Optic Materials

1.0.0

Generated by Doxygen 1.9.8

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

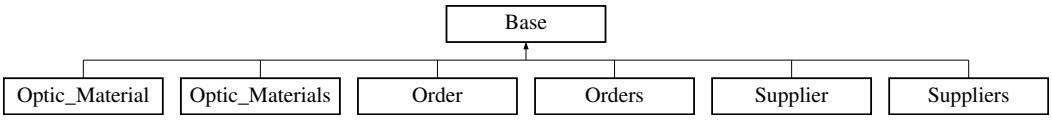
Class Documentation

4.1 Base Class Reference

A base class providing an interface for serialization and deserialization as well as stream insertion and extraction.

`#include <base.h>`

Inheritance diagram for Base:



Friends

- ostream & operator<< (ostream &output, const Base &base)

  Overloads the << operator for outputting the Base object's details to an output stream.
- istream & operator>> (istream &input, Base &base)

  Overloads the >> operator for reading data into a Base object from an input stream.

4.1.1 Detailed Description

A base class providing an interface for serialization and deserialization as well as stream insertion and extraction.

This class defines a common interface for derived classes to implement custom behavior for printing to and reading from streams, and for converting to and from JSON format.

4.1.2 Friends And Related Symbol Documentation

4.1.2.1 operator<<

```
ostream & operator<< (
            ostream & output,
            const Base & base )  [friend]
```

Overloads the << operator for outputting the Base object's details to an output stream.

Parameters

output	The output stream to which the data is to be written.
base	The Base object whose data is to be outputted.

Returns

Returns the modified output stream for chaining.

4.1.2.2 operator>>

```
istream & operator>> (
            istream & input,
            Base & base )  [friend]
```

Overloads the >> operator for reading data into a Base object from an input stream.

Parameters

input	The input stream from which the data is to be read.
base	The Base object where the read data is to be stored.

Returns

Returns the modified input stream for chaining.

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

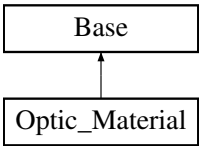
- base.h

4.2 Optic_Material Class Reference

Class representing an optical material.

```
#include <optic_material.h>
```

Inheritance diagram for Optic_Material:



Public Member Functions

- **Optic_Material** ()

  Default constructor for Optic_Material.
- Optic_Material (string type, double width, double diopter, string name, double price)

  Constructor for Optic_Material with parameters.
- string getType () const

  Gets the type of the optical material.
- void setType (string type)

  Sets the type of the optical material.
- double getWidth () const

  Gets the width of the optical material.
- void setWidth (double width)

  Sets the width of the optical material.
- double getDiopter () const

  Gets the diopter value of the optical material.
- void setDiopter (double diopter)

  Sets the diopter value of the optical material.
- string getName () const

  Gets the name of the optical material.
- void setName (string name)

  Sets the name of the optical material.
- double getPrice () const

  Gets the price of the optical material.
- void setPrice (double price)

  Sets the price of the optical material.
- ostream & print (ostream &output) const override

  Overridden method for printing the optical material's details to an output stream.
- istream & input (istream &input) override

  Overridden method for reading data into the optical material from an input stream.
- void to_json (json &j) const override

  Overridden method for converting the optical material's state to JSON format.
- void from_json (json &j) override

  Overridden method for setting the optical material's state from JSON format.

4.2.1 Detailed Description

Class representing an optical material.

This class stores details of an optical material and provides methods for setting and getting its properties. It extends the Base class, providing implementations for serialization and deserialization to/from streams and JSON.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Optic_Material()

```
Optic_Material::Optic_Material (
            string type,
            double width,
            double diopter,
            string name,
            double price )
```

Constructor for Optic_Material with parameters.

Parameters

type	The type of the optical material.
width	The width of the optical material.
diopter	The diopter value of the optical material.
name	The name of the optical material.
price	The price of the optical material.

4.2.3 Member Function Documentation

4.2.3.1 from_json()

```
void Optic_Material::from_json (
            json & j )  [override], [virtual]
```

Overridden method for setting the optical material's state from JSON format.

Parameters

j	JSON object from which the object's state is to be read.

Implements Base.

