Hotel

Generated by Doxygen 1.9.3

Chapter 1

Hotel Project Documentation

commands

1.1 Overview

This is information system, serving a hotel. The system allows the user to make reservations, receive up-to-date information about the rooms in the hotel, make suggestion which room is most suitable for the guest. The program runs in console mode with several instructions, being printed on startup. The data for the rooms of the hotel is written in file named "h.rooms" in same directory as the executable. Then if data has already been recorded about the hotel (recognized by its name) the data is being read from a binary file called "<hotel name>.dat". On closing, the data about the hotel is being written again.

1.1.1 Example

reserve: 102 15/7/2022 22/7/2022 Ivan; has dog

Reservation successfully made!

rooms:

101: No reservations.
102: Next reservation is from 15/7/2022 to 22/7/2022. (*answer depends on current date*)
103: No reservations.
...

report: 1/3/2022 1/4/2022

report "report-2022-03-01.txt" created successfully.

free: 101

Room is already free!

free: 102

Room freed successfully! (*answer depends on current date*)

request: 4 1/8/2022 7/8/2022

1. Number: <*best room*> Beds: <*best room's beds*> : <available | NOT available> between 1/8/2022 and 7/8/2022

    1. Number: <*second best room*> Beds: <*second best room's beds*> : <available | NOT available> between 1/8/2022 and 7/8/2022

    2. Number: <*third best room*> Beds: <*third best room's beds*> : <available | NOT available> between 1/8/2022 and 7/8/2022

    3. Number: <*fourth best room*> Beds: <*fourth best room's beds*> : <available | NOT available> between 1/8/2022 and 7/8/2022

    4. Number: <*fifth best room*> Beds: <*fifth best room's beds*> : <available | NOT available> between 1/8/2022 and 7/8/2022

maintenance: 103 <today> 20/8/2022 cleaning

Maintenance planned successfully.

rooms:

...
103: Next maintenance is from <today> to 20/8/2022.
...

Chapter 2

User commands

2.1 The UI works with the following commands:

(*They are displayed on startup in the console*)

Note

all dates must be input in format D/M/Y
parameters with [] are mandatory and these with {} are optional

2.2 Adding new reservation

reserve: [Room number] [Accomodation date] [Departure date] {Guest name[;]} {Note}

2.3 Free rooms on particular date

available: [date]

2.4 Free particular room

free: [Room number]

2.5 Create report about the usage of the rooms in particular period

report: [From date] [To date]

2.6 Suggest room for a group of guests and partcular period

request: [minimal number of beds] [Accomodation date] [Departure date]

2.7 Close a room for maintenance

maintenance: [room number] [From date] [To date] [Note]

2.8 See current state of all the rooms

rooms:

2.9 See when a room is free for certain number of days

plan: [Room number] [Number of nights]

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Class Documentation

5.1 Date Class Reference

Class representing date with day, month and year.

`#include <Date.hpp>`

Public Member Functions

- Date (unsigned short d=1, unsigned short m=1, unsigned short y=1900)
  Construct a new Date object from day, month and year.
- bool operator< (const Date other) const
  checks if this Date is chronologically before other Date
- bool operator<= (const Date other) const
  see operator<
- bool operator> (const Date other) const
  checks if this Date is chronologically after other Date
- bool operator>= (const Date other) const
  see operator>
- bool operator== (const Date other) const
  checks if two dates are identical
- const char * operator() (char *buf) const
  records this Date in buffer in format YYYY-MM-DD
- int operator- (Date other) const
- Date & operator++ ()
  overloaded prefix incremention operator for Date
- void writeToBinaryFile (std::ofstream &ofs)
  write the Room data into binary file opened by ofstream
- void readDataFromBinary (std::ifstream &ifs)
  read the Room data from binary file opened by ifstream

Static Public Member Functions

- static Date getToday ()
  Get the today date.

Friends

- std::istream & operator>> (std::istream &is, Date &d)

  overloaded operator for inputing Date
- std::ostream & operator<< (std::ostream &os, const Date &d)

  overloaded operator for outputing Date

5.1.1 Detailed Description

Class representing date with day, month and year.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Date()

```
Date::Date (
            unsigned short d = 1,
            unsigned short m = 1,
            unsigned short y = 1900 )  [inline]
```

Construct a new Date object from day, month and year.

Default Date is 1/1/1900

Parameters

d	day
m	month
y	year

5.1.3 Member Function Documentation

5.1.3.1 getToday()

```
Date Date::getToday ( )  [static]
```

Get the today date.

Returns

Date

5.1.3.2 operator()()

```
const char * Date::operator() (
            char * buf ) const
```

records this Date in buffer in format YYYY-MM-DD

Parameters

buf	buffer where Date is recorded

Returns

const char* pointer to beginning of buf

5.1.3.3 operator++()

```
Date & Date::operator++ ( )
```

overloaded prefix incremention operator for Date

Returns

Date& reference to this Date

5.1.3.4 operator-()

```
int Date::operator- (
            Date other ) const
```

Parameters

other	Date

Returns

int difference between of this Date and other

5.1.3.5 operator<()

```
bool Date::operator< (
            const Date other ) const
```

checks if this Date is chronologically before other Date

Parameters

other	compared Date

Returns

true this Date is chronologically before other

false this Date is not chronologically before other

5.1.3.6 operator<=()

```
bool Date::operator<= (
            const Date other ) const
```

see operator<

5.1.3.7 operator==()

```
bool Date::operator== (
            const Date other ) const
```

checks if two dates are identical

Parameters

other	compared Date

Returns

true the dates are identical

false the dates are not identical

5.1.3.8 operator>()

```
bool Date::operator> (
            const Date other ) const
```

checks if this Date is chronologically after other Date

Parameters

other	compared Date

Returns

true this Date is chronologically after other

false this Date is not chronologically after other

5.1.3.9 operator>=()

```
bool Date::operator>= (
            const Date other ) const
```

see operator>

5.1.3.10 readDataFromBinary()

```
void Date::readDataFromBinary (
            std::ifstream & ifs )
```

read the Room data from binary file opened by ifstream

Parameters

ofs	input stream connected to binary file

5.1.3.11 writeToBinaryFile()

```
void Date::writeToBinaryFile (
            std::ofstream & ofs )
```

write the Room data into binary file opened by ofstream

Parameters

ofs	output stream connected to binary file

5.1.4 Friends And Related Function Documentation

5.1.4.1 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
```

```
const Date & d )  [friend]
```

overloaded operator for outputing Date

Parameters

os	output stream
d	Date to be output

Returns

std::ostream& reference to the output stream

5.1.4.2 operator>>

```
std::istream & operator>> (
            std::istream & is,
            Date & d )  [friend]
```

overloaded operator for inputing Date

Parameters

is	input stream
d	Date to be input

Returns

std::istream& reference to the input stream

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

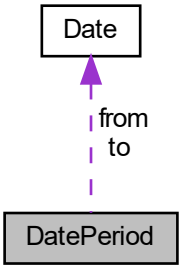
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Date.hpp
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Date.cpp

5.2 DatePeriod Struct Reference

Class containing two dates forming a period of time from Date to Date.

```
#include <Date.hpp>
```

Collaboration diagram for DatePeriod:



Public Member Functions

- unsigned length () const
  distance in days of the period
- DatePeriod & operator++ ()
  moving period one day froward
- void readProper ()
  method to read from stdin a proper period of time (from is before to)

Public Attributes

- Date from
  beginning Date of the period
- Date to
  end Date of the period

5.2.1 Detailed Description

Class containing two dates forming a period of time from Date to Date.

5.2.2 Member Function Documentation

5.2.2.1 length()

```
unsigned DatePeriod::length ( ) const  [inline]
```

distance in days of the period

Returns

unsigned days between beginning and end

5.2.2.2 operator++()

```
DatePeriod & DatePeriod::operator++ ( )
```

moving period one day froward

Returns

DatePeriod& this DatePeriod

5.2.2.3 readProper()

```
void DatePeriod::readProper ( )
```

method to read from stdin a proper period of time (from is before to)

5.2.3 Member Data Documentation

5.2.3.1 from

```
Date DatePeriod::from
```

beginning Date of the period

5.2.3.2 to

```
Date DatePeriod::to
```

end Date of the period

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

- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Date.hpp
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Date.cpp

5.3 Hotel Class Reference

Class representing hotel with name, current Date and a building (list of rooms)

```
#include <Hotel.hpp>
```

Public Member Functions

- Hotel ()=delete
- Hotel (String hotelDataFile)

  Construct a new Hotel object.
- Hotel (const Hotel &)=delete
- Hotel & operator= (const Hotel &)=delete
- ~Hotel ()

  Destroy the Hotel object.
- String getName () const

  get the name of this Hotel
- void nextDay ()

  advance to the nextDay
- bool reserveRoom (unsigned number, const DatePeriod &period, String name="-", String note="None\n")

  makes a new Reservation for particular Room and period with options for name of guest and notes to the Reservation
- Hotel & showAvailableRooms (std::ostream &, Date)

  output to stream all available rooms for a particular Date
- bool freeRoom (unsigned number)

  tries to free Room with particular ID
- Hotel & getReport (DatePeriod &period)

  Creates report for the usage of this Hotel's rooms in the period from-to.
- void searchRoom (unsigned minBeds, const DatePeriod &period) const

  given minimum number of beds and a desired period to print most suitable rooms for accomodation
- bool serviceRoom (unsigned number, const DatePeriod &period, String note)

  plans maintenance for particular Room and period leaving note for the service
- Hotel & showToday ()

  print status of the building rooms
- Hotel & seeRoomForNights (unsigned number, unsigned nights)

  print soonest period of particular number of days when particular room is free
- bool workDay ()

  work with this Hotel for a whole day

Static Public Member Functions

- static Date today ()

  get today's Date according to all Hotels

5.3.1 Detailed Description

Class representing hotel with name, current Date and a building (list of rooms)

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Hotel() [1/3]

```
Hotel::Hotel ( )  [delete]
```

5.3.2.2 Hotel() [2/3]

```
Hotel::Hotel (
            String hotelDataFile )
```

Construct a new Hotel object.

Parameters

hotelDataFile	path to file where rooms are recorded

5.3.2.3 Hotel() [3/3]

```
Hotel::Hotel (
            const Hotel &  )  [delete]
```

5.3.2.4 ~Hotel()

```
Hotel::~Hotel ( )
```

Destroy the Hotel object.

5.3.3 Member Function Documentation

5.3.3.1 freeRoom()

```
bool Hotel::freeRoom (
            unsigned number )
```

tries to free Room with particular ID

Parameters

number	Room's ID

Returns

true room is now free

false room not found

5.3.3.2 getName()

```
String Hotel::getName ( ) const  [inline]
```

get the name of this Hotel

Returns

String

5.3.3.3 getReport()

```
Hotel & Hotel::getReport (
            DatePeriod & period )
```

Creates report for the usage of this Hotel's rooms in the period from-to.

Report written in file named "report-YYYY-MM-DD.txt" where YYYY-MM-DD is the beginning of the period

Parameters

period	desired period of time

Returns

Hotel& this Hotel

5.3.3.4 nextDay()

```
void Hotel::nextDay ( )
```

advance to the nextDay

5.3.3.5 operator=()

```
Hotel & Hotel::operator= (
            const Hotel &  )  [delete]
```

5.3.3.6 reserveRoom()

```
bool Hotel::reserveRoom (
            unsigned number,
            const DatePeriod & period,
            String name = "-",
            String note = "None\n" )
```

makes a new Reservation for particular Room and period with options for name of guest and notes to the Reservation

Parameters

number	of the desired Room
from	accomodation Date
to	leaving Date
name	guest's name
note	note to the reservation

Returns

true successfull reservation

false failed reservation (not made)

5.3.3.7 searchRoom()

```
void Hotel::searchRoom (
            unsigned minBeds,
            const DatePeriod & period ) const
```

given minimum number of beds and a desired period to print most suitable rooms for accomodation

Parameters

minBeds	minimum number of beds
period	desired time period

5.3.3.8 seeRoomForNights()

```
Hotel & Hotel::seeRoomForNights (
            unsigned number,
            unsigned nights )
```

print soonest period of particular number of days when particular room is free

Parameters

number	ID of a room
nights	number of nights to stay in the Hotel for

Returns

Hotel& this Hotel

5.3.3.9 serviceRoom()

```
bool Hotel::serviceRoom (
            unsigned number,
            const DatePeriod & period,
            String note )
```

plans maintenance for particular Room and period leaving note for the service

Parameters

number	Room's ID
period	desired period of time
note	any notes to the service

Returns

true service planned successfully

false service planning failed (room not found or is reserved for the period)

5.3.3.10 showAvailableRooms()

```
Hotel & Hotel::showAvailableRooms (
            std::ostream & os,
            Date d )
```

output to stream all available rooms for a particular Date

Returns

Hotel& this Hotel

5.3.3.11 showToday()

```
Hotel & Hotel::showToday ( )
```

print status of the building rooms

Returns

Hotel& this Hotel

5.3.3.12 today()

```
static Date Hotel::today ( )  [inline], [static]
```

get today's Date according to all Hotels

Returns

Date

5.3.3.13 workDay()

```
bool Hotel::workDay ( )
```

work with this Hotel for a whole day

Returns

true day ended with the Hotel still working

false the Hotel was closed and its data written to "<name>.dat"

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

- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Hotel.hpp
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Hotel.cpp

5.4 HotelBuilding Class Reference

Class representing list of rooms.

