

MPI Implementation of some eigen libraries others

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Genfun::Argument Class Reference . . . . .	5
3.2	ArithProgression Class Reference . . . . .	5
3.3	Audit Class Reference . . . . .	6
3.4	background_task Class Reference . . . . .	7
3.5	Complex Class Reference . . . . .	7
3.6	Core Class Reference . . . . .	8
3.7	Grad Class Reference . . . . .	9
3.8	InitiateVectorMethod< ItemType > Class Template Reference . . . . .	10
3.9	MPI_BC Class Reference . . . . .	10
3.9.1	Constructor & Destructor Documentation . . . . .	11
3.9.1.1	MPI_BC() . . . . .	11
3.10	MPI_BC_Generic< T, Q, R > Class Template Reference . . . . .	11
3.11	MPI_sorting_methods Class Reference . . . . .	12
3.12	MPIInput Class Reference . . . . .	12
3.12.1	Constructor & Destructor Documentation . . . . .	13
3.12.1.1	MPIInput() . . . . .	13
3.12.1.2	~MPIInput() . . . . .	13

3.12.2 Member Function Documentation . . . . .	13
3.12.2.1 getDataPack() . . . . .	13
3.13 OMP< T > Class Template Reference . . . . .	14
3.13.1 Detailed Description . . . . .	14
3.14 Partstruct Struct Reference . . . . .	15
3.15 PassFail Class Reference . . . . .	15
3.16 part1::Point Class Reference . . . . .	16
3.16.1 Detailed Description . . . . .	16
3.17 ProbDist Class Reference . . . . .	16
3.17.1 Detailed Description . . . . .	17
3.18 Progression Class Reference . . . . .	17
3.19 QTstyle_Test Class Reference . . . . .	18
3.19.1 Detailed Description . . . . .	18
3.20 RandomNumberGenerator Class Reference . . . . .	18
3.21 Stack< T, CONT > Class Template Reference . . . . .	19
3.22 StandardNormalDistribution Class Reference . . . . .	19
3.22.1 Detailed Description . . . . .	20
3.23 StatisticalDistribution Class Reference . . . . .	20
3.23.1 Detailed Description . . . . .	21
3.23.2 Constructor & Destructor Documentation . . . . .	21
3.23.2.1 StatisticalDistribution() . . . . .	22
3.23.2.2 ~StatisticalDistribution() . . . . .	22
3.24 Str Class Reference . . . . .	22
3.24.1 Detailed Description . . . . .	23
3.24.2 Constructor & Destructor Documentation . . . . .	23
3.24.2.1 Str() [1/2] . . . . .	23
3.24.2.2 Str() [2/2] . . . . .	23
3.25 Student_info Class Reference . . . . .	24
3.26 TemplateUnderTest< T > Class Template Reference . . . . .	24
3.27 Trap Class Reference . . . . .	24
3.28 Vec< T > Class Template Reference . . . . .	25
3.29 tutorial::Vec< T > Class Template Reference . . . . .	26
3.30 VectorMethodTest Class Reference . . . . .	26

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Genfun::Argument . . . . .	5
background_task . . . . .	7
Complex . . . . .	7
Core . . . . .	8
Audit . . . . .	6
Grad . . . . .	9
PassFail . . . . .	15
InitiateVectorMethod< ItemType > . . . . .	10
MPI_BC_Generic< T, Q, R > . . . . .	11
MPI_sorting_methods . . . . .	12
OMP< T > . . . . .	14
Partstruct . . . . .	15
part1::Point . . . . .	16
ProbDist . . . . .	16
Progression . . . . .	17
ArithProgression . . . . .	5
QTeststyle_Test . . . . .	18
RandomNumberGenerator . . . . .	18
Stack< T, CONT > . . . . .	19
StatisticalDistribution . . . . .	20
StandardNormalDistribution . . . . .	19
Str . . . . .	22
Student_info . . . . .	24
TemplateUnderTest< T > . . . . .	24
TestCase . . . . .	
MPI_BC . . . . .	10
MPIInput . . . . .	12
Vec< T > . . . . .	25
Vec< char > . . . . .	25
TestFixture . . . . .	
VectorMethodTest . . . . .	26
Trap . . . . .	24
tutorial::Vec< T > . . . . .	26