4.2.3.2 getDiopter()

```
double Optic_Material::getDiopter ( ) const
```

Gets the diopter value of the optical material.

Returns

The diopter value of the optical material.

4.2.3.3 getName()

```
std::string Optic_Material::getName ( ) const
```

Gets the name of the optical material.

Returns

The name of the optical material.

4.2.3.4 getPrice()

```
double Optic_Material::getPrice ( ) const
```

Gets the price of the optical material.

Returns

The price of the optical material.

4.2.3.5 getType()

```
std::string Optic_Material::getType ( ) const
```

Gets the type of the optical material.

Returns

The type of the optical material.

4.2.3.6 getWidth()

```
double Optic_Material::getWidth ( ) const
```

Gets the width of the optical material.

Returns

The width of the optical material.

4.2.3.7 input()

```
istream & Optic_Material::input (
            istream & input )  [override], [virtual]
```

Overridden method for reading data into the optical material from an input stream.

Parameters

input	The input stream from which the data is to be read.

Returns

Reference to the modified input stream.

Implements Base.

4.2.3.8 print()

```
ostream & Optic_Material::print (
            ostream & output ) const  [override], [virtual]
```

Overridden method for printing the optical material's details to an output stream.

Parameters

output	The output stream to which the data is to be written.

Returns

Reference to the modified output stream.

Implements Base.

4.2.3.9 setDiopter()

```
void Optic_Material::setDiopter (
            double diopter )
```

Sets the diopter value of the optical material.

Parameters

diopter	The diopter value of the optical material.

4.2.3.10 setName()

```
void Optic_Material::setName (
            string name )
```

Sets the name of the optical material.

Parameters

name	The name of the optical material.

4.2.3.11 setPrice()

```
void Optic_Material::setPrice (
            double price )
```

Sets the price of the optical material.

Parameters

price	The price of the optical material.

4.2.3.12 setType()

```
void Optic_Material::setType (
            string type )
```

Sets the type of the optical material.

Parameters

type	The type of the optical material.

4.2.3.13 setWidth()

```
void Optic_Material::setWidth (
            double width )
```

Sets the width of the optical material.

Parameters

width	The width of the optical material.

4.2.3.14 to_json()

```
void Optic_Material::to_json (
            json & j ) const  [override], [virtual]
```

Overridden method for converting the optical material's state to JSON format.

Parameters

j	JSON object to which the object's state is to be written.

Implements Base.

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

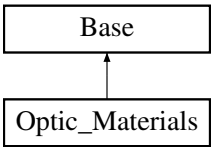
- optic_material.h
- optic_material.cpp

4.3 Optic_Materials Class Reference

Class representing a collection of Optic_Material objects.

`#include <optic_materials.h>`

Inheritance diagram for Optic_Materials:



Public Member Functions

- **Optic_Materials** ()=default
  Default constructor for Optic_Materials.
- int getSize () const
  Gets the number of optic materials in the collection.
- void addOpticMaterial (const Optic_Material &optic_material)
  Adds an Optic_Material object to the collection.
- vector< Optic_Material > getOpticMaterials ()
  Retrieves all optic materials in the collection.
- Optic_Material getOpticMaterialByIndex (int index) const
  Retrieves an Optic_Material object at a specified index in the collection.
- ostream & print (ostream &output) const override
  Overridden method for printing the collection's details to an output stream.
- void **print_on_one_line** () const
  Prints a concise one-line description of each optic material in the collection.
- istream & input (istream &input) override
  Overridden method for reading data into the collection from an input stream.
- void to_json (json &j) const override
  Overridden method for converting the collection's state to JSON format.
- void from_json (json &j) override
  Overridden method for setting the collection's state from JSON format.

4.3.1 Detailed Description

Class representing a collection of Optic_Material objects.

This class encapsulates a collection of Optic_Material objects and provides methods for managing this collection, including adding and retrieving optic materials, and implementing serialization and deserialization to/from streams and JSON.

4.3.2 Member Function Documentation

4.3.2.1 addOpticMaterial()

```
void Optic_Materials::addOpticMaterial (
            const Optic_Material & optic_material )
```

Adds an Optic_Material object to the collection.

Parameters

optic_material	The Optic_Material object to be added.

4.3.2.2 from_json()

```
void Optic_Materials::from_json (
            json & j )  [override], [virtual]
```

Overridden method for setting the collection's state from JSON format.