```
#include <HotelBuilding.hpp>
```

Public Member Functions

- HotelBuilding (std::ifstream &ifs)

  Construct a new HotelBuilding object from a text file containing rooms info Format of the file:
- HotelBuilding (const HotelBuilding &other)=delete
- HotelBuilding & operator= (HotelBuilding &other)=delete
- ~HotelBuilding ()

  Destroy the HotelBuilding object.
- size_t getRoomCount () const

  Get the room count.
- Room * operator[ ] (unsigned roomNumber) const

  seek for a room with particular number
- void newDate (Date d)

  update this HotelBuilding rooms data on a new Date
- void showAvailableRooms (std::ostream &os, Date d) const

  show available rooms on a particular Date
- void createReport (DatePeriod &period) const

  Create a report for the usage of rooms for a particular period of time (ending before the today Date) in folder reports Format of the report:
- void suggestRoom (unsigned beds, const DatePeriod &period)

  show top DISPLAY (or all rooms if less than DISPLAY + 1) sorted by suitabilty for the guest, given minimal number of beds and particular period of time for an eventual reservation
- void showRoomsStatesToday (Date today) const

  prints to stdout all rooms together with up to today Date info about their Availability (now, in future or in the past)
- void showRoomForNights (unsigned number, unsigned nights, Date today) const

  show soonest period of particular nights when particular room is free
- void writeToBinaryFile (std::ofstream &ofs)

  write the HotelBuilding data into binary file opened by ofstream
- void readDataFromBinary (std::ifstream &ifs)

  read the HotelBuilding data from binary file opened by ifstream

Friends

- class RoomAnalyzer

  using RoomAnalyzer to perform algorithms for the room list (database)

5.4.1 Detailed Description

Class representing list of rooms.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 HotelBuilding() [1/2]

```
HotelBuilding::HotelBuilding (
            std::ifstream & ifs )
```

Construct a new HotelBuilding object from a text file containing rooms info Format of the file:

1. <size> ... n(>1). <room #(n-2) number> <room #(n-2) count of beds>

Parameters

ifs	input file stream to text file, containing rooms data

5.4.2.2 HotelBuilding() [2/2]

```
HotelBuilding::HotelBuilding (
            const HotelBuilding & other )  [delete]
```

5.4.2.3 ~HotelBuilding()

```
HotelBuilding::~HotelBuilding ( )
```

Destroy the HotelBuilding object.

5.4.3 Member Function Documentation

5.4.3.1 createReport()

```
void HotelBuilding::createReport (
            DatePeriod & period ) const
```

Create a report for the usage of rooms for a particular period of time (ending before the today Date) in folder reports Format of the report:

Report for the usage of the rooms between <beginning of period> and <end of period>: ... Room #" <Room number> between <beginning of period> and <end of period>: <count of nights> nights.

Parameters

period	period of time

5.4.3.2 getRoomCount()

```
size_t HotelBuilding::getRoomCount ( ) const  [inline]
```

Get the room count.

Returns

size_t count of rooms

5.4.3.3 newDate()

```
void HotelBuilding::newDate (
            Date d )
```

update this HotelBuilding rooms data on a new Date

Parameters

d	new Date

5.4.3.4 operator=()

```
HotelBuilding & HotelBuilding::operator= (
            HotelBuilding & other )  [delete]
```

5.4.3.5 operator[]()

```
Room * HotelBuilding::operator[] (
            unsigned roomNumber ) const
```

seek for a room with particular number

Parameters

roomNumber	number if the sought room

Returns

Room∗ if found -> pointer to this Room else -> nullptr

5.4.3.6 readDataFromBinary()

```
void HotelBuilding::readDataFromBinary (
            std::ifstream & ifs )
```

read the HotelBuilding data from binary file opened by ifstream

Parameters

ofs	input stream connected to binary file

5.4.3.7 showAvailableRooms()

```
void HotelBuilding::showAvailableRooms (
            std::ostream & os,
            Date d ) const
```

show available rooms on a particular Date

Parameters

os	output stream where the available rooms will be shown in format: Available rooms for <today>: Number: <room number> Bed count: <count of beds> ...
d	Date

5.4.3.8 showRoomForNights()

```
void HotelBuilding::showRoomForNights (
            unsigned number,
            unsigned nights,
            Date today ) const
```

show soonest period of particular nights when particular room is free

Parameters

number	ID of the room
nights	length of the period
today	today's Date

5.4.3.9 showRoomsStatesToday()

```
void HotelBuilding::showRoomsStatesToday (
            Date today ) const
```

prints to stdout all rooms together with up to today Date info about their Availability (now, in future or in the past)

Parameters

today	Date

5.4.3.10 suggestRoom()

```
void HotelBuilding::suggestRoom (
            unsigned beds,
            const DatePeriod & period )
```

show top DISPLAY (or all rooms if less than DISPLAY + 1) sorted by suitabilty for the guest, given minimal number of beds and particular period of time for an eventual reservation

Parameters

beds	minimal number of beds insisted in the room
period	period of time

5.4.3.11 writeToBinaryFile()

```
void HotelBuilding::writeToBinaryFile (
            std::ofstream & ofs )
```

write the HotelBuilding data into binary file opened by ofstream

Parameters

ofs	output stream connected to binary file

5.4.4 Friends And Related Function Documentation

5.4.4.1 RoomAnalyzer

```
friend class RoomAnalyzer  [friend]
```

using RoomAnalyzer to perform algorithms for the room list (database)

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

- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelBuilding.hpp
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelBuilding.cpp

5.5 HotelInterface Class Reference

Utility Class taking care of the UI of the program.

`#include <HotelInterface.hpp>`

Static Public Member Functions

- static void createHeader (Hotel &H)
  print to stdout centered name of the Hotel
- static void beginDay ()
  print the today's date and available commands to use for a Hotel

5.5.1 Detailed Description

Utility Class taking care of the UI of the program.

5.5.2 Member Function Documentation

5.5.2.1 beginDay()

```
void HotelInterface::beginDay ( )  [static]
```

print the today's date and available commands to use for a Hotel

5.5.2.2 createHeader()

```
void HotelInterface::createHeader (
            Hotel & H )  [static]
```

print to stdout centered name of the Hotel

Parameters

H	Hotel whose name is to be printed

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

- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelInterface.hpp
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelInterface.cpp

5.6 Reservation Class Reference

Class representing information about a reservation.

`#include <Reservation.hpp>`

Public Member Functions

- Reservation (String name, const DatePeriod &p, String n="None.\n", bool s=false)
  Construct a new Reservation object.
- Reservation (const Reservation &)=delete
- Reservation & operator= (const Reservation &)=delete
- bool isActive () const
  see if this Reservation is active (today is part of the period)
- bool isPast () const
  see if this Reservation is past (today is after end of period)
- bool isServiced () const
  see if this Reservation is a maintenance
- Date getFrom () const
  get beginning Date of this Reservation
- Date getTo () const
  get end Date of this Reservation
- unsigned getNights () const
  get count of nights of this Reservation
- String getNote () const
  get the note to this Reservation
- void onDate (Date d)
  update the state of the reservation based on new today's Date (d)
- ReservationState stateOnDate (Date) const
  see what would the state of this Reservation be on particular Date
- bool LeavingInAdvance (Date)
  try to change end of period for earlier end of this Reservation
- void writeToBinaryFile (std::ofstream &ofs)
  write the Reservation data into binary file opened by ofstream
- void readDataFromBinary (std::ifstream &ifs)
  read the Reservation data from binary file opened by ifstream

5.6.1 Detailed Description

Class representing information about a reservation.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 Reservation() [1/2]

```
Reservation::Reservation (
            String name,
            const DatePeriod & p,
            String n = "None.\n",
            bool s = false )
```

Construct a new Reservation object.

Parameters

name	of the reserver
p	
n	note left for the reservation
s	whether it is reservation or maintenance

5.6.2.2 Reservation() [2/2]

```
Reservation::Reservation (
            const Reservation &  )  [delete]
```

5.6.3 Member Function Documentation

5.6.3.1 getFrom()

```
Date Reservation::getFrom ( ) const  [inline]
```

get beginning Date of this Reservation

Returns

Date beginning of the Reservation

5.6.3.2 getNights()

```
unsigned Reservation::getNights ( ) const  [inline]
```

get count of nights of this Reservation

Returns

unsigned count of nights of this Reservation

5.6.3.3 getNote()

```
String Reservation::getNote ( ) const  [inline]
```

get the note to this Reservation

Returns

String

5.6.3.4 getTo()

```
Date Reservation::getTo ( ) const  [inline]
```

get end Date of this Reservation

Returns

Date end of the Reservation

5.6.3.5 isActive()

```
bool Reservation::isActive ( ) const  [inline]
```

see if this Reservation is active (today is part of the period)

Returns

true Reservation is active

false Reservation is not active (past or future)

5.6.3.6 isPast()

```
bool Reservation::isPast ( ) const  [inline]
```

see if this Reservation is past (today is after end of period)

Returns

true Reservation is past

false Reservation is not past (active or future)

5.6.3.7 isServiced()

```
bool Reservation::isServiced ( ) const  [inline]
```

see if this Reservation is a maintenance

Returns

true this Reservation is a maintenace

false this Reservation is for a guest

5.6.3.8 LeavingInAdvance()

```
bool Reservation::LeavingInAdvance (
            Date newTo )
```

try to change end of period for earlier end of this Reservation

Returns

true end date modified for leaving in advance

false new leaving Date not appropriate for earlier leaving

5.6.3.9 onDate()

```
void Reservation::onDate (
            Date d )
```

update the state of the reservation based on new today's Date (d)

Parameters

d	new today's Date

5.6.3.10 operator=()

```
Reservation & Reservation::operator= (
            const Reservation &  )  [delete]
```

5.6.3.11 readDataFromBinary()

```
void Reservation::readDataFromBinary (
            std::ifstream & ifs )
```

read the Reservation data from binary file opened by ifstream

Parameters

ofs	input stream connected to binary file

5.6.3.12 stateOnDate()

```
ReservationState Reservation::stateOnDate (
            Date d ) const
```

see what would the state of this Reservation be on particular Date

Returns

ReservationState state on desired Date

5.6.3.13 writeToBinaryFile()