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Genfun::Argument	5
ArithProgression	5
Audit	6
background_task	7
Complex	7
Core	8
Grad	9
InitiateVectorMethod< ItemType >	10
MPI_BC	10
MPI_BC_Generic< T, Q, R >	11
MPI_sorting_methods	12
MPIInput	12
OMP< T >	
Openmp explanation	14
Partstruct	15
PassFail	15
part1::Point	16
ProbDist	16
Progression	17
QTstyle_Test	
A test class - find and replace from this template for future class definitions	18
RandomNumberGenerator	18
Stack< T, CONT >	19
StandardNormalDistribution	
Standard Normal Distribution Implementation	19
StatisticalDistribution	
Statistical Distribution Class	20
Str	
A constructor with a character pointer input	22
Student_info	24
TemplateUnderTest< T >	24
Trap	24
Vec< T >	25
tutorial::Vec< T >	26
VectorMethodTest	26





## Chapter 3

# Class Documentation

### 3.1 Genfun::Argument Class Reference

#### Public Member Functions

- **Argument** (int ndim=0)
- **Argument** (const [Argument](#) &)
- **Argument** (std::initializer\_list< double >)
- const [Argument](#) & **operator=** (const [Argument](#) &)
- double & **operator[]** (int I)
- const double & **operator[]** (int i) const
- unsigned int **dimension** () const

#### Friends

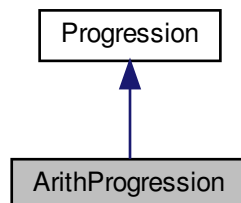
- std::ostream & **operator<<** (std::ostream &o, const [Argument](#) &a)

The documentation for this class was generated from the following file:

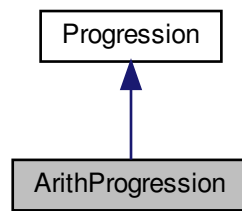
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Argument.h

### 3.2 ArithProgression Class Reference

Inheritance diagram for ArithProgression:



Collaboration diagram for ArithProgression:



### Public Member Functions

- **ArithProgression** (long i=1)

### Protected Member Functions

- virtual long **nextValue** ()

### Protected Attributes

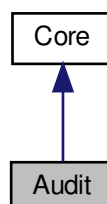
- long **inc**

The documentation for this class was generated from the following file:

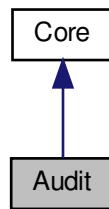
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp2.hpp

## 3.3 Audit Class Reference

Inheritance diagram for Audit:



Collaboration diagram for Audit:



### Public Member Functions

- **Audit** (std::istream &is)
- std::istream & **read** (std::istream &)
- double **grade** () const
- bool **valid** () const
- bool **fulfill\_reqs** () const

### Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp\_dynamicbindingandinheritance.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp\_dynamicbindingandinheritance.cxx

## 3.4 background\_task Class Reference

### Public Member Functions

- void **operator()** () const

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp\_thread.cxx

## 3.5 Complex Class Reference

### Public Member Functions

- **Complex** (double r, double i=0)

## Friends

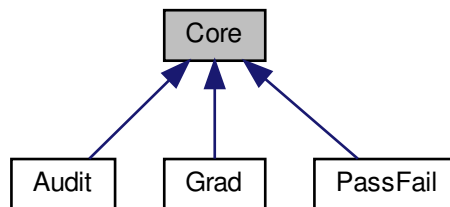
- `bool operator== (const Complex &a, const Complex &b)`

The documentation for this class was generated from the following file:

- `/home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp2.cxx`

## 3.6 Core Class Reference

Inheritance diagram for Core:



## Public Member Functions

- **Core** (`std::istream &is`)
- `std::string name () const`
- `virtual std::istream & read (std::istream &)`
- `virtual double grade () const`
- `virtual bool valid () const`
- `virtual bool fulfill_reqs () const`

## Protected Member Functions

- `std::istream & read_common (std::istream &)`
- `virtual Core * clone () const`

## Protected Attributes

- `std::string n`
- `double midterm`
- `double final`
- `std::vector< double > homework`