Parameters

j	JSON object from which the collection's state is to be read.

Implements Base.

4.3.2.3 getOpticMaterialByIndex()

```
Optic_Material Optic_Materials::getOpticMaterialByIndex (
            int index ) const
```

Retrieves an Optic_Material object at a specified index in the collection.

Parameters

index	The index of the Optic_Material object to retrieve.

Returns

The Optic_Material object at the specified index.

4.3.2.4 getOpticMaterials()

```
vector< Optic_Material > Optic_Materials::getOpticMaterials ( )
```

Retrieves all optic materials in the collection.

Returns

A vector containing all Optic_Material objects in the collection.

4.3.2.5 getSize()

```
int Optic_Materials::getSize ( ) const
```

Gets the number of optic materials in the collection.

Returns

The size of the optic materials collection.

4.3.2.6 input()

```
istream & Optic_Materials::input (
            istream & input )  [override], [virtual]
```

Overridden method for reading data into the collection from an input stream.

Parameters

input	The input stream from which the data is to be read.

Returns

Reference to the modified input stream.

Implements Base.

4.3.2.7 print()

```
ostream & Optic_Materials::print (
            ostream & output ) const  [override], [virtual]
```

Overridden method for printing the collection's details to an output stream.

Parameters

output	The output stream to which the collection's data is to be written.

Returns

Reference to the modified output stream.

Implements Base.

4.3.2.8 to_json()

```
void Optic_Materials::to_json (
            json & j ) const  [override], [virtual]
```

Overridden method for converting the collection's state to JSON format.

Parameters

j	JSON object to which the collection's state is to be written.

Implements Base.

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

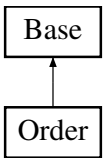
- optic_materials.h
- optic_materials.cpp

4.4 Order Class Reference

Class representing an order which includes optic materials and a supplier.

`#include <order.h>`

Inheritance diagram for Order:



Public Member Functions

- **Order** ()

  Default constructor for Order.
- Order (int id, vector< Optic_Material > materials, Supplier supplier)

  Constructor for Order with parameters.
- void setId (int id)

  Sets the unique identifier of the order.
- int getId ()

  Gets the unique identifier of the order.
- vector< Optic_Material > getMaterials ()

  Retrieves all optic materials included in the order.
- void addMaterial (const Optic_Material &material)

  Adds an optic material to the order.
- void addSupplier (const Supplier &supplier)

  Sets the supplier for the order.
- Supplier getSupplier ()

  Gets the supplier associated with the order.
- double getTotalRaw ()

  Calculates the total cost of the order without profit margin.
- double getTotal ()

  Calculates the total cost of the order including the supplier's profit margin.
- ostream & print (ostream &output) const override

  Overridden method for printing the order's details to an output stream.
- istream & input (istream &input) override

  Overridden method for reading data into the order from an input stream.
- void to_json (json &j) const override

  Overridden method for converting the order's state to JSON format.
- void from_json (json &j) override

  Overridden method for setting the order's state from JSON format.

4.4.1 Detailed Description

Class representing an order which includes optic materials and a supplier.

This class encapsulates the details of an order, including a list of optic materials, a supplier, and methods for managing and calculating the total cost of the order. It extends the Base class, providing implementations for serialization and deserialization to/from streams and JSON.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Order()

```
Order::Order (
            int id,
            vector< Optic_Material > materials,
            Supplier supplier )
```

Constructor for Order with parameters.

Parameters

id	The unique identifier for the order.
materials	A vector of Optic_Material objects included in the order.
supplier	The Supplier associated with the order.

4.4.3 Member Function Documentation

4.4.3.1 addMaterial()

```
void Order::addMaterial (
            const Optic_Material & material )
```

Adds an optic material to the order.

Parameters

material	The Optic_Material object to be added to the order.

4.4.3.2 addSupplier()

```
void Order::addSupplier (
            const Supplier & supplier )
```

Sets the supplier for the order.

Parameters

supplier	The Supplier to be associated with the order.

4.4.3.3 from_json()

```
void Order::from_json (
            json & j )  [override], [virtual]
```

Overridden method for setting the order's state from JSON format.

Parameters

j	JSON object from which the order's state is to be read.

