each element of the vector
void	DivideScalar	<T> X	Divide each element of the vector to *X*
void	PowScalar	<T> X	Pow each element of the vector to *X*
void	SumMatrix	Matrix<T>* M	Sum a vector with a matrix
void	SubtractMatrix	Matrix<T>* M	Subtract a vector with a matrix
Vector<T>*	ProductVector	Vector<T>* V	Product matrix x vector
Vector<T>*	ProductVector	Matrix<T>* M Vector<T>* V	Product matrix x vector
Matrix<T>*	ProductVectorVector	Vector<T>* V1 Vector<T>* V2	Product vector x vector‘
Matrix<T>*	ProductMatrixMatrix	Matrix<T>* M1 Matrix<T>* M2	Product matrix x matrix
INPUT / OUTPUT METHODS			
Matrix<double>*	Load	char* Filename	Load a matrix from a file
void	Save	char* Filename	Save a matrix to a file
void	Print	void	Print a matrix to output
void	Print	char* MatrixName	Print a matrix to output

CLASS MATRIX EXAMPLES

```
#include "Matrix.h"

using namespace onlinesvr;


int main ()
{
      // Make a new matrix
      Matrix<int>* M1 = new Matrix<int>();

      // Fill matrix
      for (int i=0; i<5; i++) {
            Vector<int>* V1 = new Vector<int>();
            for (int j=0; j<5; j++) {
                  V1->Add(j);
            }
            M1->AddRowRef(V1);
      }

      // Add 5 to each element of the matrix
      M1->SumScalar(5);

      // Print matrix
      M1->Print("M1");

      // Product Matrix x Vector
      Vector<int>* V2 = new Vector<int>();
      for (int i=0; i<5; i++) {
            V2->Add(2);
      }
      Vector<int>* V3 = M1->ProductVector(V2);
      V3->Print("M1xV2");

      // Rand matrix with 4 rows and 3 columns
      Matrix<double>* M2 = Matrix<double>::RandMatrix(4,3);
      M2->Print("M2");

      // Save matrix
      M2->Save("M2.vec");

      // Delete matrix and vectors
      delete M1;
      delete M2;
      delete V2;
      delete V3;
}
```