```
void Reservation::writeToBinaryFile (
            std::ofstream & ofs )
```

write the Reservation data into binary file opened by ofstream

Parameters

ofs	output stream connected to binary file

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

- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Reservation.hpp
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Reservation.cpp

5.7 Room Class Reference

Class representing a room in hotel.

`#include <Room.hpp>`

Public Member Functions

- Room (unsigned n, unsigned bC)

  Construct a new Room object.
- Room (const Room &)=delete

  forbidden copying of rooms
- Room & operator= (const Room &)=delete

  forbidden copying of rooms
- ~Room ()

  Destroy the Room object.
- unsigned getNumber () const
- unsigned getBedCount () const
- bool isFreeNow () const
- bool freeRoom (Reservation *&currentRes)

  try to free this room
- void newDate (Date)

  apply new Date to state of all reservations and respectively of the room availability
- bool isFreeOnDate (Date) const

  see if this room is free in certain date
- bool isFreeInPeriod (const DatePeriod &period) const

  see if this Room is free in particular period of time (it is free in all days of the period)
- bool showReservationsInPeriod (std::ostream &os, const DatePeriod &period) const

  print to output stream info about the number of nights (if positive) in a period this Room has been taken
- bool addReservation (String name, String note, const DatePeriod &period)

  try to add Reservation to this Room
- bool closeForService (String note, const DatePeriod &period)

  try to add Reservation (about a maintenance) to this Room
- void showActivity () const

  print to stdout information about this Room latest busyness
- void writeToBinaryFile (std::ofstream &ofs)

  write the Room data into binary file opened by ofstream
- void readDataFromBinary (std::ifstream &ifs)

  read the Room data from binary file opened by ifstream

5.7.1 Detailed Description

Class representing a room in hotel.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 Room() [1/2]

```
Room::Room (
            unsigned n,
            unsigned bC )
```

Construct a new Room object.

Parameters

n	number of constructed Room
bC	number of beds in constructed Room

5.7.2.2 Room() [2/2]

```
Room::Room (
            const Room &  )  [delete]
```

forbidden copying of rooms

5.7.2.3 ~Room()

```
Room::~Room ( )
```

Destroy the Room object.

5.7.3 Member Function Documentation

5.7.3.1 addReservation()

```
bool Room::addReservation (
            String name,
            String note,
            const DatePeriod & period )
```

try to add Reservation to this Room

Parameters

name	name of the guest
note	note to this Reservation
period	period of time

Returns

true successfully added Reservation

false adding a Reservation failed (the room is not free in this DatePeriod)

5.7.3.2 closeForService()

```
bool Room::closeForService (
            String note,
            const DatePeriod & period )
```

try to add Reservation (about a maintenance) to this Room

Parameters

note	note to this maintenance
period	period of time

Returns

true successfully added maintenance

false adding a maintenance failed (the room is not free in this DatePeriod)

5.7.3.3 freeRoom()

```
bool Room::freeRoom (
            Reservation *& currentRes )
```

try to free this room

Returns

true sucesfully freed room

false room is already free

5.7.3.4 getBedCount()

```
unsigned Room::getBedCount ( ) const  [inline]
```

5.7.3.5 getNumber()

```
unsigned Room::getNumber ( ) const  [inline]
```

5.7.3.6 isFreeInPeriod()

```
bool Room::isFreeInPeriod (
            const DatePeriod & period ) const
```

see if this Room is free in particular period of time (it is free in all days of the period)

Parameters

period	period of time

Returns

true the room is free (in all days of the period)

false the room is not free (there is a day in period when the room is taken)

5.7.3.7 isFreeNow()

```
bool Room::isFreeNow ( ) const
```

5.7.3.8 isFreeOnDate()

```
bool Room::isFreeOnDate (
            Date d ) const
```

see if this room is free in certain date

Returns

true the room is free

false the room is taken

5.7.3.9 newDate()

```
void Room::newDate (
            Date newD )
```

apply new Date to state of all reservations and respectively of the room availability

5.7.3.10 operator=()

```
Room & Room::operator= (
            const Room &  )  [delete]
```

forbidden copying of rooms

5.7.3.11 readDataFromBinary()

```
void Room::readDataFromBinary (
            std::ifstream & ifs )
```

read the Room data from binary file opened by ifstream

Parameters

ofs	input stream connected to binary file

5.7.3.12 showActivity()

```
void Room::showActivity ( ) const
```

print to stdout information about this Room latest busyness

5.7.3.13 showReservationsInPeriod()

```
bool Room::showReservationsInPeriod (
            std::ostream & os,
            const DatePeriod & period ) const
```

print to output stream info about the number of nights (if positive) in a period this Room has been taken

Parameters

os	output stream
period	period of time

Returns

true there has been taken for at least one night and info has been printed

false the room has been free during this period and no info has been printed

5.7.3.14 writeToBinaryFile()

```
void Room::writeToBinaryFile (
            std::ofstream & ofs )
```

write the Room data into binary file opened by ofstream

Parameters

ofs	output stream connected to binary file

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

- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Room.hpp
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Room.cpp

5.8 RoomAnalyzer Class Reference

Utility class to perform algorythms on the rooms in a building.

`#include <RoomAnalyzer.hpp>`

Static Public Member Functions

- static void suggest (HotelBuilding &hB, unsigned beds, DatePeriod period)

  print top DISPLAY rooms info based on suitability of a Room (desired number of beds and period of time)
- static void soonestFreePeriod (const HotelBuilding &hB, unsigned number, unsigned nights, Date today)

  print soonest period when a particular room is free for particular number of nights

5.8.1 Detailed Description

Utility class to perform algorythms on the rooms in a building.

5.8.2 Member Function Documentation

5.8.2.1 soonestFreePeriod()

```
void RoomAnalyzer::soonestFreePeriod (
            const HotelBuilding & hB,
            unsigned number,
            unsigned nights,
            Date today )  [static]
```

print soonest period when a particular room is free for particular number of nights

Parameters

hB	HotelBuilding
number	ID of the Room
nights	length of a period
today	today's Date

5.8.2.2 suggest()

```
void RoomAnalyzer::suggest (
            HotelBuilding & hB,
            unsigned beds,
            DatePeriod period )  [static]
```

print top DISPLAY rooms info based on suitability of a Room (desired number of beds and period of time)

Parameters

hB	HotelBuilding
beds	desired number of beds
period	period of time

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

- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/RoomAnalyzer.hpp
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/RoomAnalyzer.cpp

5.9 String Class Reference

Class representing dynamic string (array of characters)

`#include <String.hpp>`

Public Member Functions

- String (const char *_str="")
  Construct a new String object by array of chars.
- String (const String &other)
  copy constructor (makes deep copy of other)
- String & operator= (const String &other)
  makes deep copy of other into this
- ~String ()
  Destroy the String object.
- size_t size () const
  get the size of the array (number of symbols before the '\0')
- size_t capacity () const
  get the maximum characters the array can store
- char * c_str () const
  get pointer to the beginning of the array
- String operator+ (const String &other) const
  concatenation of two strings
- String & operator+= (const String &other)
  concatenate another String to this String
- operator const char * () const
  *cast this String to const char * by returning pointer to the beginning*
- String & shrink_to_fit ()
  make the capacity equal to the size

5.9.1 Detailed Description

Class representing dynamic string (array of characters)

5.9.2 Constructor & Destructor Documentation

5.9.2.1 String() [1/2]

```
String::String (
            const char * _str = "" )
```

Construct a new String object by array of chars.

Parameters

_str	

5.9.2.2 String() [2/2]

```
String::String (
            const String & other )
```

copy constructor (makes deep copy of other)

Parameters

other	String

5.9.2.3 ~String()

```
String::~String ( )
```

Destroy the String object.

5.9.3 Member Function Documentation

5.9.3.1 c_str()

```
char * String::c_str ( ) const  [inline]
```

get pointer to the beginning of the array

5.9.3.2 capacity()

```
size_t String::capacity ( ) const  [inline]
```

get the maximum characters the array can store

5.9.3.3 operator const char *()

```
String::operator const char * ( ) const  [inline]
```

cast this String to const char * by returning pointer to the beginning

5.9.3.4 operator+()

```
String String::operator+ (
            const String & other ) const
```

concatenation of two strings

5.9.3.5 operator+=()

```
String & String::operator+= (
            const String & other )
```

concatenate another String to this String

5.9.3.6 operator=()

```
String & String::operator= (
            const String & other )
```

makes deep copy of other into this

Parameters

other	String

Returns

String& this String

5.9.3.7 shrink_to_fit()

```
String & String::shrink_to_fit ( )
```

make the capacity equal to the size

5.9.3.8 size()

```
size_t String::size ( ) const  [inline]
```

get the size of the array (number of symbols before the '\0')

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

- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/String.hpp
- C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/String.cpp

Chapter 6

File Documentation

6.1 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/3.23.0-rc2/CompilerIdC/CMakeCCompilerId.c File Reference

Macros

- #define __has_include(x) 0
- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X) #X
- #define STRINGIFY(X) STRINGIFY_HELPER(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define C_VERSION

Functions

- int main (int argc, char *argv[ ])

Variables

- char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char * info_language_standard_default
- const char * info_language_extensions_default

6.1.1 Macro Definition Documentation

6.1.1.1 __has_include

```
#define __has_include(
              x ) 0
```

6.1.1.2 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

6.1.1.3 C_VERSION

```
#define C_VERSION
```

6.1.1.4 COMPILER_ID

```
#define COMPILER_ID ""
```

6.1.1.5 DEC

```
#define DEC(
              n )
```

Value:

```
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))
```

6.1.1.6 HEX

```
#define HEX(
              n )
```

Value:

```
  ('0' + ((n)»28 & 0xF)), \
  ('0' + ((n)»24 & 0xF)), \
  ('0' + ((n)»20 & 0xF)), \
  ('0' + ((n)»16 & 0xF)), \
  ('0' + ((n)»12 & 0xF)), \
  ('0' + ((n)»8  & 0xF)), \
  ('0' + ((n)»4  & 0xF)), \
  ('0' + ((n)     & 0xF))
```

6.1.1.7 PLATFORM_ID

```
#define PLATFORM_ID
```

6.1.1.8 STRINGIFY

```
#define STRINGIFY(
              X ) STRINGIFY_HELPER(X)
```

6.1.1.9 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(
              X ) #X
```

6.1.2 Function Documentation

6.1.2.1 main()

```
int main (
             int argc,
             char * argv[] )
```

6.1.3 Variable Documentation

6.1.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

6.1.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

6.1.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

Initial value:

```
= "INFO" ":" "extensions_default["
  "OFF"
"]"
```

6.1.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

Initial value:

```
=
  "INFO" ":" "standard_default[" C_VERSION "]"
```

6.1.3.5 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

6.2 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/3.23.0-rc2/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

Macros

- #define __has_include(x) 0
- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X) #X
- #define STRINGIFY(X) STRINGIFY_HELPER(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define CXX_STD __cplusplus

Functions

- int main (int argc, char *argv[])

Variables

- char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char * info_language_standard_default
- const char * info_language_extensions_default

6.2.1 Macro Definition Documentation

6.2.1.1 __has_include

```
#define __has_include(
              x ) 0
```

6.2.1.2 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

6.2.1.3 COMPILER_ID

```
#define COMPILER_ID ""
```

6.2.1.4 CXX_STD

```
#define CXX_STD __cplusplus
```

6.2.1.5 DEC

```
#define DEC(
              n )
```

Value:

```
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))
```

6.2.1.6 HEX

```
#define HEX(
              n )
```

Value:

```
  ('0' + ((n)»28 & 0xF)), \
  ('0' + ((n)»24 & 0xF)), \
  ('0' + ((n)»20 & 0xF)), \
  ('0' + ((n)»16 & 0xF)), \
  ('0' + ((n)»12 & 0xF)), \
  ('0' + ((n)»8  & 0xF)), \
  ('0' + ((n)»4  & 0xF)), \
  ('0' + ((n)     & 0xF))
```

6.2.1.7 PLATFORM_ID

```
#define PLATFORM_ID
```

6.2.1.8 STRINGIFY

```
#define STRINGIFY(
              X ) STRINGIFY_HELPER(X)
```

6.2.1.9 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(
              X ) #X
```

6.2.2 Function Documentation

6.2.2.1 main()

```
int main (
             int argc,
             char * argv[] )
```

6.2.3 Variable Documentation

6.2.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

6.2.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

6.2.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

Initial value:

```
= "INFO" ":" "extensions_default["
  "OFF"
"]"
```

6.2.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

Initial value:

```
= "INFO" ":" "standard_default["
  "98"
"]"
```

6.2.3.5 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

6.3 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/hotel.dir/Date.cpp.obj.d File Reference

6.4 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/hotel.dir/Hotel.cpp.obj.d File Reference

6.5 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/hotel.dir/HotelBuilding.cpp.obj.d File Reference

6.6 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/hotel.dir/HotelInterface.cpp.obj.d File Reference

6.7 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/hotel.dir/main.cpp.obj.d File Reference

6.8 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/hotel.dir/Reservation.cpp.obj.d File Reference

6.9 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/hotel.dir/Room.cpp.obj.d File Reference

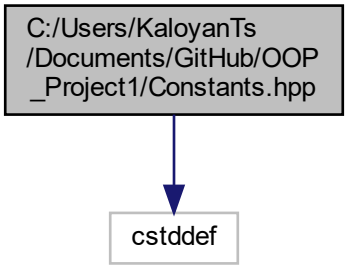
6.10 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/hotel.dir/RoomAnalyzer.cpp.obj.d File Reference

6.11 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/build/CMakeFiles/hotel.dir/String.cpp.obj.d File Reference

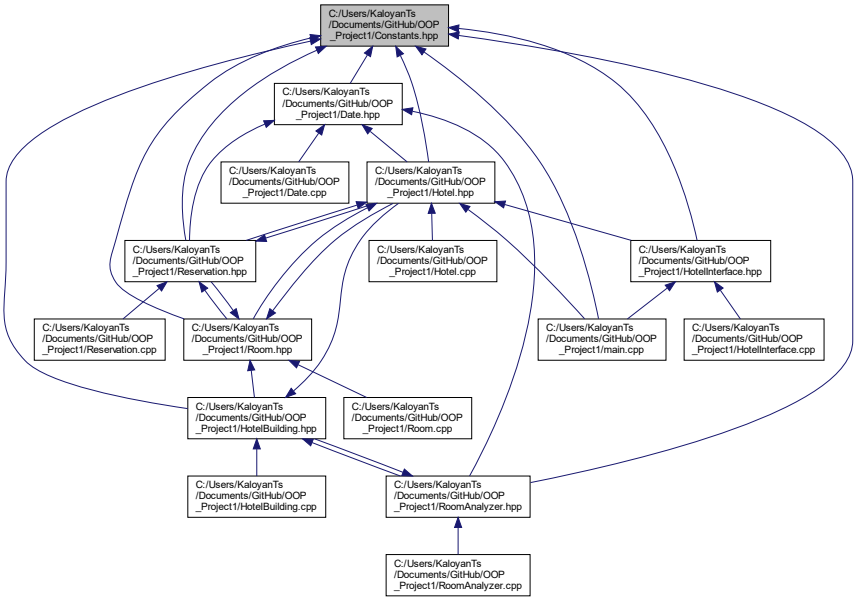
6.12 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Constants.hpp File Reference

```
#include <cstddef>
```

Include dependency graph for Constants.hpp:



This graph shows which files directly or indirectly include this file:



Variables

- const unsigned daysFromBeginning [ ] = {0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334}

  array keeping days past from January 1st
- const size_t DISPLAY_WIDTH = 130

  width of the console used for centering the text
- const size_t COMMANDS = 8

  number of different commands to use for a Hotel
- const size_t STRING_MAX_LENGTH = 128

  max length of strings (char arrays)
- const size_t INIT_CAPACITY = 2

const variable keeping minimal size of the reservations list

- const size_t DISPLAY = 5

  max number of rooms to be printed

- const char cmdArr [COMMANDS][2][STRING_MAX_LENGTH]

  array of all available commands and instructions how to use them

6.12.1 Variable Documentation

6.12.1.1 cmdArr

```
const char cmdArr[COMMANDS][2][STRING_MAX_LENGTH]
```

Initial value:

```
= {
    {{"To make a reservation, enter "},
     {"<reserve: [Room number] [Accomodation date] [Departure date] {Guest name[;]} {Note}>"}},
    {{"To see list of free rooms for a particular date, enter "},
     {"<available: [date]>"}},
    {{"To free a room now, enter "},
     {"<free: [Room number]>"}},
    {{"To get report about the reservations of a room over a period of time, enter "},
     {"<report: [From date] [To date]>"}},
    {{"To request a room for guests, enter"},
     {"<request: [minimal number of beds] [Accomodation date] [Departure date]>"}},
    {{"To close a room for maintenance, enter"},
     {"<maintenance: [room number] [From date] [To date] [Note]>"}},
    {{"To see activity of all rooms, enter "}, {"<rooms:>"}},
    {{"To see soonest date a room is free for some nights, enter "},
     {"<plan: [Room number] [Number of nights]>"}}}
```

array of all available commands and instructions how to use them

6.12.1.2 COMMANDS

```
const size_t COMMANDS = 8
```

number of different commands to use for a Hotel

6.12.1.3 daysFromBeginning

```
const unsigned daysFromBeginning[] = {0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334}
```

array keeping days past from January 1st

6.12.1.4 DISPLAY

```
const size_t DISPLAY = 5
```

max number of rooms to be printed

6.12.1.5 DISPLAY_WIDTH

```
const size_t DISPLAY_WIDTH = 130
```

width of the console used for centering the text

6.12.1.6 INIT_CAPACITY

```
const size_t INIT_CAPACITY = 2
```

const variable keeping minimal size of the reservations list

6.12.1.7 STRING_MAX_LENGTH

```
const size_t STRING_MAX_LENGTH = 128
```

max length of strings (char arrays)

6.13 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/↩ Constants.hpp

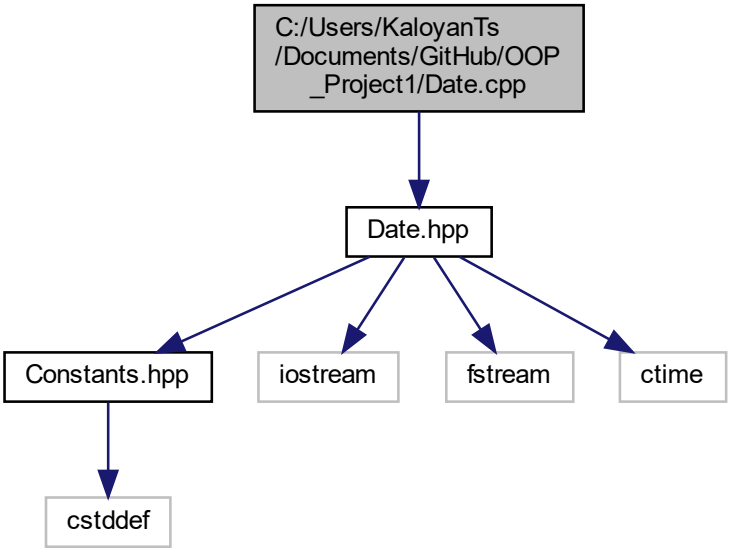
Go to the documentation of this file.

```
1 #ifndef __CONSTANTS_HPP
2 #define __CONSTANTS_HPP
3 #include <cstddef>
4
9 const unsigned daysFromBeginning[] = {0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334};
10
15 const size_t DISPLAY_WIDTH = 130;
16
21 const size_t COMMANDS = 8;
22
27 const size_t STRING_MAX_LENGTH = 128;
28
33 const size_t INIT_CAPACITY = 2;
34
39 const size_t DISPLAY = 5;
40
45 const char cmdArr[COMMANDS][2][STRING_MAX_LENGTH] = {
46     {{"To make a reservation, enter "},
47      {"<reserve: [Room number] [Accomodation date] [Departure date] {Guest name[;]} {Note}>"}},
48     {{"To see list of free rooms for a particular date, enter "},
49      {"<available: [date]>"}},
50     {{"To free a room now, enter "},
51      {"<free: [Room number]>"}},
52     {{"To get report about the reservations of a room over a period of time, enter "},
53      {"<report: [From date] [To date]>"}},
54     {{"To request a room for guests, enter"},
55      {"<request: [minimal number of beds] [Accomodation date] [Departure date]>"}},
56     {{"To close a room for maintenance, enter"},
57      {"<maintenance: [room number] [From date] [To date] [Note]>"}},
58     {{"To see activity of all rooms, enter "}, {"<rooms:>"}},
59     {{"To see soonest date a room is free for some nights, enter "},
60      {"<plan: [Room number] [Number of nights]>"}}};
61
62 #endif
```

6.14 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Date.cpp File Reference

```
#include "Date.hpp"
```

Include dependency graph for Date.cpp:



Functions

- std::istream & operator>> (std::istream &is, Date &d)
- std::ostream & operator<< (std::ostream &os, const Date &d)
- std::istream & operator>> (std::istream &is, DatePeriod &dP)

  overloaded operator>> for DatePeriod input

6.14.1 Function Documentation

6.14.1.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & os,
            const Date & d )
```

Parameters

os	output stream
d	Date to be output

Returns

std::ostream& reference to the output stream

6.14.1.2 operator>>() [1/2]

```
std::istream & operator>> (
            std::istream & is,
            Date & d )
```

Parameters

is	input stream
d	Date to be input

Returns

std::istream& reference to the input stream

6.14.1.3 operator>>() [2/2]

```
std::istream & operator>> (
            std::istream & is,
            DatePeriod & dP )
```

overloaded operator>> for DatePeriod input

Parameters

is	input stream
dP	DatePeriod to be input

Returns

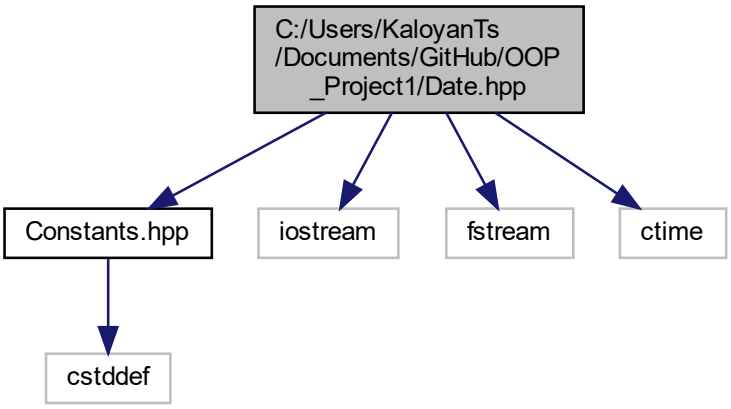
std::istream& this input stream

6.15 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Date.hpp File Reference

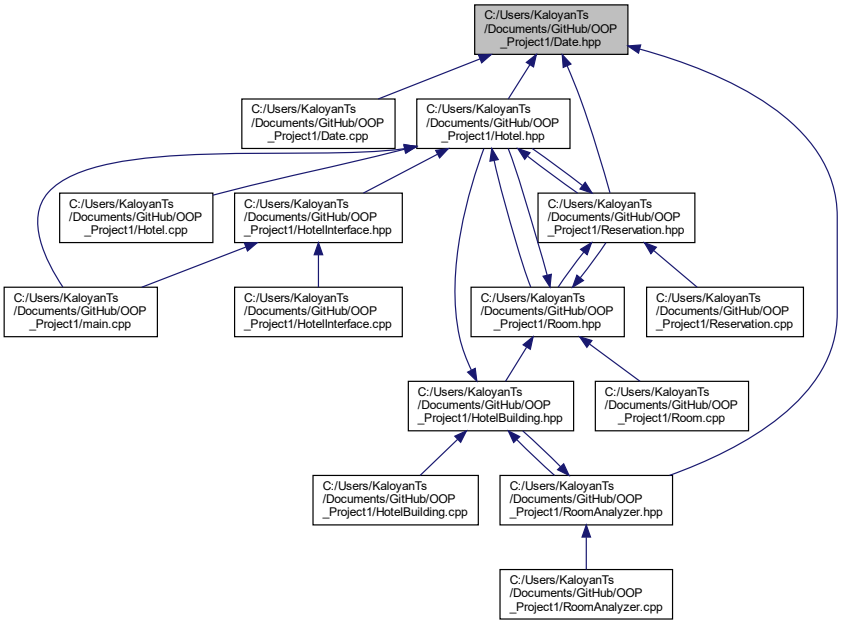
```
#include "Constants.hpp"
#include <iostream>
#include <fstream>
```

```
#include <ctime>
```

Include dependency graph for Date.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class Date

  Class representing date with day, month and year.