Implements Base.

4.4.3.4 getId()

```
int Order::getId ( )
```

Gets the unique identifier of the order.

Returns

The unique identifier of the order.

4.4.3.5 getMaterials()

```
vector< Optic_Material > Order::getMaterials ( )
```

Retrieves all optic materials included in the order.

Returns

A vector of Optic_Material objects in the order.

4.4.3.6 getSupplier()

```
Supplier Order::getSupplier ( )
```

Gets the supplier associated with the order.

Returns

The Supplier associated with the order.

4.4.3.7 getTotal()

```
double Order::getTotal ( )
```

Calculates the total cost of the order including the supplier's profit margin.

Returns

The total cost of the order.

4.4.3.8 getTotalRaw()

```
double Order::getTotalRaw ( )
```

Calculates the total cost of the order without profit margin.

Returns

The total raw cost of the order.

4.4.3.9 input()

```
istream & Order::input (
            istream & input )  [override], [virtual]
```

Overridden method for reading data into the order from an input stream.

Parameters

input	The input stream from which the order data is to be read.

Returns

Reference to the modified input stream.

Implements Base.

4.4.3.10 print()

```
ostream & Order::print (
            ostream & output ) const  [override], [virtual]
```

Overridden method for printing the order's details to an output stream.

Parameters

output	The output stream to which the order's data is to be written.

Returns

Reference to the modified output stream.

Implements Base.

4.4.3.11 setId()

```
void Order::setId (
            int id )
```

Sets the unique identifier of the order.

Parameters

id	The unique identifier for the order.

4.4.3.12 to_json()

```
void Order::to_json (
            json & j ) const  [override], [virtual]
```

Overridden method for converting the order's state to JSON format.

Parameters

j	JSON object to which the order's state is to be written.

Implements Base.

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

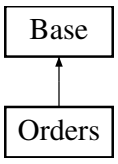
- order.h
- order.cpp

4.5 Orders Class Reference

Class representing a collection of Order objects.

```
#include <orders.h>
```

Inheritance diagram for Orders:



Public Member Functions

- **Orders** ()

  Default constructor for Orders.
- void addOrder (const Order &order)

  Adds an Order object to the collection.
- vector< Order > getOrders ()

  Retrieves all orders in the collection.
- void addMaterialToLastOrder (const Optic_Material &material)

  Adds an optic material to the last order in the collection.
- void addSupplierToLastOrder (const Supplier &supplier)

  Adds a supplier to the last order in the collection.
- void addIdToLastOrder (int id)

  Sets the ID for the last order in the collection.
- void **printOrderTotal** ()

  Prints the total cost of each order in the collection.
- ostream & print (ostream &output) const override

  Overridden method for printing the collection's details to an output stream.
- istream & input (istream &input) override

  Overridden method for reading data into the collection from an input stream.
- void to_json (json &j) const override

  Overridden method for converting the collection's state to JSON format.
- void from_json (json &j) override

  Overridden method for setting the collection's state from JSON format.

4.5.1 Detailed Description

Class representing a collection of Order objects.

This class manages a collection of Order objects and provides methods for adding orders and managing their contents. It extends the Base class, implementing serialization and deserialization to/from streams and JSON.

4.5.2 Member Function Documentation

4.5.2.1 addIdToLastOrder()

```
void Orders::addIdToLastOrder (
            int id )
```

Sets the ID for the last order in the collection.

Parameters

id	The unique identifier for the last order.

4.5.2.2 addMaterialToLastOrder()

```
void Orders::addMaterialToLastOrder (
```

```
            const Optic_Material & material )
```

Adds an optic material to the last order in the collection.

Parameters

material	The Optic_Material object to be added to the last order.

4.5.2.3 addOrder()

```
void Orders::addOrder (
            const Order & order )
```

Adds an Order object to the collection.

Parameters

order	The Order object to be added.

4.5.2.4 addSupplierToLastOrder()

```
void Orders::addSupplierToLastOrder (
            const Supplier & supplier )
```

Adds a supplier to the last order in the collection.

Parameters

supplier	The Supplier to be associated with the last order.

4.5.2.5 from_json()

```
void Orders::from_json (
            json & j )  [override], [virtual]
```

Overridden method for setting the collection's state from JSON format.