## Friends

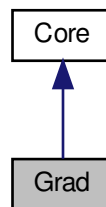
- class **Student\_info**

The documentation for this class was generated from the following files:

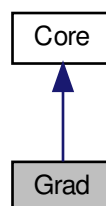
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp\_dynamicbindingandinheritance.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp\_dynamicbindingandinheritance.cxx

## 3.7 Grad Class Reference

Inheritance diagram for Grad:



Collaboration diagram for Grad:



## Public Member Functions

- **Grad** (std::istream &is)
- std::istream & **read** (std::istream &)
- double **grade** () const
- bool **fulfill\_reqs** () const

### Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp\_dynamicbindingandinheritance.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp\_dynamicbindingandinheritance.cxx

## 3.8 InitiateVectorMethod< ItemType > Class Template Reference

### Public Member Functions

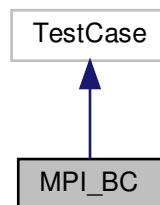
- **InitiateVectorMethod** (int, int)
- void **setup** (int \*)
- void **traits** ()
- void **SendVector** ()
- void **GetData** ()

The documentation for this class was generated from the following file:

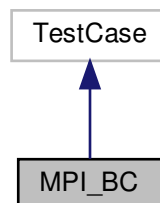
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/new.hpp

## 3.9 MPI\_BC Class Reference

Inheritance diagram for MPI\_BC:



Collaboration diagram for MPI\_BC:



## Public Member Functions

- [MPI\\_BC](#) ()  
*MPI\_BC class template filling.*
- void **Get\_input** (int, int, double \*, double \*, int \*)
- void **packData** ()
- void **time\_ellapsed** ()
- void **broadcast\_input** ()
- void **broadcast\_vector** ()
- void **buildMpiType** (double \*, double \*, int \*, MPI\_Datatype \*)
- void **Send** (float, float, int, int)
- void **SendVector** ()
- void **Receive** (float \*, float \*, int \*, int)
- void **parallelAllocateVec** (double \*, double \*, int, std::vector< int > \*, MPI\_Datatype \*)

### 3.9.1 Constructor & Destructor Documentation

#### 3.9.1.1 MPI\_BC()

`MPI_BC::MPI_BC ( )`

[MPI\\_BC](#) class template filling.

Placeholder

The documentation for this class was generated from the following files:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/MPI\_broadcast.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/MPI\_broadcast.cxx

## 3.10 MPI\_BC\_Generic< T, Q, R > Class Template Reference

## Public Member Functions

- **MPI\_BC\_Generic** (std::size\_t n)

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/MPI\_broadcast.hpp

### 3.11 MPI\_sorting\_methods Class Reference

#### Public Member Functions

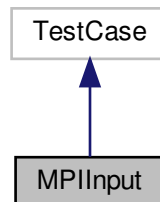
- void **Bubble\_sort** (int a[], int n)

The documentation for this class was generated from the following file:

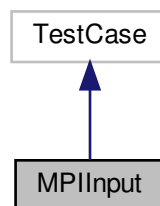
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/MPI\_reduce.cxx

### 3.12 MPIInput Class Reference

Inheritance diagram for MPIInput:



Collaboration diagram for MPIInput:



#### Public Member Functions

- [MPIInput](#) (int, int)  
*MPIInput class - constructor.*
- void **getData** (int \*, int \*, int \*)
- void **bubbleSort** (int \*, int)
- void [getDataPack](#) (float \*, float \*, int \*)
- [~MPIInput](#) ()  
*Class destructor for MPI.*
- void **test1** ()



### 3.12.1 Constructor & Destructor Documentation

#### 3.12.1.1 MPIInput()

```
MPIInput::MPIInput (
    int mr,
    int pe )
```

[MPIInput](#) class - constructor.

int int constructor

Sets up the processor ranks and size for use in other functions

At the moment, we do not have a default constructor

#### 3.12.1.2 ~MPIInput()

```
MPIInput::~MPIInput ( )
```

Class destructor for MPI.

#### Destructor

MPI\_Finalize() - What does it do?