- struct DatePeriod

  Class containing two dates forming a period of time from Date to Date.

Functions

- std::istream & operator>> (std::istream &is, DatePeriod &dP)

  overloaded operator>> for DatePeriod input

6.15.1 Function Documentation

6.15.1.1 operator>>()

```
std::istream & operator>> (
            std::istream & is,
            DatePeriod & dP )
```

overloaded operator>> for DatePeriod input

Parameters

is	input stream
dP	DatePeriod to be input

Returns

std::istream& this input stream

6.16 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Date.hpp

Go to the documentation of this file.

```
1 #ifndef __DATE_HPP
2 #define __DATE_HPP
3 #include "Constants.hpp"
4 #include <iostream>
5 #include <fstream>
6 #include <ctime>
7
12 class Date
13 {
18     unsigned short day, month, year;
25     bool isVaid() const;
33     bool isLeap(unsigned y) const;
34
35 public:
43     Date(unsigned short d = 1, unsigned short m = 1, unsigned short y = 1900) : day(d), month(m), year(y)
       {}
51     bool operator<(const Date other) const;
55     bool operator<=(const Date other) const;
63     bool operator>(const Date other) const;
67     bool operator>=(const Date other) const;
75     bool operator==(const Date other) const;
82     const char *operator()(char *buf) const;
89     int operator-(Date other) const;
95     Date &operator++();
96
102     static Date getToday();
103
111     friend std::istream &operator»(std::istream &is, Date &d);
119     friend std::ostream &operator«(std::ostream &os, const Date &d);
120
126     void writeToBinaryFile(std::ofstream &ofs);
```

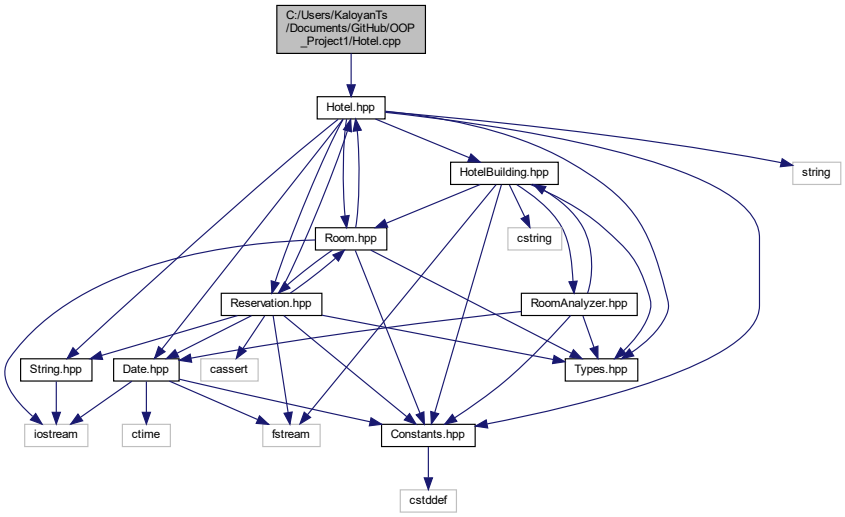
```
127
133     void readDataFromBinary(std::ifstream &ifs);
134 };
135
140 struct DatePeriod
141 {
146     Date from;
147
152     Date to;
153
159     unsigned length() const { return to - from; }
160
166     DatePeriod &operator++();
167
172     void readProper();
173 };
174
182 std::istream &operator»(std::istream &is, DatePeriod &dP);
183
184 #endif
```

6.17 markdowns/commands.md File Reference

6.18 markdowns/mainpage.md File Reference

6.19 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Hotel.cpp File Reference

`#include "Hotel.hpp"`
Include dependency graph for Hotel.cpp:



Functions

- String readFromIfstream (std::ifstream &ifs, size_t len)

  read string from input file stream with certain length

6.19.1 Function Documentation

6.19.1.1 readFromIfstream()

```
String readFromIfstream (
            std::ifstream & ifs,
            size_t len )
```

read string from input file stream with certain length

Parameters

ifs	input stream
len	maximum length of the read string

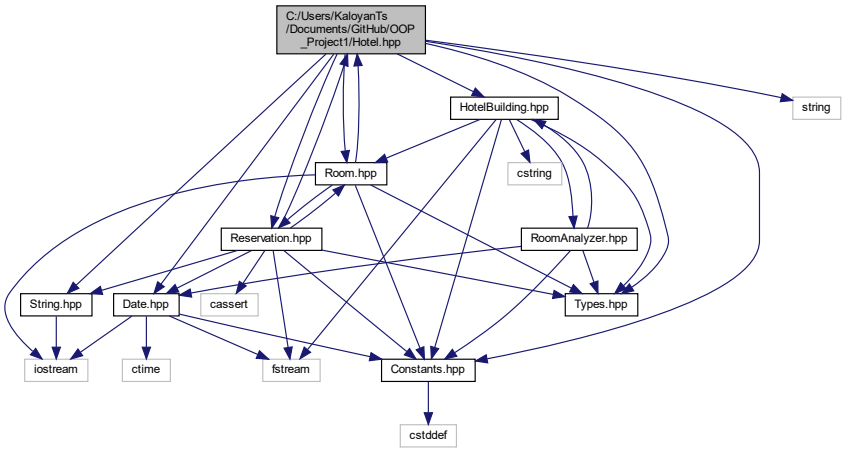
Returns

String

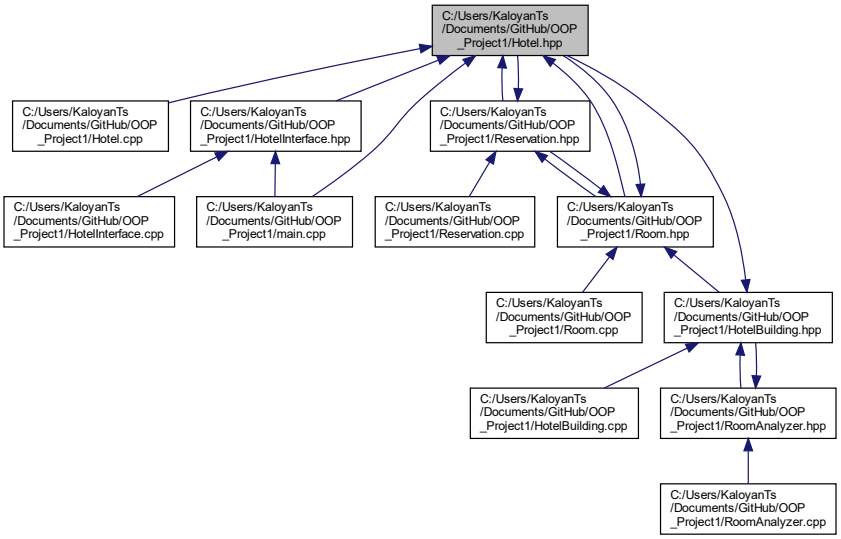
6.20 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Hotel.hpp File Reference

```
#include "Types.hpp"
#include "Constants.hpp"
#include "String.hpp"
#include "Date.hpp"
#include "Room.hpp"
#include "Reservation.hpp"
#include "HotelBuilding.hpp"
#include <string>
```

Include dependency graph for Hotel.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class Hotel

  Class representing hotel with name, current Date and a building (list of rooms)

Functions

- String readFromIfstream (std::ifstream &ifs, size_t len)

  read string from input file stream with certain length

6.20.1 Function Documentation

6.20.1.1 readFromIfstream()

```
String readFromIfstream (
            std::ifstream & ifs,
            size_t len )
```

read string from input file stream with certain length

Parameters

ifs	input stream
len	maximum length of the read string

Returns

String

6.21 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Hotel.hpp

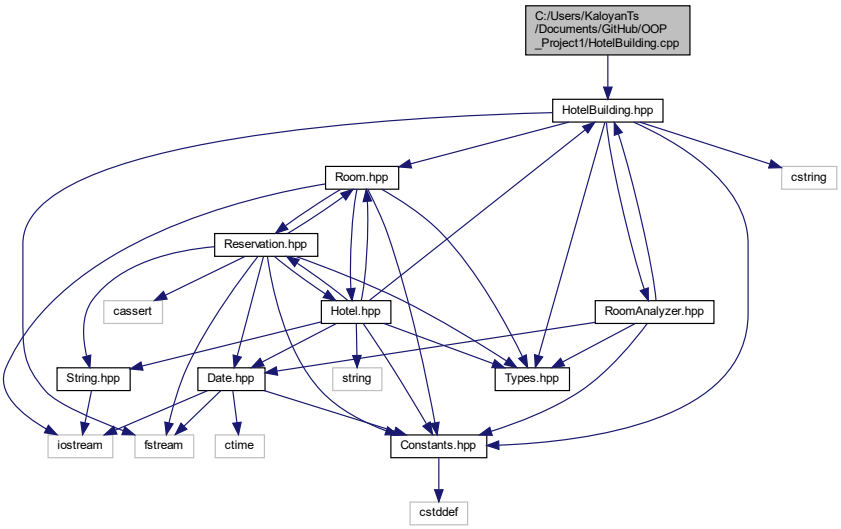
Go to the documentation of this file.

```
1 #ifndef __HOTEL_HPP
2 #define __HOTEL_HPP
3 #include "Types.hpp"
4 #include "Constants.hpp"
5 #include "String.hpp"
6 #include "Date.hpp"
7 #include "Room.hpp"
8 #include "Reservation.hpp"
9 #include "HotelBuilding.hpp"
10 #include <string>
11
19 String readFromIfstream(std::ifstream &ifs, size_t len);
20
25 class Hotel
26 {
31     String name;
32
37     static Date now;
38
43     HotelBuilding *building;
44
49     void writeToBinaryFile();
50
51 public:
52     Hotel() = delete;
53
59     Hotel(String hotelDataFile);
60     Hotel(const Hotel &) = delete;
61     Hotel &operator=(const Hotel &) = delete;
62
67     ~Hotel();
68
74     static Date today() { return now; }
75
81     String getName() const { return name; }
82
87     void nextDay();
88
100     bool reserveRoom(unsigned number, const DatePeriod &period, String name = "-", String note =
        "None\n");
101
107     Hotel &showAvailableRooms(std::ostream &, Date);
108
116     bool freeRoom(unsigned number);
117
124     Hotel &getReport(DatePeriod &period);
125
132     void searchRoom(unsigned minBeds, const DatePeriod &period) const;
133
143     bool serviceRoom(unsigned number, const DatePeriod &period, String note);
144
150     Hotel &showToday();
151
159     Hotel &seeRoomForNights(unsigned number, unsigned nights);
160
167     bool workDay();
168 };
169
170 #endif
```

6.22 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelBuilding.cpp File Reference

```
#include "HotelBuilding.hpp"
```

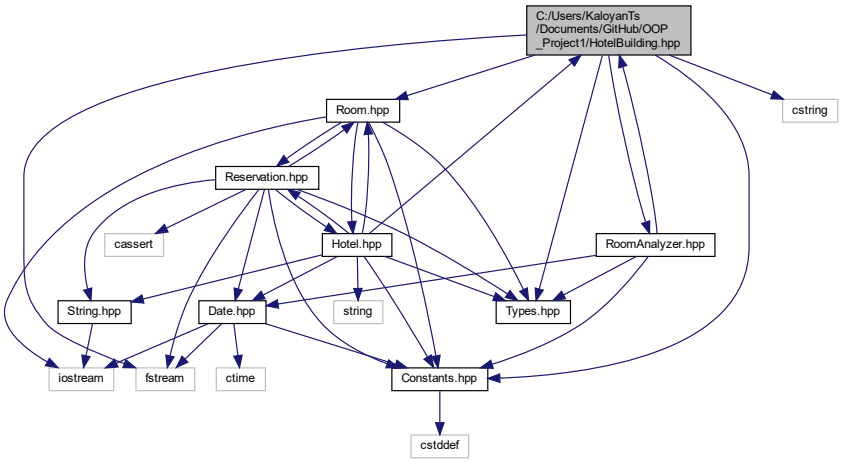
Include dependency graph for HotelBuilding.cpp:



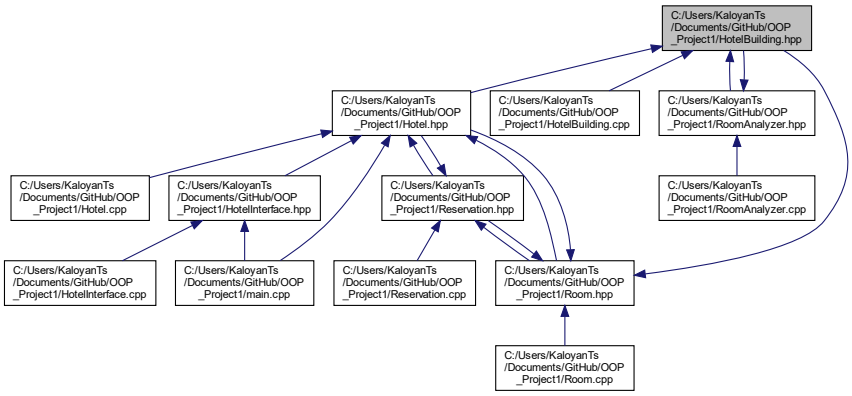
6.23 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelBuilding.hpp File Reference

```
#include "Types.hpp"
#include "Constants.hpp"
#include "Room.hpp"
#include "RoomAnalyzer.hpp"
#include <fstream>
#include <cstring>
```

Include dependency graph for HotelBuilding.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class HotelBuilding

  Class representing list of rooms.

