//cpp code for event management system (Structured enquiry)

#include<iostream>

#include<stdlib.h>

using namespace std;

class Event

{

public:

int eventId;

string title;

string location;

static int counter; // number of events

double totalCharges;

string status;

// Parameterized constructor

Event(int id,string t,string l,string stat):eventId(id),title(t),location(l),status(stat),totalCharges(0)

{

counter++; //counts the number of events

}

virtual void computeCharges()

{

}

virtual void printEventDetails()

{

cout << "Event id = " << eventId << endl;

cout << "Event title = " << title << endl;

cout << "Location = " << location << endl;

cout << "Total charges = " << totalCharges << endl;

cout << "Event Status = " << status << endl;

}

};

int Event::counter = 0;

class EventSystem

{

public :

string name;

double totalRevenue;

Event \*events[]; // pointer array to all the events

// Parameterized constructor

EventSystem(string n):name(n),totalRevenue(0){}

//function to compute total revenue generated by all the events

void computeRevenue()

{

int num = events[0]->counter;

int i;

for(i=0;i<num;i++)

{

totalRevenue += events[i]->totalCharges;

}

}

// function used to cancel events

void cancelEvent(int id)

{

int i,num = events[0]->counter;

for(i=0;i<num;i++)

{

if(events[i]->eventId == id)

{

if(events[i]->status == "completed" || events[i]->status == "ongoing")

{

cout << "\n\nEvent completed or is ongoing, hence cannot be canceled\n\n";

break;

}

events[i]->status = "Canceled";

cout << "\n\nEVENT CANCELLED IS " << endl;

events[i]->printEventDetails();

}

}

}

void printSystemDetails()

{

cout << "\nEvent Management Name = " << name << endl;

cout << "Total Revenue generated = " << totalRevenue << endl;

cout << "Total number of events = " << events[0]->counter << endl;

int num = events[0]->counter;

int i;

for(i=0;i<num;i++)

{

cout << endl;

events[i]->printEventDetails();

}

}

};

class IndoorEvent : public Event

{

public:

int hours;

double chargePerHour;

// Parameterized constructor

IndoorEvent(int id,string t,string l,string stat,int h,double charge):Event(id,t,l,stat),hours(h),chargePerHour(charge)

{

computeCharges();

}

void computeCharges()

{

if(hours < 0 && chargePerHour < 0)

throw 10;

totalCharges = hours \* chargePerHour;

}

void printEventDetails()

{

Event::printEventDetails();

cout << "Charge per hour = " << chargePerHour << endl;

cout << "Number of hours = " << hours << endl;

}

};

class OutdoorEvent : public Event

{

public:

int days;

double chargePerDay;

OutdoorEvent(int id,string t,string l,string stat,int h,double charge):Event(id,t,l,stat),days(h),chargePerDay(charge)

{

computeCharges();

}

void computeCharges()

{

if(days < 0 && chargePerDay < 0)

throw 11.11;

totalCharges = days \* chargePerDay;

}

void printEventDetails()

{

Event::printEventDetails();

cout << "Charge per day = " << chargePerDay << endl;

cout << "Number of days = " << days << endl;

}

};

int main()

{

EventSystem es("TEDX Event Management System");

try

{

es.events[0] = new IndoorEvent(1001,"Enterprenuer minds","Mumbai","completed",12,1000);

es.events[1] = new IndoorEvent(1002,"Green India","Bangalore","ongoing",10,2000);

es.events[2] = new IndoorEvent(1003,"Innovative minds","Hubballi","booked",9,2200);

es.events[3] = new OutdoorEvent(2001,"Sports goal","Calcutta","completed",8,900);

es.events[4] = new OutdoorEvent(2002,"Save water","Delhi","ongoing",15,300);

es.events[5] = new OutdoorEvent(2003,"Atomic habits","Chennai","booked",7,2500);

es.computeRevenue();

es.printSystemDetails();

es.cancelEvent(1001);

es.cancelEvent(2003);

}

catch(double n)

{

cout << "Days or chargePerDay cannot be negative" << endl;

exit(-1);

}

catch(int n)

{

cout << "Hours or chargePerHour cannot be negative" << endl;

exit(-1);

}

return 0;

}

**OUTPUT :** -

Event Management Name = TEDX Event Management System

Total Revenue generated = 81000

Total number of events = 6

Event id = 1001

Event title = Enterprenuer minds

Location = Mumbai

Total charges = 12000

Event Status = completed

Charge per hour = 1000

Number of hours = 12

Event id = 1002

Event title = Green India

Location = Bangalore

Total charges = 20000

Event Status = ongoing

Charge per hour = 2000

Number of hours = 10

Event id = 1003

Event title = Innovative minds

Location = Hubballi

Total charges = 19800

Event Status = booked

Charge per hour = 2200

Number of hours = 9

Event id = 2001

Event title = Sports goal

Location = Calcutta

Total charges = 7200

Event Status = completed

Charge per day = 900

Number of days = 8

Event id = 2002

Event title = Save water

Location = Delhi

Total charges = 4500

Event Status = ongoing

Charge per day = 300

Number of days = 15

Event id = 2003

Event title = Atomic habits

Location = Chennai

Total charges = 17500

Event Status = booked

Charge per day = 2500

Number of days = 7

Event completed or is ongoing, hence cannot be canceled

EVENT CANCELLED IS

Event id = 2003

Event title = Atomic habits

Location = Chennai

Total charges = 17500

Event Status = Canceled

Charge per day = 2500

Number of days = 7

Process returned 0 (0x0) execution time : 0.078 s

Press any key to continue.