### 3.12.2 Member Function Documentation

#### 3.12.2.1 getDataPack()

```
void MPIInput::getDataPack (
    float * a_ptr,
    float * b_ptr,
    int * n_ptr )
```

< Placeholder

< Keeping track of where the data is

MPI\_pack

MPI\_pack allows one to explicitly store unctiguous data in contiguous memory locations, and MPI\_unpack can be used to copy data from a contiguous buffer into noncontiguous memory locations.

< Now pack the data into buffer. Positon = 0 says start at beginning of buffer

MPI\_unpack

MPI\_unpack can be used to copy data from a contiguous buffer into noncontiguous memory locations. i.e. it is just the complete opposite of pack

The documentation for this class was generated from the following files:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/MPI\_IO.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/MPI\_IO.cxx

### 3.13 OMP< T > Class Template Reference

Openmp explanation.

```
#include <openmp1.hpp>
```

#### Public Member Functions

- **OMP** (int)
- **OMP** (const **OMP** &OMPCopy)
- **OMP** & **operator=** (const **OMP** &ref)
- void **add** (T)
- void **addup** ()
- void **pi** ()

#### 3.13.1 Detailed Description

```
template<class T>
class OMP< T >
```

Openmp explanation.

As we noted earlier, we'll usually specify the number of threads on the command line, so we'll modify parallel directive was with the num\_threads clause. A clause in OpenMP is just some text that modifies a directive. The num\_threads clause, it allows a programmer to specify the number of threads that should be executed in the following block:

```
pragma omp parallel num_threads (thread_count)
```

What actually happens when the program gets to the parallel directive? Prior to the parallel directive, the program is using a single thread, the process started when the program started execution.

```
< std::cout, std::endl
```

```
< std::size_t
```

```
< std::max
```

```
< std::allocator, std::uninitialized_fill, std::uninitialized_copy
```

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp1.hpp

## 3.14 Partstruct Struct Reference

### Public Attributes

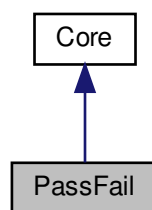
- int **class**
- double **d** [6]
- char **b** [7]

The documentation for this struct was generated from the following file:

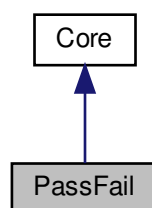
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/MPI\_struct.cxx

## 3.15 PassFail Class Reference

Inheritance diagram for PassFail:



Collaboration diagram for PassFail:



### Public Member Functions

- **PassFail** (std::istream &is)
- double **grade** () const
- bool **valid** () const
- bool **fulfill\_reqs** () const

## Additional Inherited Members

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp\_dynamicbindingandinheritance.hpp

## 3.16 part1::Point Class Reference

```
#include <lib_mpi.hpp>
```

### Public Member Functions

- **Point** (float \_x, float \_y, float \_z)

### Public Attributes

- float **x**
- float **y**
- float **z**

### 3.16.1 Detailed Description

This is a simple 3D point class

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/lib\_mpi.hpp

## 3.17 ProbDist Class Reference

```
#include <openmp_LA.hpp>
```

### Public Member Functions

- double **norm\_pdf** (const double &x)
- double **norm\_cdf** (const double &x)
- double **d\_j** (const int &, const double &, const double &, const double &, const double &, const double &)
- double **call\_price** (const double &, const double &, const double &, const double &, const double &)
- double **put\_price** (const double &, const double &, const double &, const double &, const double &)

### 3.17.1 Detailed Description

#### European Options with Monte Carlo

In this chapter, we will price a European Vanilla option via the correct analytical solution of the Black-Scholes equation, as well as via the Monte Carlo method. We won't be concentrating on an extremely efficient or optimised implementation at this stage.

