// main test driver for BST

// Created by RASNA RAHMAT HUSAIN

using namespace std;

#include "Shootings.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\* DEFAULT CONSTRUCTOR \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*p

Shootings::Shootings()

{

ID = 0;

title = "";

location = "";

date = "";

incidentArea = "";

target = "";

cause = "";

name = "";

fatalities = 0;

injured = 0;

age = 0;

race = "";

gender = "";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\* CONSTRUCTOR \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Shootings::Shootings(int sID, string sTitle, string sLocation, string sDate,

string sIncidentArea, string sTarget, string sCause, string sName,

int sFatalities, int sInjured, int sAge, string sRace, string sGender)

{

ID = sID;

title = sTitle;

location = sLocation;

date = sDate;

incidentArea = sIncidentArea;

target = sTarget;

cause = sCause;

name = sName;

fatalities = sFatalities;

injured = sInjured;

age = sAge;

race = sRace;

gender = sGender;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\* SETTER DEFINITIONS \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// setters to set respective data properties

// sets ID

void Shootings::setID(int sID)

{

ID = sID;

}

// sets title

void Shootings::setTitle(string sTitle)

{

title = sTitle;

}

// sets location

void Shootings::setLocation(string sLocation)

{

location = sLocation;

}

// sets date

void Shootings::setDate(string sDate)

{

date = sDate;

}

// sets incident

void Shootings::setIncidentArea(string sIncidentArea)

{

incidentArea = sIncidentArea;

}

// sets target

void Shootings::setTarget(string sTarget)

{

target = sTarget;

}

// sets cause

void Shootings::setCause(string sCause)

{

cause = sCause;

}

// sets name

void Shootings::setName(string sName)

{

name = sName;

}

// sets fatalities

void Shootings::setFatalities(int sFatalities)

{

fatalities = sFatalities;

}

// sets injured

void Shootings::setInjured(int sInjured)

{

injured = sInjured;

}

// sets age

void Shootings::setAge(int sAge)

{

age = sAge;

}

// sets race

void Shootings::setRace(string sRace)

{

race = sRace;

}

// sets gender

void Shootings::setGender(string sGender)

{

gender = sGender;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\* GETTER DEFINITIONS \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// returns ID

int Shootings::getID() const

{

return ID;

}

// returns title

string Shootings::getTitle() const

{

return title;

}

// returns location

string Shootings::getLocation() const

{

return location;

}

// returns date

string Shootings::getDate() const

{

return date;

}

// returns incidentArea

string Shootings::getIncidentArea() const

{

return incidentArea;

}

// returns target

string Shootings::getTarget() const

{

return target;

}

// returns cause

string Shootings::getCause() const

{

return cause;

}

// returns name

string Shootings::getName() const

{

return name;

}

// returns fatalities

int Shootings::getFatalities() const

{

return fatalities;

}

// returns injured

int Shootings::getInjured() const

{

return injured;

}

// returns age

int Shootings::getAge() const

{

return age;

}

// returns race

string Shootings::getRace() const

{

return race;

}

// returns gender

string Shootings::getGender() const

{

return gender;

}

/\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\* OVERLOADING OPERATORS \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

bool Shootings::operator == (const Shootings& toyData)

{

return ID == toyData.getID();

}

bool Shootings::operator < (const Shootings& toyData)

{

return ID < toyData.getID();

}

bool Shootings::operator > (const Shootings& toyData)

{

return ID > toyData.getID();

}

ostream& operator << (ostream& os, const Shootings& toyData)

{

os << toyData.getID() << " " << toyData.getName() << " "

<< toyData.getAge() << " " << toyData.getPrice() << endl;

return os;

};

\*/