C++ Code Documentation for Event Management System

1. Classes and Their Relationships

1.1. Date

Description: Represents a calendar date.

- **Attributes**:
    - int year: The year of the date.
    - int month: The month of the date.
    - int day: The day of the date.
- **Constructors**:
    - Date(int year, int month, int day): Initializes a Date with the specified year, month, and day.
    - Date(): Default constructor for completeness.
- **Operators**:
    - Comparison operators (>=, <=, >, <, ==) for comparing dates.
    - Date operator++(int): Post-increment operator to increment the date.
    - int operator-(const Date& other) const: Computes the difference in days between two dates.
- **Friend Functions**:
    - ostream &operator<<(ostream &os, const Date &date): Outputs the date in a stream.

1.2. Time

Description: Represents a time of day.

- **Attributes**:
    - int hour: The hour of the time.
    - int minute: The minute of the time.
- **Constructors**:
    - Time(int hour, int minute): Initializes a Time with the specified hour and minute.
    - Time(): Default constructor for completeness.
- **Operators**:
    - Comparison operators (>=, >, ==) for comparing times.
    - int operator-(const Time &other): Computes the difference in minutes between two times.

- **Friend Functions**:
    - ostream &operator<<(ostream &os, const Time &time): Outputs the time in a stream.

1.3. Venue

Description: Represents a venue where programs can be held.

- **Attributes**:
    - static vector<Venue *> venues: A static list of all venues.
    - string name: The name of the venue.
    - string country: The country where the venue is located.
    - vector<string> location: The location details of the venue.
    - int capacity: The maximum capacity of the venue.
    - vector<Program *> programs: A list of programs held at the venue.
- **Constructors**:
    - Venue(string venueName, string country, vector<string> location, int capacity): Initializes a Venue with the given parameters.
- **Destructor**:
    - virtual ~Venue(): Destructor to clean up resources.
- **Methods**:
    - static int showVenues(vector<string> &inputLocation, string opt = ""): Displays venues based on location and optional filter.
    - static int searchVenue(string ven, string country, Venue *&ptr): Searches for a venue by name and country.
    - int checkReservation(Date startDate, Date endDate): Checks if the venue is available for reservation within the specified date range.
    - virtual int checkProgram(string progType) = 0: Pure virtual function to check if the venue can host a program of a specific type.
    - int reserveVenue(Program *prog): Reserves the venue for a given program.
    - int freeVenue(Program *prog): Frees the venue from a given program.
- **Friend Functions**:
    - ostream &operator<<(ostream &os, const Venue &ven): Outputs the venue details in a stream.

1.4. Congregate

Description: Represents a general congregation such as a conference, game, concert, or convention.

- **Attributes**:
    - static vector<Congregate *> congregates: A static list of all congregations.
    - string name: The name of the congregation.
    - Date startDate: The start date of the congregation.
    - Date endDate: The end date of the congregation.
    - vector<Program *> programs: A list of programs associated with the congregation.
    - string type: The type of the congregation.
- **Constructors**:
    - Congregate(string name, Date startDate, Date endDate): Initializes a Congregate with the specified name, start date, and end date.
- **Destructor**:
    - virtual ~Congregate(): Destructor to clean up resources.
- **Methods**:
    - virtual int addProgram(string programType, string programName, Date startDate, Date endDate): Adds a program to the congregation (implementation varies by derived classes).
    - int deleteProgram(Program *prog): Deletes a specified program from the congregation.
    - int showPrograms(): Displays the list of programs in the congregation.
    - int searchProgram(string progName, Program *&prog): Searches for a program by name.
    - int showReserved(): Shows reserved programs.

1.5. Program

Description: Represents a program held within a congregation at a venue.

- **Attributes**:
    - Date startDate: The start date of the program.
    - Date endDate: The end date of the program.
    - Congregate *congregation: Pointer to the congregation hosting the program.
    - vector<Venue *> venues: List of venues where the program is held.
    - string name: The name of the program.
    - string type: The type of the program.
- **Constructors**:

    - Program(Date start, Date end, Congregate *congregation, string name, string type): Initializes a Program with the given parameters.

- **Destructor**:
    - ~Program(): Destructor to clean up resources.
- **Methods**:
    - int showReserved(): Displays reserved venues for the program.

1.6. Derived Venue Classes

- **Stadium**: Abstract base class derived from Venue. Represents general stadiums with specific types.
- **Outdoor_stadium**: Derived from Stadium. Represents outdoor stadiums.
- **Indoor_stadium**: Derived from Stadium. Represents indoor stadiums.
- **Swimming_pool**: Derived from Stadium. Represents swimming pools.
- **Hotel**: Derived from Venue. Represents hotels.
- **Convention_centre**: Derived from Venue. Represents convention centers.
- **Concert_hall**: Derived from Venue. Represents concert halls.

Each derived class overrides the checkProgram method to specify the types of programs they can accommodate.

2. Utility Functions

- **int getDetails(vector<string> &arr)**: Function to obtain details (e.g., from user input or a file).
- **int daysInYear(int year, int month, int day)**: Calculates the number of days in a specific year, month, and day.
- **bool Leap(int year)**: Determines if a year is a leap year.
- **int MonthDays(int year, int month)**: Returns the number of days in a given month of a year.
- **int isValidDate(string date, Date *&ptr)**: Validates a date string and initializes a Date object if valid.
- **int checkAddress(string address, vector<string> &, bool flag = true)**: Validates an address, potentially using a flag for additional checks.

Class UML Diagram:-


Date
+int year
+int month
+int day
+bool operator>=(Date)
+bool operator<=(Date)
+bool operator>(Date)
+bool operator<(Date)
+bool operator==(Date)
+int operator-(Date)
+Date operator++(int)
+ostream &operator<<(Date)
Time
+int hour
+int minute
+bool operator>=(Time)
+bool operator>(Time)
+bool operator==(Time)
+int operator-(Time)
+ostream &operator<<(Time)
Program
+Date startDate
+Date endDate
+Congregate *congregation
+string name
+string type
+int showReserved()
0..* venues
0..* programs
1 congregation
0..* programs
Venue
+string name
+string country
+vector<string> location
+int capacity
+string type
+vector<Program *> programs
+int reserveVenue(Program)
+int freeVenue(Program)
+int checkReservation(Date, Date)
+virtual int checkProgram(string) = 0
+static int showVenues(vector<string> &, string opt = "")
+static int searchVenue(string ven, string country, Venue *&ptr)
Congregate
+string name
+Date startDate
+Date endDate
+string type
+vector<Program *> programs
+int addProgram(string, string, Date, Date)
+int deleteProgram(Program)
+int showPrograms()
+int searchProgram(string, Program)
+int showReserved()
Stadium
+string type
+int checkProgram(string)
Hotel
+int checkProgram(string)
Convention_centre
+int checkProgram(string)
Concert_hall
+int checkProgram(string)
Conference
+int addProgram(string, string, Date, Date)
Games
+int addProgram(string, string, Date, Date)
Concerts
+int addProgram(string, string, Date, Date)
Convention
+int addProgram(string, string, Date, Date)
Outdoor_stadium
+int checkProgram(string)
Indoor_stadium
+int checkProgram(string)
Swimming_pool
+int checkProgram(string)