Parameters

j	JSON object from which the collection's state is to be read.

Implements Base.

4.5.2.6 getOrders()

```
vector< Order > Orders::getOrders ( )
```

Retrieves all orders in the collection.

Returns

A vector containing all Order objects in the collection.

4.5.2.7 input()

```
istream & Orders::input (
            istream & input )  [override], [virtual]
```

Overridden method for reading data into the collection from an input stream.

Parameters

input	The input stream from which the data is to be read.

Returns

Reference to the modified input stream.

Implements Base.

4.5.2.8 print()

```
ostream & Orders::print (
            ostream & output ) const  [override], [virtual]
```

Overridden method for printing the collection's details to an output stream.

Parameters

output	The output stream to which the collection's data is to be written.

Returns

Reference to the modified output stream.

Implements Base.

4.5.2.9 to_json()

```
void Orders::to_json (
            json & j ) const  [override], [virtual]
```

Overridden method for converting the collection's state to JSON format.

Parameters

j	JSON object to which the collection's state is to be written.

Implements Base.

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

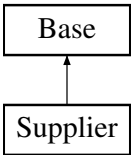
- orders.h
- orders.cpp

4.6 Supplier Class Reference

Class representing a supplier in the optics materials system.

`#include <supplier.h>`

Inheritance diagram for Supplier:



Public Member Functions

- **Supplier** ()
  Default constructor for Supplier.
- Supplier (std::string bulstat, std::string name, std::string location, std::string phone, double profit_margin)
  Constructor for Supplier with parameters.
- std::string getBulstat () const
  Gets the unique identifier of the supplier.
- void setBulstat (std::string bulstat)
  Sets the unique identifier of the supplier.
- std::string getName () const
  Gets the name of the supplier.
- void setName (std::string name)
  Sets the name of the supplier.
- std::string getLocation () const
  Gets the location of the supplier.
- void setLocation (std::string location)
  Sets the location of the supplier.
- std::string getPhone () const
  Gets the contact phone number of the supplier.
- void setPhone (std::string phone)
  Sets the contact phone number of the supplier.
- double getProfitMargin () const
  Gets the profit margin of the supplier.
- void setProfitMargin (double profit_margin)
  Sets the profit margin for the supplier.
- ostream & print (ostream &output) const override
  Overridden method for printing the supplier's details to an output stream.
- istream & input (istream &input) override
  Overridden method for reading data into the supplier from an input stream.
- void to_json (nlohmann::json &j) const override
  Overridden method for converting the supplier's state to JSON format.
- void from_json (json &j) override
  Overridden method for setting the supplier's state from JSON format.

4.6.1 Detailed Description

Class representing a supplier in the optics materials system.

This class stores details about a supplier, including identification, contact information, and profit margin. It extends the Base class, providing implementations for serialization and deserialization to/from streams and JSON.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 Supplier()

```
Supplier::Supplier (
            std::string bulstat,
            std::string name,
            std::string location,
            std::string phone,
            double profit_margin )
```

Constructor for Supplier with parameters.

Parameters

bulstat	The unique identifier for the supplier.
name	The name of the supplier.
location	The location of the supplier.
phone	The contact phone number of the supplier.
profit_margin	The profit margin of the supplier.

4.6.3 Member Function Documentation

4.6.3.1 from_json()

```
void Supplier::from_json (
            json & j )  [override], [virtual]
```

Overridden method for setting the supplier's state from JSON format.

Parameters

j	JSON object from which the supplier's state is to be read.

Implements Base.

4.6.3.2 getBulstat()

```
std::string Supplier::getBulstat ( ) const
```

Gets the unique identifier of the supplier.

Returns

The bulstat of the supplier.

4.6.3.3 getLocation()

```
std::string Supplier::getLocation ( ) const
```

Gets the location of the supplier.

Returns

The location of the supplier.

4.6.3.4 getName()

```
std::string Supplier::getName ( ) const
```

Gets the name of the supplier.

Returns

The name of the supplier.

4.6.3.5 getPhone()

```
std::string Supplier::getPhone ( ) const
```

Gets the contact phone number of the supplier.

Returns

The phone number of the supplier.

4.6.3.6 getProfitMargin()

```
double Supplier::getProfitMargin ( ) const
```

Gets the profit margin of the supplier.