-> Black Scholes Analytic pricing formula

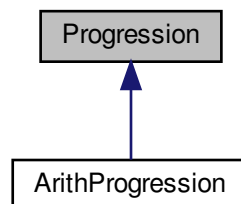
The first stage in implementation is to briefly discuss the Black Scholes analytic solution for the price of a vanilla call or put option

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp\_LA.hpp

## 3.18 Progression Class Reference

Inheritance diagram for Progression:



### Public Member Functions

- **Progression** (long f=0)
- void **printProgression** (int n)

### Protected Member Functions

- virtual long **firstValue** ()
- virtual long **nextValue** ()

## Protected Attributes

- long **first**
- long **cur**

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp2.hpp

## 3.19 QTstyle\_Test Class Reference

A test class - find and replace from this template for future class definitions.

```
#include <statistics.h>
```

### 3.19.1 Detailed Description

A test class - find and replace from this template for future class definitions.

One of the most common examples of concepts in quantitative finance is that of a statistical distribution. Random variables play a huge part in quantitative financial modelling. Derivatives, pricing, cash-flow forecasting and quantitative trading all make use of statistical methods in some fashion

Many of the chapters within this book have made use of random number generators in order to carry out pricing tasks.

In a nutshell, we are splitting the generation of (uniform integer) random numbers from draws of specific statistical distributions such that we can use the static classes elsewhere without bringing along the heavy random number generation functions.

Equally useful is the fact that we will be able to "swap out" different random number generators for our statistics classes for reliability, extensibility and efficiency

A more elaborate class definition

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/statistics.h

## 3.20 RandomNumberGenerator Class Reference

### Public Member Functions

- **RandomNumberGenerator** (unsigned long \_num\_draws, unsigned long \_init\_seed)
- virtual unsigned long **get\_random\_seed** () const
- virtual void **set\_random\_seed** (unsigned long \_seed)
- virtual void **set\_num\_draws** (unsigned long \_num\_draws)
- virtual unsigned long **get\_random\_integer** ()=0

### Protected Attributes

- unsigned long **init\_seed**
- unsigned long **cur\_seed**
- unsigned long **num\_draws**

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/random.hpp

## 3.21 Stack< T, CONT > Class Template Reference

### Public Member Functions

- void **push** (T const &)
- void **pop** ()
- T **top** () const
- bool **empty** () const

The documentation for this class was generated from the following file:

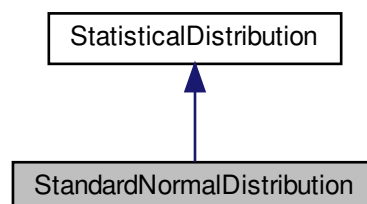
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp\_templates.hpp

## 3.22 StandardNormalDistribution Class Reference

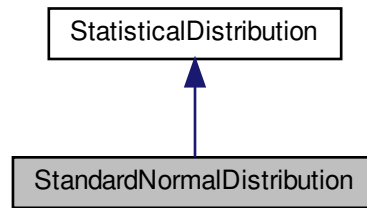
Standard Normal Distribution Implementation.

```
#include <statistics.h>
```

Inheritance diagram for StandardNormalDistribution:



Collaboration diagram for StandardNormalDistribution:



## Public Member Functions

- virtual double **pdf** (const double &x) const
- virtual double **cdf** (const double &x) const
- virtual double **inv\_cdf** (const double &quantile) const
- virtual double **mean** () const
- virtual double **var** () const  
*Equal to 0.*
- virtual double **stddev** () const  
*Equal to 1.*
- virtual void **random\_draws** (const std::vector< double > &uniform\_draws, std::vector< double > &dist\_↔ draws)  
*Variable 1.*

### 3.22.1 Detailed Description

Standard Normal Distribution Implementation.

A more elaborate explanation here

The documentation for this class was generated from the following files:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/statistics.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/statistics.cxx

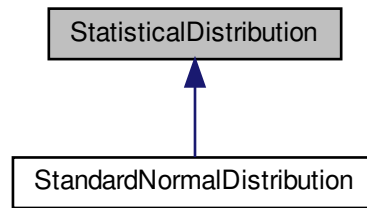
## 3.23 StatisticalDistribution Class Reference

Statistical Distribution Class.

```
#include <statistics.h>
```



Inheritance diagram for StatisticalDistribution:



## Public Member Functions

- [StatisticalDistribution](#) ()  
*A constructor.*
- virtual [~StatisticalDistribution](#) ()  
*Virtual destructor.*
- virtual double **pdf** (const double &x) const =0
- virtual double **cdf** (const double &x) const =0
- virtual double **inv\_cdf** (const double &quantile) const =0
- virtual double **mean** () const =0
- virtual double **var** () const =0  
*Variable 1.*
- virtual double **stdev** () const =0  
*Variable 2.*
- virtual void **random\_draws** (const std::vector< double > &uniform\_draws, std::vector< double > &dist\_← draws)=0  
*Variable 3.*

### 3.23.1 Detailed Description

Statistical Distribution Class.