6.24 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelBuilding.hpp

Go to the documentation of this file.

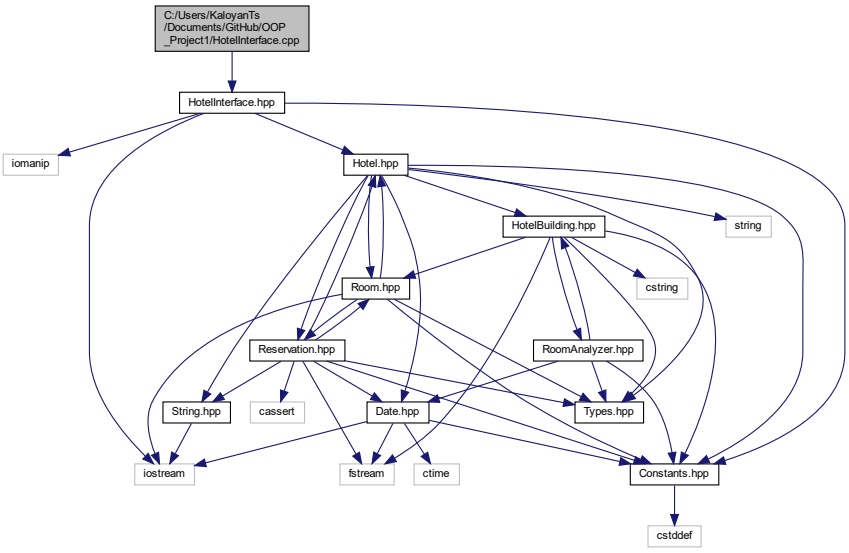
```
1 #ifndef __HOTELBUILDING_HPP
2 #define __HOTELBUILDING_HPP
3 #include "Types.hpp"
4 #include "Constants.hpp"
5 #include "Room.hpp"
```

```
6 #include "RoomAnalyzer.hpp"
7 #include <fstream>
8 #include <cstring>
9
14 class HotelBuilding
15 {
20     Room **rooms;
25     size_t size;
26
27 public:
37     HotelBuilding(std::ifstream &ifs);
38     HotelBuilding(const HotelBuilding &other) = delete;
39     HotelBuilding &operator=(HotelBuilding &other) = delete;
44     ~HotelBuilding();
45
51     size_t getRoomCount() const { return size; }
52
60     Room *operator[](unsigned roomNumber) const;
61
67     void newDate(Date d);
68
79     void showAvailableRooms(std::ostream &os, Date d) const;
80
91     void createReport(DatePeriod &period) const;
92
99     void suggestRoom(unsigned beds, const DatePeriod &period);
100
106     void showRoomsStatesToday(Date today) const;
107
115     void showRoomForNights(unsigned number, unsigned nights, Date today) const;
116
122     void writeToBinaryFile(std::ofstream &ofs);
123
129     void readDataFromBinary(std::ifstream &ifs);
130
135     friend class RoomAnalyzer;
136 };
137
138 #endif
```

6.25 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelInterface.cpp File Reference

`#include "HotelInterface.hpp"`

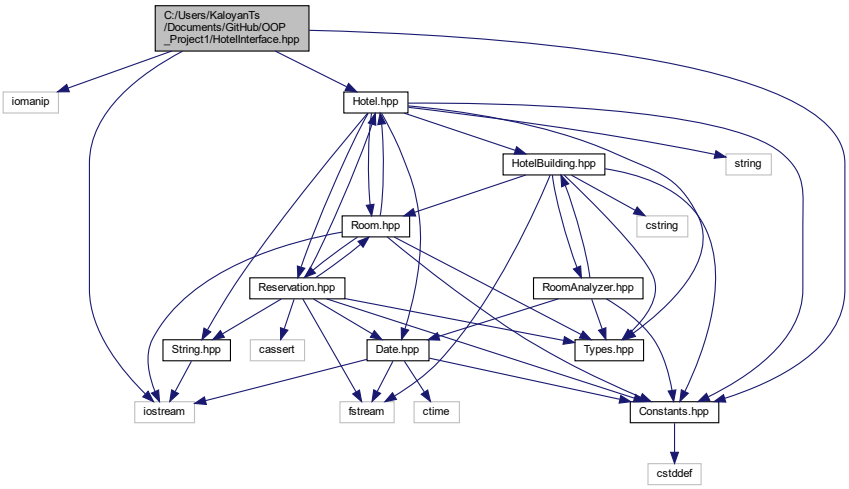
Include dependency graph for HotelInterface.cpp:



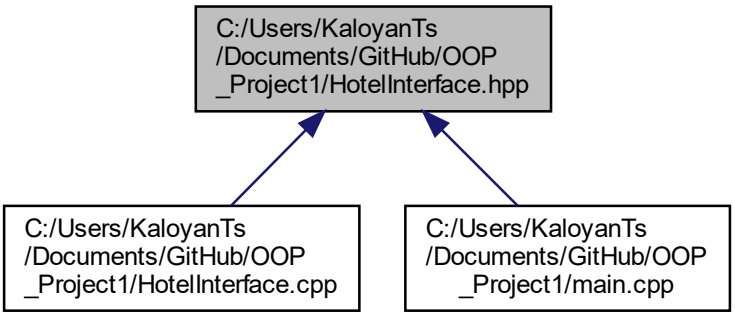
6.26 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelInterface.hpp File Reference

```
#include <iomanip>
#include <iostream>
#include "Constants.hpp"
#include "Hotel.hpp"
```

Include dependency graph for HotelInterface.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class HotelInterface

  Utility Class taking care of the UI of the program.

6.27 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/HotelInterface.hpp

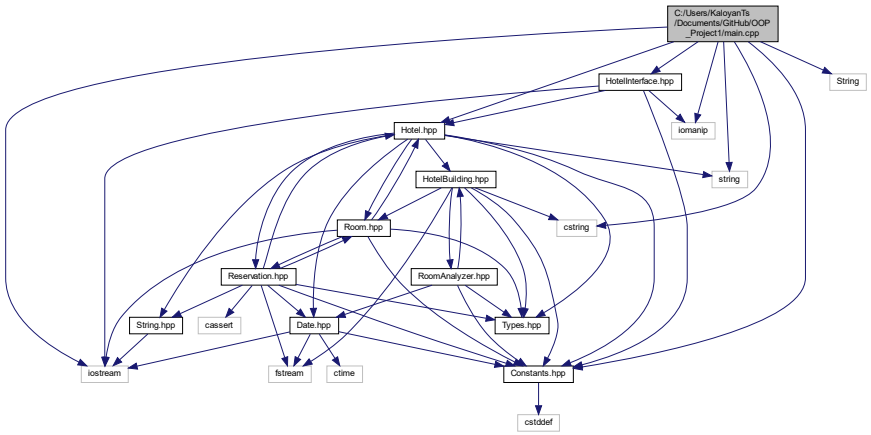
Go to the documentation of this file.

```
1 #ifndef __HOTELINTERFACE_HPP
2 #define __HOTELINTERFACE_HPP
3 #include <iomanip>
4 #include <iostream>
5 #include "Constants.hpp"
6 #include "Hotel.hpp"
7
9 class HotelInterface
10 {
11 public:
17     static void createHeader(Hotel &H);
18
23     static void beginDay();
24 };
25
26 #endif
```

6.28 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/main.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <cstring>
#include <string>
#include "Constants.hpp"
#include "String"
#include "HotelInterface.hpp"
#include "Hotel.hpp"
```

Include dependency graph for main.cpp:



Functions

- int main ()

6.28.1 Function Documentation

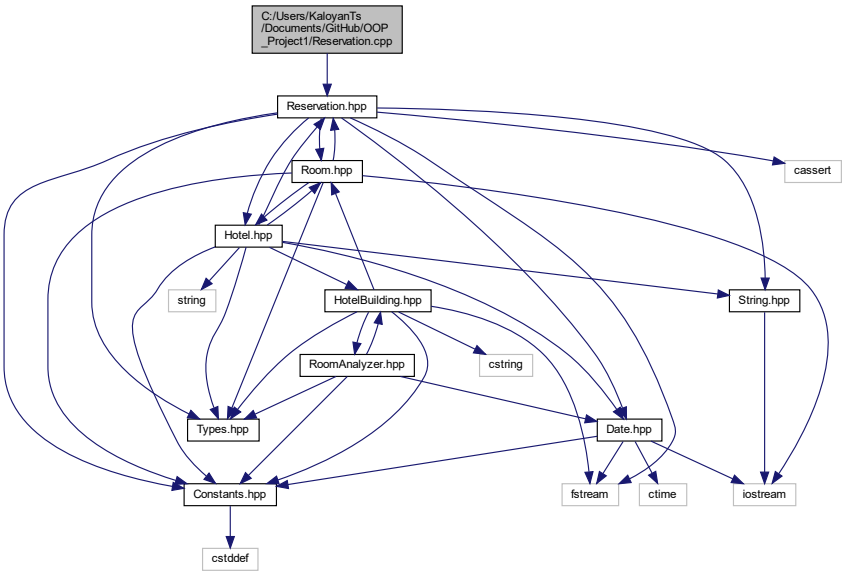
6.28.1.1 main()

```
int main ( )
```

6.29 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/↩ Reservation.cpp File Reference