Returns

The profit margin of the supplier.

4.6.3.7 input()

```
istream & Supplier::input (
            istream & input )  [override], [virtual]
```

Overridden method for reading data into the supplier from an input stream.

Parameters

input	The input stream from which the supplier data is to be read.

Returns

Reference to the modified input stream.

Implements Base.

4.6.3.8 print()

```
ostream & Supplier::print (
            ostream & output ) const  [override], [virtual]
```

Overridden method for printing the supplier's details to an output stream.

Parameters

output	The output stream to which the supplier's data is to be written.

Returns

Reference to the modified output stream.

Implements Base.

4.6.3.9 setBulstat()

```
void Supplier::setBulstat (
            std::string bulstat )
```

Sets the unique identifier of the supplier.

Parameters

bulstat	The bulstat to set for the supplier.

4.6.3.10 setLocation()

```
void Supplier::setLocation (
            std::string location )
```

Sets the location of the supplier.

Parameters

location	The location to set for the supplier.

4.6.3.11 setName()

```
void Supplier::setName (
            std::string name )
```

Sets the name of the supplier.

Parameters

name	The name to set for the supplier.

4.6.3.12 setPhone()

```
void Supplier::setPhone (
            std::string phone )
```

Sets the contact phone number of the supplier.

Parameters

phone	The phone number to set for the supplier.

4.6.3.13 setProfitMargin()

```
void Supplier::setProfitMargin (
            double profit_margin )
```

Sets the profit margin for the supplier.

Parameters

profit_margin	The profit margin to set for the supplier.

4.6.3.14 to_json()

```
void Supplier::to_json (
            nlohmann::json & j ) const  [override], [virtual]
```

Overridden method for converting the supplier's state to JSON format.

Parameters

j	JSON object to which the supplier's state is to be written.

Implements Base.

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

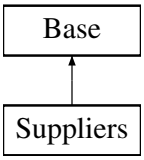
- supplier.h
- supplier.cpp

4.7 Suppliers Class Reference

Class representing a collection of Supplier objects.

`#include <suppliers.h>`

Inheritance diagram for Suppliers:



Public Member Functions

- **Suppliers** ()=default

  Default constructor for Suppliers.
- void addSupplier (const Supplier &supplier)

  Adds a Supplier object to the collection.
- vector< Supplier > getSuppliers ()

  Retrieves all suppliers in the collection.
- Supplier getSupplierByIndex (int index) const

  Retrieves a Supplier object at a specified index in the collection.
- int getSize () const

  Gets the number of suppliers in the collection.
- void **print_on_one_line** () const

  Prints a concise one-line description of each supplier in the collection.
- ostream & print (ostream &output) const override

  Overridden method for printing the collection's details to an output stream.
- istream & input (istream &input) override

  Overridden method for reading data into the collection from an input stream.
- void to_json (json &j) const override

  Overridden method for converting the collection's state to JSON format.
- void from_json (json &j) override

  Overridden method for setting the collection's state from JSON format.

4.7.1 Detailed Description

Class representing a collection of Supplier objects.

This class manages a collection of Supplier objects and provides methods for adding suppliers and accessing them. It extends the Base class, implementing serialization and deserialization to/from streams and JSON.

4.7.2 Member Function Documentation

4.7.2.1 addSupplier()

```
void Suppliers::addSupplier (
            const Supplier & supplier )
```

Adds a Supplier object to the collection.

Parameters

supplier	The Supplier object to be added.

4.7.2.2 from_json()

```
void Suppliers::from_json (
            json & j )  [override], [virtual]
```

Overridden method for setting the collection's state from JSON format.

Parameters

j	JSON object from which the collection's state is to be read.

Implements Base.

4.7.2.3 getSize()

```
int Suppliers::getSize ( ) const
```

Gets the number of suppliers in the collection.

Returns

The size of the suppliers collection.

4.7.2.4 getSupplierByIndex()

```
Supplier Suppliers::getSupplierByIndex (
            int index ) const
```

Retrieves a Supplier object at a specified index in the collection.

Parameters

index	The index of the Supplier object to retrieve.

Returns

The Supplier object at the specified index.