We've specified pure virtual methods for the probability density function (pdf), cumulative density function (cdf), inverse cdf (inv\_cdf), as well as descriptive statistics functions such as as mean, var (variance) and stdev.

Finally, we have a method that takes in a vector of uniform random variables on the open interval (0,1), then fills a vector of identical length with draws from the distribution

### 3.23.2 Constructor & Destructor Documentation

### 3.23.2.1 StatisticalDistribution()

```
StatisticalDistribution::StatisticalDistribution ( )
```

A constructor.

Statistical Distribution.

Statistical Distribution constructor

A more elaborate class definition

### 3.23.2.2 ~StatisticalDistribution()

```
StatisticalDistribution::~~StatisticalDistribution ( ) [virtual]
```

Virtual destructor.

Constructor.

A more elaborate explanation here

The documentation for this class was generated from the following files:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/statistics.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/statistics.cxx

## 3.24 Str Class Reference

A constructor with a character pointer input.

```
#include <openmp2.hpp>
```

### Public Types

- typedef [Vec](#)< char >::size\_type **size\_type**
- typedef [Vec](#)< char >::size\_type **size\_type**

### Public Member Functions

- **Str** (size\_type n, char c)
- **Str** (const char \*cp)
- template<class In >  
  **Str** (In b, In e)
- [Str](#) ()
- **Str** (size\_type n, char c)
- [Str](#) (const char \*cp)  
  *A constructor with a character pointer input.*
- template<class In >  
  [Str](#) (In b, In e)  
  *create a [Str](#) from the range denoted by iterators b and e*

### 3.24.1 Detailed Description

A constructor with a character pointer input.

Custom [Str](#) class

A numeric progression is a sequence of numbers, where the value of each number depends on one or more of the previous value.

Objects of built-in types generally behave like values: Whenever we copy an object of such a type, the original and copy have the same value but are otherwise independent.

For most of the built-in types, the language also defines a rich set of operators and provides automatic conversions between logically similar types. For example, if we add an int and a double, the compiler automatically converts the int into a double

When we define our own classes, we control the extent to which the resulting objects behave like values. By defining copying and assigning appropriately, the class author can arrange for objects of that class to act like values - that is, the class author can arrange for each object to have state that is independent of any other object.

Our [Vec](#) and [Student\\_info](#) classes are examples of types that act like values

We shall see that the class author can also control conversions and related operations on class objects, thereby providing classes whose objects behave even more similarly to objects of built-in types.

Defining a [Str](#) class that lets us create objects that behave approximately as we would like.

### 3.24.2 Constructor & Destructor Documentation

#### 3.24.2.1 [Str\(\)](#) [1/2]

```
Str::Str ( ) [inline]
```

default constructor, create an empty str

#### 3.24.2.2 [Str\(\)](#) [2/2]

```
Str::Str (
    const char * cp ) [inline]
```

A constructor with a character pointer input.

create a [Str](#) containing n copies of c

Copy constructor

The documentation for this class was generated from the following files:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/openmp2.cxx
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp2.hpp

### 3.25 Student\_info Class Reference

#### Public Member Functions

- **Student\_info** (std::istream &is)
- **Student\_info** (const [Student\\_info](#) &)
- [Student\\_info](#) & **operator=** (const [Student\\_info](#) &)
- std::istream & **read** (std::istream &)
- std::string **name** () const
- double **grade** () const

#### Static Public Member Functions

- static bool **compare** (const [Student\\_info](#) &s1, const [Student\\_info](#) &s2)

The documentation for this class was generated from the following files:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/Student\_info.h
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/src/Student\_info.cxx