```
#include "Reservation.hpp"
```

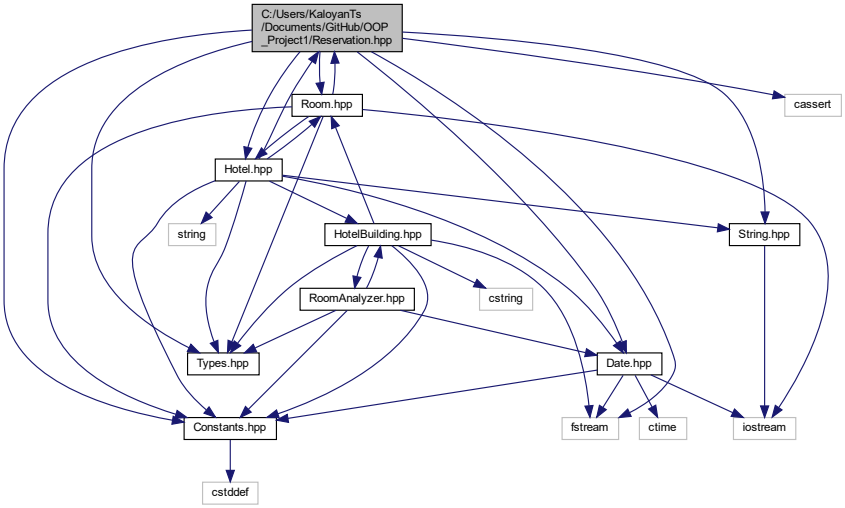
Include dependency graph for Reservation.cpp:



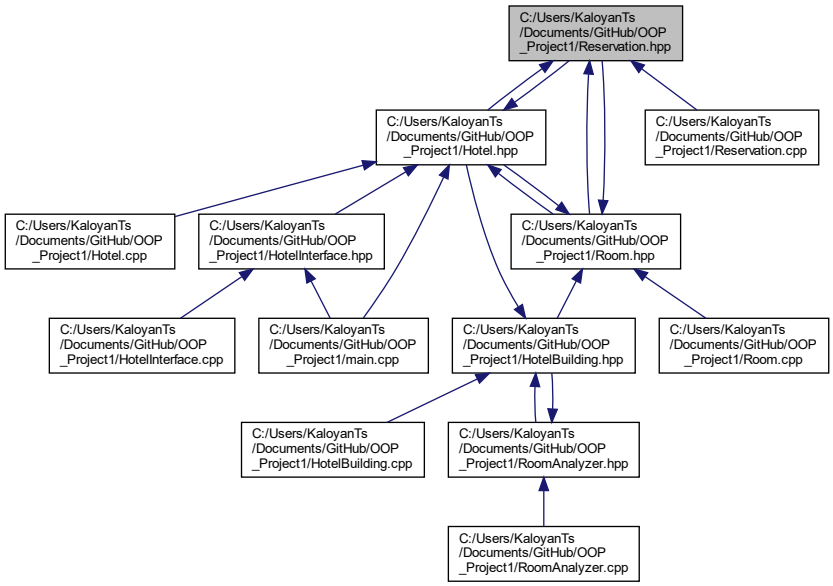
6.30 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/↩ Reservation.hpp File Reference

```
#include "Types.hpp"
#include "Constants.hpp"
#include "String.hpp"
#include "Room.hpp"
#include "Date.hpp"
#include "Hotel.hpp"
#include <cassert>
#include <fstream>
```

Include dependency graph for Reservation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class Reservation

  Class representing information about a reservation.

Enumerations

- enum ReservationState { UNKNOWN = 0 , PAST , ACTIVE , FUTURE }

  different states in time for a reservation

Functions

- std::ostream & operator<< (std::ostream &, const Reservation &)

  overloaded operator for ouputting a reservation

6.30.1 Enumeration Type Documentation

6.30.1.1 ReservationState

```
enum ReservationState
```

different states in time for a reservation

Enumerator

UNKNOWN	
PAST	
ACTIVE	
FUTURE	

6.30.2 Function Documentation

6.30.2.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & ,
            const Reservation &  )
```

overloaded operator for ouputting a reservation

Returns

std::ostream& ouput stream

6.31 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/↩ Reservation.hpp

Go to the documentation of this file.

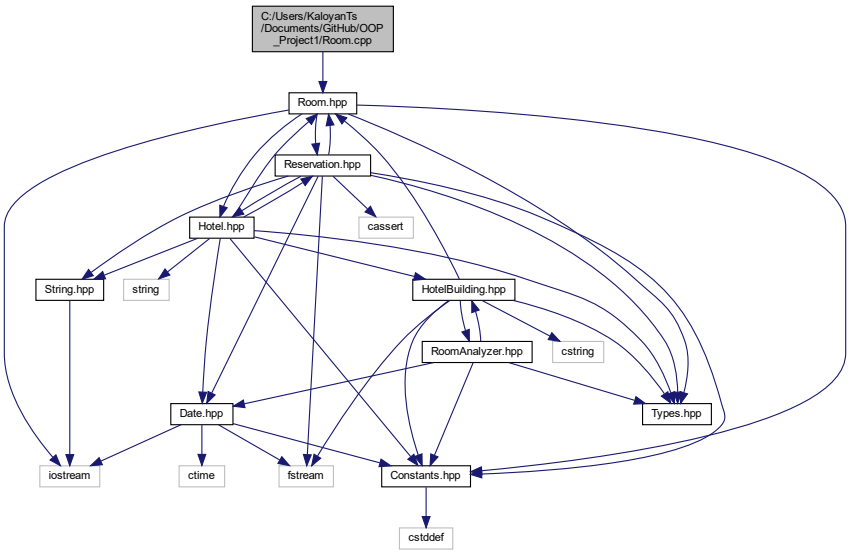
```
1 #ifndef __RESERVATION_HPP
2 #define __RESERVATION_HPP
3 #include "Types.hpp"
4 #include "Constants.hpp"
5 #include "String.hpp"
```

```
6 #include "Room.hpp"
7 #include "Date.hpp"
8 #include "Hotel.hpp"
9 #include <cassert>
10 #include <fstream>
11
16 enum ReservationState
17 {
18     UNKNOWN = 0,
19     PAST,
20     ACTIVE,
21     FUTURE
22 };
23
28 class Reservation
29 {
34     String guestName;
39     String note;
44     DatePeriod period;
49     ReservationState state;
54     bool service;
55
56 public:
65     Reservation(String name, const DatePeriod &p, String n = "None.\n", bool s = false);
66     Reservation(const Reservation &) = delete;
67     Reservation &operator=(const Reservation &) = delete;
68
75     bool isActive() const { return state == ACTIVE; }
76
83     bool isPast() const { return state == PAST; }
84
91     bool isServiced() const { return service; }
92
98     Date getFrom() const { return period.from; }
99
105     Date getTo() const { return period.to; }
106
112     unsigned getNights() const { return period.length(); }
113
119     String getNote() const { return note; }
120
126     void onDate(Date d);
127
133     ReservationState stateOnDate(Date) const;
134
141     bool LeavingInAdvance(Date);
142
148     void writeToBinaryFile(std::ofstream &ofs);
149
155     void readDataFromBinary(std::ifstream &ifs);
156 };
157
163 std::ostream &operator«(std::ostream &, const Reservation &);
164
165 #endif
```

6.32 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Room.cpp File Reference

```
#include "Room.hpp"
```

Include dependency graph for Room.cpp:



Functions

- std::ostream & operator<< (std::ostream &os, const Room &R)

  prints Room's base info (number and bedCount)

6.32.1 Function Documentation

6.32.1.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & os,
            const Room & R )
```

prints Room's base info (number and bedCount)

Parameters

os	output stream
R	Room to be printed

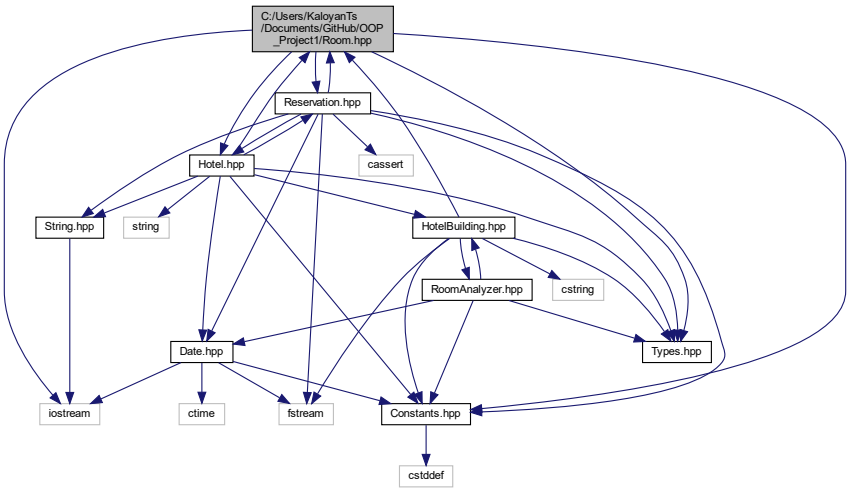
Returns

std::ostream& same output stream

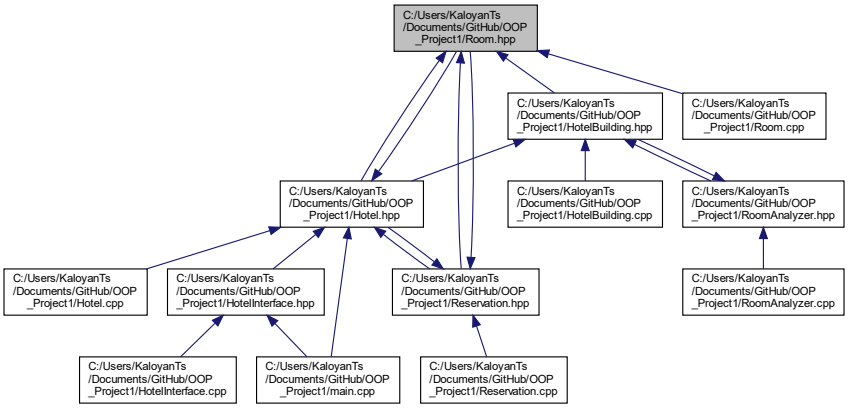
6.33 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Room.hpp File Reference

```
#include <iostream>
#include "Types.hpp"
#include "Reservation.hpp"
#include "Hotel.hpp"
#include "Constants.hpp"
```

Include dependency graph for Room.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class Room

  Class representing a room in hotel.

Functions

- std::ostream & operator<< (std::ostream &os, const Room &R)

  prints Room's base info (number and bedCount)

6.33.1 Function Documentation

6.33.1.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & os,
            const Room & R )
```

prints Room's base info (number and bedCount)

Parameters

os	output stream
R	Room to be printed

Returns

std::ostream& same output stream

6.34 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Room.hpp

Go to the documentation of this file.

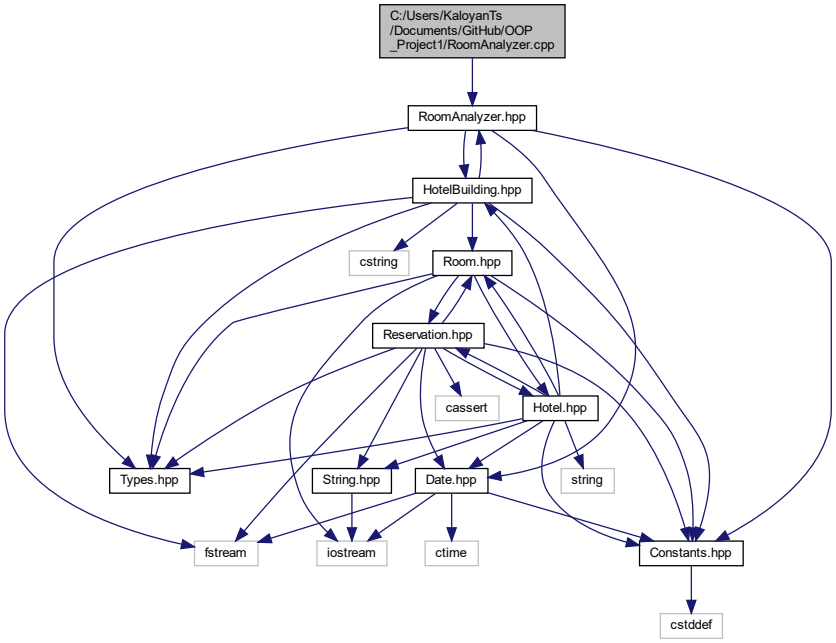
```
1 #ifndef __ROOM_HPP
2 #define __ROOM_HPP
3 #include <iostream>
4 #include "Types.hpp"
5 #include "Reservation.hpp"
6 #include "Hotel.hpp"
7 #include "Constants.hpp"
8
13 class Room
14 {
19     unsigned number;
24     unsigned bedCount;
29     Reservation **reservations;
34     size_t resCount, resCapacity;
35
40     Reservation **pastReservations;
45     size_t pastCount, pastCapacity;
46
51     void expand(Reservation **&arr, size_t &size, size_t &capacity);
56     void shrink(Reservation **&arr, size_t &size, size_t &capacity);
57
```

```
64      unsigned daysTakenInPeriod(const DatePeriod &period) const;
65
76      bool newReservation(String name, String note, const DatePeriod &period, bool service);
77
82      void moveToPast();
83
84  public:
91      Room(unsigned n, unsigned bC);
96      Room(const Room &) = delete;
101      Room &operator=(const Room &) = delete;
106      ~Room();
107
108      unsigned getNumber() const { return number; }
109      unsigned getBedCount() const { return bedCount; }
110      bool isFreeNow() const;
111
118      bool freeRoom(Reservation *&currentRes);
119
124      void newDate(Date);
125
132      bool isFreeOnDate(Date) const;
133
141      bool isFreeInPeriod(const DatePeriod &period) const;
142
151      bool showReservationsInPeriod(std::ostream &os, const DatePeriod &period) const;
152
162      bool addReservation(String name, String note, const DatePeriod &period);
163
172      bool closeForService(String note, const DatePeriod &period);
173
178      void showActivity() const;
179
185      void writeToBinaryFile(std::ofstream &ofs);
186
192      void readDataFromBinary(std::ifstream &ifs);
193  };
194
202  std::ostream &operator<<(std::ostream &os, const Room &R);
203
204  #endif
```