4.7.2.5 getSuppliers()

```
vector< Supplier > Suppliers::getSuppliers ( )
```

Retrieves all suppliers in the collection.

Returns

A vector containing all Supplier objects in the collection.

4.7.2.6 input()

```
istream & Suppliers::input (
            istream & input )  [override], [virtual]
```

Overridden method for reading data into the collection from an input stream.

Parameters

input	The input stream from which the data is to be read.

Returns

Reference to the modified input stream.

Implements Base.

4.7.2.7 print()

```
ostream & Suppliers::print (
            ostream & output ) const  [override], [virtual]
```

Overridden method for printing the collection's details to an output stream.

Parameters

output	The output stream to which the collection's data is to be written.

Returns

Reference to the modified output stream.

Implements Base.

4.7.2.8 to_json()

```
void Suppliers::to_json (
            json & j ) const  [override], [virtual]
```

Overridden method for converting the collection's state to JSON format.

Parameters

j	JSON object to which the collection's state is to be written.

Implements Base.

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

- suppliers.h
- suppliers.cpp

Chapter 5

File Documentation

5.1 base.h

```
00001 #ifndef BASE_H
00002 #define BASE_H
00003
00004 #include <iostream>
00005 #include <nlohmann/json.hpp>
00006
00007 using json = nlohmann::json;
00008 using namespace std;
00009
00018 class Base
00019 {
00020 private:
00026     virtual ostream& print(ostream& output) const = 0;
00027
00033     virtual istream& input(istream& input) = 0;
00034
00039     virtual void to_json(json& j) const = 0;
00040
00045     virtual void from_json(json& j) = 0;
00046
00047 public:
00054     friend ostream& operator<<(ostream& output, const Base& base);
00055
00062     friend istream& operator>>(istream& input, Base& base);
00063 };
00064
00065 #endif
00066
```

5.2 optic_material.h

```
00001 #ifndef OPTIC_MATERIAL_H
00002 #define OPTIC_MATERIAL_H
00003
00004 #include <iostream>
00005 #include <string>
00006 #include "base.h"
00007
00008 using namespace std;
00009
00018 class Optic_Material : public Base
00019 {
00020 private:
00021     string type;
00022     double width;
00023     double diopter;
00024     string name;
00025     double price;
00026
00027 public:
00031     Optic_Material();
00032
00041     Optic_Material(string type, double width, double diopter, string name, double price);
00042
```

```
00047     string getType() const;
00048
00053     void setType(string type);
00054
00059     double getWidth()const;
00060
00065     void setWidth(double width);
00066
00071     double getDiopter()const;
00072
00077     void setDiopter(double diopter);
00078
00083     string getName()const;
00084
00089     void setName(string name);
00090
00095     double getPrice()const;
00096
00101     void setPrice(double price);
00102
00108     ostream& print(ostream& output) const override;
00109
00115     istream& input(istream& input) override;
00116
00121     void to_json(json& j) const override;
00122
00127     void from_json(json& j) override;
00128 };
00129
00130 #endif
```

5.3 optic_materials.h

```
00001 #pragma once
00002 #ifndef OPTIC_MATERIALS_H
00003 #define OPTIC_MATERIALS_H
00004
00005 #include <vector>
00006 #include <fstream>
00007 #include "optic_material.h"
00008 #include "base.h"
00009 #include <nlohmann/json.hpp>
00010
00011 using json = nlohmann::json;
00012
00020 class Optic_Materials : public Base
00021 {
00022 private:
00023     vector<Optic_Material> optic_materials;
00024
00025 public:
00029     Optic_Materials() = default;
00030
00035     int getSize() const;
00036
00041     void addOpticMaterial(const Optic_Material& optic_material);
00042
00047     vector<Optic_Material> getOpticMaterials();
00048
00054     Optic_Material getOpticMaterialByIndex(int index) const;
00055
00061     ostream& print(ostream& output) const override;
00062
00066     void print_on_one_line() const;
00067
00073     istream& input(istream& input) override;
00074
00079     void to_json(json& j) const override;
00080
00085     void from_json(json& j) override;
00086 };
00087
00088 #endif
```