### 3.26 TemplateUnderTest< T > Class Template Reference

#### Public Member Functions

- **TemplateUnderTest** (T \*t)
- void **SomeMethod** ()

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp2.hpp

### 3.27 Trap Class Reference

#### Public Member Functions

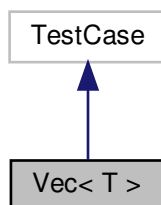
- void **read** ()
- void **computeTrapezium** ()

The documentation for this class was generated from the following file:

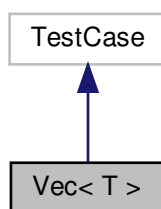
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/trapezoid.hpp

## 3.28 Vec< T > Class Template Reference

Inheritance diagram for Vec< T >:



Collaboration diagram for Vec< T >:



### Public Types

- typedef T \* **iterator**
- typedef const T \* **const\_iterator**
- typedef size\_t **size\_type**
- typedef T \* **iterator**
- typedef const T \* **const\_iterator**
- typedef size\_t **size\_type**
- typedef T **value\_type**
- typedef T & **reference**
- typedef const T & **const\_reference**

### Public Member Functions

- **Vec** (size\_type n, const T &t=T())
- **Vec** (const Vec &v)
- Vec & **operator=** (const Vec &)

- const T & **operator[]** (size\_type i) const
- void **push\_back** (const T &t)
- size\_type **size** () const
- iterator **begin** ()
- const\_iterator **begin** () const
- iterator **end** ()
- const\_iterator **end** () const
- void **runTest** ()

The documentation for this class was generated from the following files:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/MPI\_str.hpp
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/openmp1.hpp

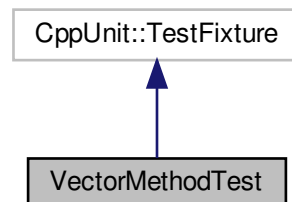
### 3.29 tutorial::Vec< T > Class Template Reference

The documentation for this class was generated from the following file:

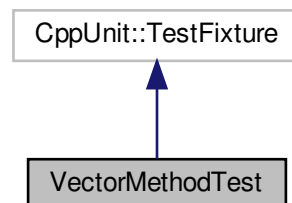
- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/lib\_mpi.hpp

### 3.30 VectorMethodTest Class Reference

Inheritance diagram for VectorMethodTest:



Collaboration diagram for VectorMethodTest:



### Public Member Functions

- void **setUp** ()
- void **tearDown** ()
- void **testConstructor** ()

The documentation for this class was generated from the following file:

- /home/oohnohnoh1/Desktop/GIT/Research/Parallel/include/new.hpp





# Index

- ~MPIInput
  - MPIInput, [13](#)
- ~StatisticalDistribution
  - StatisticalDistribution, [22](#)
- ArithProgression, [5](#)
- Audit, [6](#)
- background\_task, [7](#)
- Complex, [7](#)
- Core, [8](#)
- Genfun::Argument, [5](#)
- getDataPack
  - MPIInput, [13](#)
- Grad, [9](#)
- InitiateVectorMethod< ItemType >, [10](#)
- MPI\_BC\_Generic< T, Q, R >, [11](#)
- MPI\_BC, [10](#)
  - MPI\_BC, [11](#)
- MPI\_sorting\_methods, [12](#)
- MPIInput, [12](#)
  - ~MPIInput, [13](#)
  - getDataPack, [13](#)
  - MPIInput, [13](#)
- OMP< T >, [14](#)
- part1::Point, [16](#)
- Partstruct, [15](#)
- PassFail, [15](#)
- ProbDist, [16](#)
- Progression, [17](#)
- QTstyle\_Test, [18](#)
- RandomNumberGenerator, [18](#)
- Stack< T, CONT >, [19](#)
- StandardNormalDistribution, [19](#)
- StatisticalDistribution, [20](#)
  - ~StatisticalDistribution, [22](#)
  - StatisticalDistribution, [21](#)
- Str, [22](#)
  - Str, [23](#)
- Student\_info, [24](#)
- TemplateUnderTest< T >, [24](#)
- Trap, [24](#)
- tutorial::Vec< T >, [26](#)
- Vec< T >, [25](#)
- VectorMethodTest, [26](#)