|  |  |  |  |
| --- | --- | --- | --- |
|  | 11/23/2017 |  | |
|  | |  | |
| MyVector Class  *My Vector Class work like the Vector Template Class* | | | |
|  |  | |  |
|  |  | | Muhammad Hamza  2016-UET-NML-CS-28 |

MyVector Class

My Vector Class work like the Vector Template Class

It is my created class that provide you all feature that (Vector template) give you. You can use my (MyVector) class as a Vector template.

class MyVector{

public:

int Lenth;

double \*ptr;

public:

MyVector();

MyVector(int);

MyVector(const MyVector&); /// For Constructor Copy...

~MyVector();

bool Redimention(int);

void Display();

int SIZE();

void PushBack(int);

/// Over Loaded Methods...

double operator[](int);

MyVector operator\*(int);

MyVector operator\*(MyVector &);

MyVector operator+(MyVector &);

/// Friend Functions...

friend MyVector operator\*(MyVector & , int);

friend MyVector operator\*(MyVector & , MyVector &);

friend MyVector operator+(MyVector & , MyVector &);

friend void operator <<(ostream& out, MyVector &temp);

};

Member Functions and Data Member Detail

class MyVector {

private: /// Private Members

* int Lenth;

Length is Private Data Member of the Class MyVector it store the Vector Length of any object.

* double \*ptr;

It is Double type pointer with the help of this pointer we make the vector for the object of MyVector Class its SIZE depend on the length that object.

public: /// Public Member Functions

# MyVector ();

MyVector::MyVector(){

Lenth=0;

ptr=NULL;

cout<<"One object is Created that has zero Lenth and Null Pointer Array...\n";

}

This is a by default constructor that automatically called when an Object of MyVector Class is Created without any parameters in any place in program.

# MyVector(int);

It is Second constructor of MyVector Class that accept a int type data as a parameter. With the help of this constant constructor (Set the Length of that object and a vector create for it).

# MyVector (const MyVector&) ;

It is also a Constructor of MyVector class but it is deferent from above Constructors because it is accepting an Object as a parameter and create a copy of this object then return this to where it is calling copy.

# ~MyVector();

This is MyVector Class Destructor it automatically called when object of MyVector class Destroyed.

# bool Redimention(int);

This is a Re Dimension member function of MyVector Class. It Accept a int type constant and update the length SIZE of that object.

For Example: if the Previous SIZE the Length and Vector is 5 and when we call Re dimension (10) and passed 10 through parameter this function Set the Length 10 and Increase the SIZE of Vector.

# int SIZE();

This is a SIZE function that return the SIZE of object Vector when you call it.

# /// Over Loaded Member Methods...

# double operator[](int);

This is an overloaded Function of MyVector Class and accept the int type constant and return the value that place a specific index in the object vector.

# MyVector operator\*(int);

This is also an overloaded Function for Multiplication of the vector index by index with a constant number. And return the addition vector.

# MyVector operator\*(MyVector &);

This is a Multiplication overloaded function it is accept the address of one object through the (operator) key word and one object address through parameter. This function returns the new vector that fill with the product of those two vectors.

# MyVector operator+(MyVector &);

This is an Addition overloaded function it is accept the address of one object through the (operator) key word and one object address through parameter. This function returns the new vector that fill with the Addition of those two vectors.

# /// Friend Member Functions...

# friend MyVector operator\*(MyVector & , int);

It is a Friend member function that accept the address one Object of MyVector Class and an int type constant number and return the Multiplication of the vector index by index with a constant number. And return the addition vector.

# friend MyVector operator\*(MyVector & , MyVector &);

This is a Multiplication overloaded Friend function it is accept the addresses of two Object of class MyVector. This function returns the new vector that fill with the product of those two vectors index by index.

# friend MyVector operator+(MyVector & , MyVector &);

This is an Addition overloaded Friend function it is accept the addresses of two Object of class MyVector. This function returns the new Vector that fill with the Addition of those two vectors index by index.

# friend void operator << (ostream & out, MyVector &temp);

This is a friend Function for the

# Operator Overloading through Non-Member Functions...

Remarks: These Nonmember Functions use the a friend function of MyVector class and Perform the their related Works…

# MyVector operator\*=(MyVector &, int);

It is an overloaded Nonmember function that accept the address one Object of MyVector Class and an int type constant number and return the Multiplication of the vector index by index with a constant number. And return the addition vector of MyVector class.

# MyVector operator\*=(MyVector &,MyVector &);

This is a Nonmember Multiplication overloaded function it is accept the addresses of two Object of class MyVector. This function returns the new vector that fill with the product of those two vectors index by index.

# MyVector operator+=(MyVector &,MyVector &);

This is an Nonmember Addition overloaded function it is accept the addresses of two Object of class MyVector. This function returns the new Vector that fill with the Addition of those two vectors index by index.