5.4 order.h

```
00001 #ifndef ORDER_H
00002 #define ORDER_H
00003
```

```
00004 #include <iostream>
00005 #include <vector>
00006 #include "supplier.h"
00007 #include "optic_material.h"
00008
00017 class Order : public Base
00018 {
00019 private:
00020     int id;
00021     vector<Optic_Material> materials;
00022     Supplier supplier;
00023
00024 public:
00028     Order();
00029
00036     Order(int id, vector<Optic_Material> materials, Supplier supplier);
00037
00042     void setId(int id);
00043
00048     int getId();
00049
00054     vector<Optic_Material> getMaterials();
00055
00060     void addMaterial(const Optic_Material& material);
00061
00066     void addSupplier(const Supplier& supplier);
00067
00072     Supplier getSupplier();
00073
00078     double getTotalRaw();
00079
00084     double getTotal();
00085
00091     ostream& print(ostream& output) const override;
00092
00098     istream& input(istream& input) override;
00099
00104     void to_json(json& j) const override;
00105
00110     void from_json(json& j) override;
00111 };
00112
00113 #endif
```

5.5 orders.h

```
00001 #ifndef ORDERS_H
00002 #define ORDERS_H
00003
00004 #include <vector>
00005 #include <fstream>
00006 #include "order.h"
00007 #include "base.h"
00008 #include <nlohmann/json.hpp>
00009
00010 using json = nlohmann::json;
00011
00019 class Orders : public Base {
00020 private:
00021     vector<Order> orders;
00022
00023 public:
00027     Orders();
00028
00033     void addOrder(const Order& order);
00034
00039     vector<Order> getOrders();
00040
00045     void addMaterialToLastOrder(const Optic_Material& material);
00046
00051     void addSupplierToLastOrder(const Supplier& supplier);
00052
00057     void addIdToLastOrder(int id);
00058
00062     void printOrderTotal();
00063
00069     ostream& print(ostream& output) const override;
00070
00076     istream& input(istream& input) override;
00077
00082     void to_json(json& j) const override;
00083
00088     void from_json(json& j) override;
```

```
00089 };
00090
00091 #endif
```

5.6 supplier.h

```
00001 #ifndef SUPPLIER_H
00002 #define SUPPLIER_H
00003
00004 #include <iostream>
00005 #include <string>
00006 #include "base.h"
00007
00015 class Supplier : public Base
00016 {
00017 private:
00018     std::string bulstat;
00019     std::string name;
00020     std::string location;
00021     std::string phone;
00022     double profit_margin;
00023
00024 public:
00028     Supplier();
00029
00038     Supplier(std::string bulstat, std::string name, std::string location, std::string phone, double
      profit_margin);
00039
00044     std::string getBulstat() const;
00045
00050     void setBulstat(std::string bulstat);
00051
00056     std::string getName() const;
00057
00062     void setName(std::string name);
00063
00068     std::string getLocation() const;
00069
00074     void setLocation(std::string location);
00075
00080     std::string getPhone() const;
00081
00086     void setPhone(std::string phone);
00087
00092     double getProfitMargin() const;
00093
00098     void setProfitMargin(double profit_margin);
00099
00105     ostream& print(ostream& output) const override;
00106
00112     istream& input(istream& input) override;
00113
00118     void to_json(nlohmann::json& j) const override;
00119
00124     void from_json(json& j) override;
00125 };
00126
00127 #endif
```

5.7 suppliers.h

```
00001 #pragma once
00002 #ifndef SUPPLIERS_H
00003 #define SUPPLIERS_H
00004
00005 #include <vector>
00006 #include <fstream>
00007 #include "supplier.h"
00008 #include "base.h"
00009 #include <nlohmann/json.hpp>
00010
00011 using json = nlohmann::json;
00012
00020 class Suppliers : public Base
00021 {
00022 private:
00023     vector<Supplier> suppliers;
00024
00025 public:
```

```
00029     Suppliers() = default;
00030
00035     void addSupplier(const Supplier& supplier);
00036
00041     vector<Supplier> getSuppliers();
00042
00048     Supplier getSupplierByIndex(int index) const;
00049
00054     int getSize() const;
00055
00059     void print_on_one_line() const;
00060
00066     ostream& print(ostream& output) const override;
00067
00073     istream& input(istream& input) override;
00074
00079     void to_json(json& j) const override;
00080
00085     void from_json(json& j) override;
00086 };
00087
00088 #endif
```