6.35 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/RoomAnalyzer.cpp File Reference

```
#include "RoomAnalyzer.hpp"
```

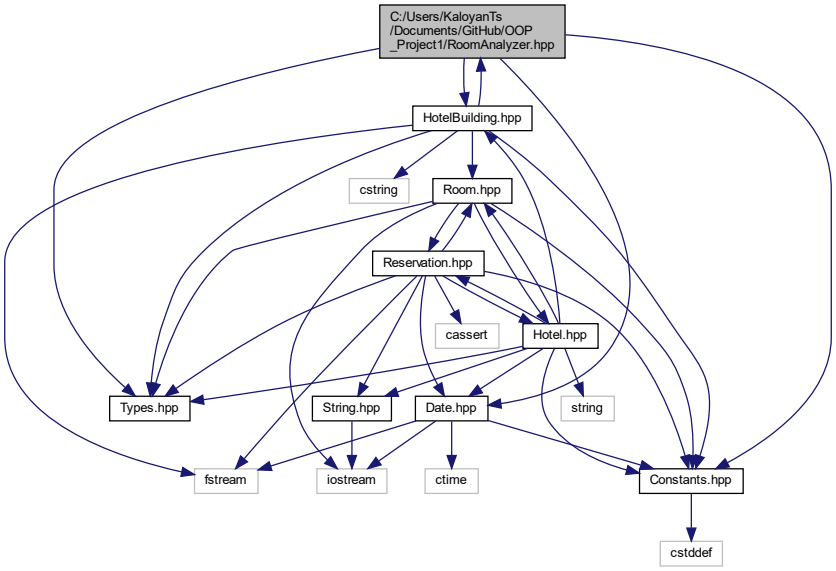
Include dependency graph for RoomAnalyzer.cpp:



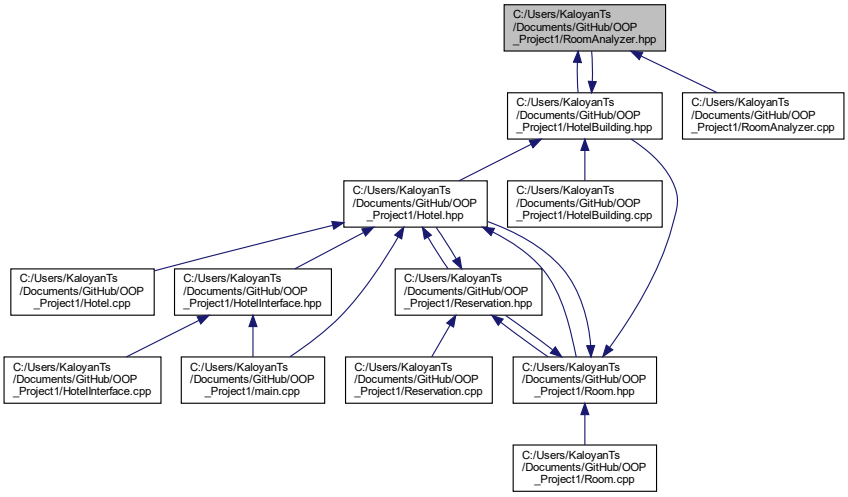
6.36 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/RoomAnalyzer.hpp File Reference

```
#include "Types.hpp"
#include "Constants.hpp"
#include "HotelBuilding.hpp"
#include "Date.hpp"
```

Include dependency graph for RoomAnalyzer.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class RoomAnalyzer

  Utility class to perform algorythms on the rooms in a building.

6.37 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/RoomAnalyzer.hpp

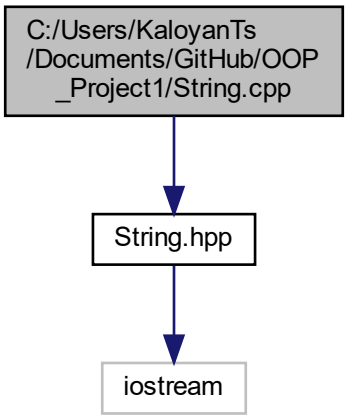
Go to the documentation of this file.

```
1 #ifndef __ROOMANALYZER_HPP
2 #define __ROOMANALYZER_HPP
3 #include "Types.hpp"
4 #include "Constants.hpp"
5 #include "HotelBuilding.hpp"
6 #include "Date.hpp"
7
9 class RoomAnalyzer
10 {
19     static void sortRoomsByScore(HotelBuilding &hB, unsigned *score, size_t from, size_t to);
20
28     static void sortRoomsByNumber(HotelBuilding &hB, size_t from, size_t size);
29
37     template <typename T>
38     static void swap(T &a, T &b);
39
40 public:
48     static void suggest(HotelBuilding &hB, unsigned beds, DatePeriod period);
49
58     static void soonestFreePeriod(const HotelBuilding &hB, unsigned number, unsigned nights, Date today);
59 };
60
61 template <typename T>
62 void RoomAnalyzer::swap(T &a, T &b)
63 {
64     T c = a;
65     a = b;
66     b = c;
67 }
68
69 #endif
```

6.38 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/String.cpp File Reference

`#include "String.hpp"`

Include dependency graph for String.cpp:



Functions

- std::ostream & operator<< (std::ostream &os, const String &S)

  print this String to the given output stream

6.38.1 Function Documentation

6.38.1.1 operator<<()

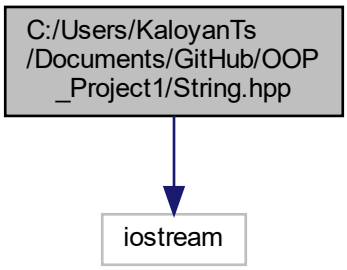
```
std::ostream & operator<< (
            std::ostream & os,
            const String & S )
```

print this String to the given output stream

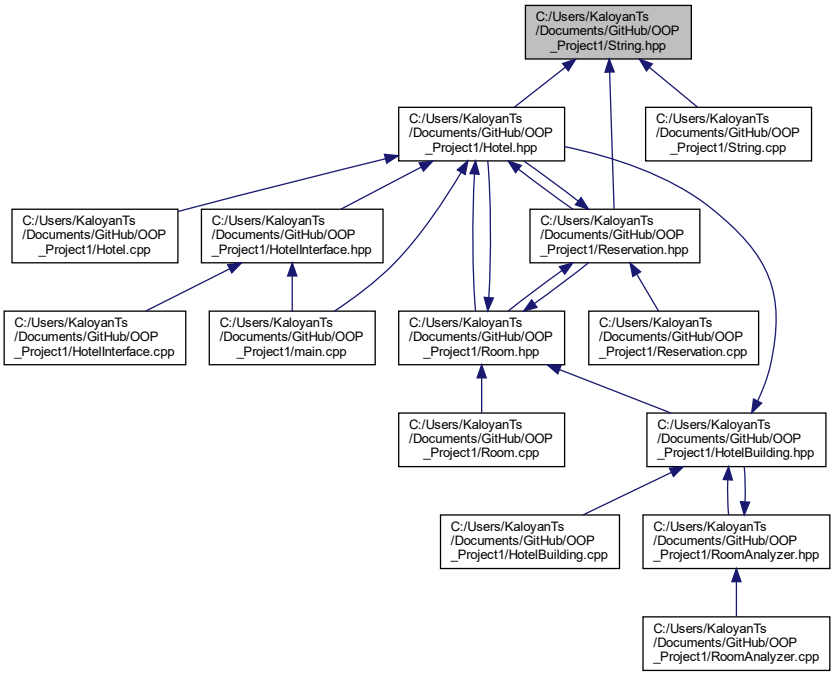
6.39 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/String.hpp File Reference

```
#include <iostream>
```

Include dependency graph for String.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class String

  Class representing dynamic string (array of characters)

Functions

- std::ostream & operator<< (std::ostream &os, const String &S)

  print this String to the given output stream

6.39.1 Function Documentation

6.39.1.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & os,
            const String & S )
```

print this String to the given output stream

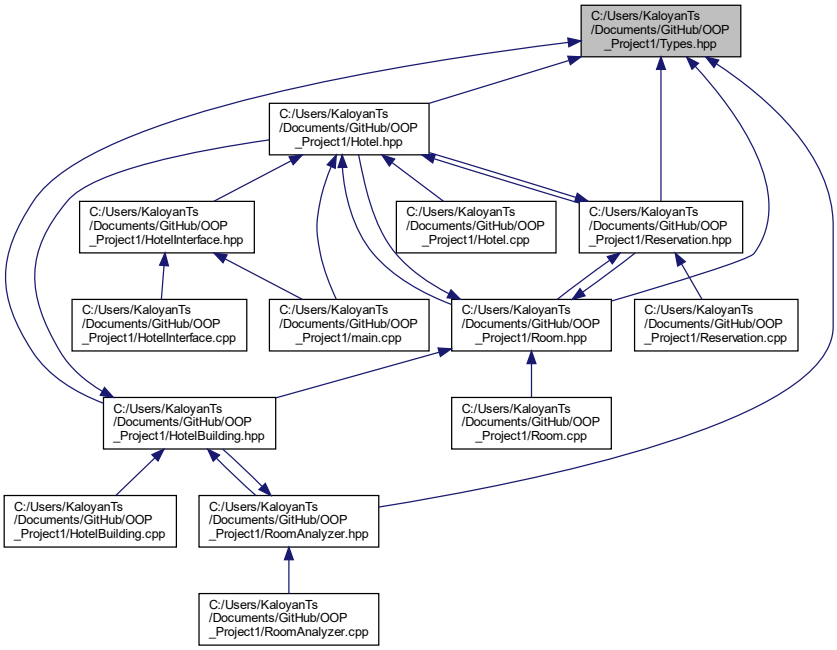
6.40 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/String.hpp

Go to the documentation of this file.

```
1 #ifndef __STRING_HPP
2 #define __STRING_HPP
3 #include <iostream>
4
6 class String
7 {
9     char *str;
10
12     size_t m_size, m_capacity;
13
19     void copy(const String &other);
20
27     void expand(size_t newCap);
28
29 public:
35     String(const char *_str = "");
41     String(const String &other);
42
49     String &operator=(const String &other);
50
55     ~String();
56
60     size_t size() const { return m_size; }
61
65     size_t capacity() const { return m_capacity; }
66
70     char *c_str() const { return str; }
71
73     String operator+(const String &other) const;
74
76     String &operator+=(const String &other);
77
79     operator const char *() const { return c_str(); }
80
82     String &shrink_to_fit();
83 };
84
86 std::ostream &operator«(std::ostream &os, const String &S);
87
88 #endif
```

6.41 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Types.hpp File Reference

This graph shows which files directly or indirectly include this file:



6.42 C:/Users/KaloyanTs/Documents/GitHub/OOP_Project1/Types.hpp

Go to the documentation of this file.

```
1 #ifndef __TYPES_HPP
2 #define __TYPES_HPP
3
4 class String;
5 class Date;
6 class Room;
7 class HotelBuilding;
8 class Reservation;
9 class RoomAnalyzer;
10 class Hotel;
11
12 #endif
```

Index