**RELATIONSHIPS BETWEEN CLASSES**

****

****

****

**Complete code**

**Example from book: PersonType**

#include <string>

#include <iostream>

using namespace std;

**class dateType**

**{**

**private:**

**int dMonth; //variable to store the month**

**int dDay; //variable to store the day**

**int dYear; //variable to store the year**

**public:**

**void setDate(int month, int day, int year);**

**int getDay() const;**

**int getMonth() const;**

**int getYear() const;**

**void printDate() const;**

**dateType(int month = 1, int day = 1, int year = 1900);**

**dateType(const dateType &);**

**};**

**void dateType::setDate(int month, int day, int year)**

**{**

**dMonth = month;**

**dDay = day;**

**dYear = year;**

**}**

**int dateType::getDay() const**

**{**

**return dDay;**

**}**

**int dateType::getMonth() const**

**{**

**return dMonth;**

**}**

**int dateType::getYear() const**

**{**

**return dYear;**

**}**

**void dateType::printDate() const**

**{**

**cout << dMonth << "-" << dDay << "-" << dYear;}**

**dateType::dateType(int month, int day, int year)**

**{**

**dMonth = month;**

**dDay = day;**

**dYear = year;**

**}**

**dateType::dateType(const dateType &B)**

**{**

**dMonth = B.dMonth;**

**dDay = B.dDay;**

**dYear = B.dYear;**

**}**

**class personType**

**{**

**private:**

**string firstName; //variable to store the first name**

**string lastName; //variable to store the last name**

**public:**

**void print() const;**

**void setName(string first, string last);**

**string getFirstName() const;**

**string getLastName() const;**

**personType(string first = "", string last = "");**

**personType(const personType &); //copy constructor**

**};**

**void personType::print() const**

**{ cout << firstName << " " << lastName;}**

**void personType::setName(string first, string last)**

**{ firstName = first;**

**lastName = last;}**

**string personType::getFirstName() const**

**{ return firstName;}**

**string personType::getLastName() const**

**{ return lastName;}**

**personType::personType(const personType & B)**

**{**

**firstName = B.firstName;**

**lastName = B.lastName;**

**}**

**personType::personType(string first, string last)**

**{**

**firstName = first;**

**lastName = last;**

**}**

**CONTAINER CLASS**

**class personalInfo**

**{**

**private:**

**personType name;**

**dateType bDay;**

**int personID;**

**public:**

**personalInfo(string first = "", string last = "",**

**int month = 1, int day = 1, int year = 1900,**

**int ID = 0);**

**void setpersonalInfo(string first, string last, int month,**

**int day, int year, int ID);**

**void printpersonalInfo () const;**

**};**

**CONSTRUCTOR OF CONTAINER CLASS**

**personalInfo::personalInfo(string first, string last, int month,**

**int day, int year, int ID):name(first, last), bDay(month, day, year)**

{

personID = ID;

}

Constructor with parameters of class dataType is called

Constructor with parameters of class persontype is called

**//second option**

**personalInfo::personalInfo(const personType & pName, const dataType &d, int ID):name(pName), bDay(d)**

{

personID = ID;

}

**OTHER MEMBER FUNCTIONS OF CONTAINER CLASS**

**void personalInfo::setpersonalInfo(string first, string last,**

**int month, int day, int year, int ID)**

**{**

**name.setName(first, last);**

**bDay.setDate(month,day,year);**

**personID = ID;**

**}**

**void personalInfo::printpersonalInfo() const**

**{**

**name.print();**

**cout << "'s date of birth is ";**

**bDay.printDate();**

**cout << endl;**

**cout << "and personal ID is " << personID;**

**}**

**NOTE:**

* The private data members of composed objects still cannot be accessed by the container class directly
* Container class can call the public functions of the composed class

Copy Constructor with parameters of class dataType is called

Copy Constructor with parameters of class persontype is called

int main()

{

personalInfo newStudent("William", "Jordan", 8, 24, 1963,555238911);

newStudent.printpersonalInfo();

//-------------------------------------------//

**personType pName(“William”, “Jordan”);**

**dataType dob(8,24,1963);**

personalInfo Student2(pName, dob, 555238912);

return 0;

}

**Order of constructor calls:**

First constructor of composed class is called then container class constructor is called

**Nested classes**

Class A

{

private:

int A\_val;

public:

A(int=0);

};

A::A(int a)

{

A\_val=a;

}

Class B

{

private:

A aObj;

int B\_val;

public:

B(int=0, int=0);}

B::B(int val\_for\_a, int val\_for\_b): aObj(val\_for\_a)

{ B\_val=val\_for\_b;}

Class C

{

private:

B bObj;

int C\_val;

public:

C(int=0, int=0, int=0);

};

C::C(int val\_for\_a, int val\_for\_b, int val\_for\_c):bObj(val\_for\_a, val\_for\_b)

{

val\_for\_c= C\_val;

}

int main()

{

C cobj(1,2,3);

return 0;

};

void A::print()

{

cout<<A\_val;

}

void B::print()

{

aOjb.print();//we cannot do this-> **cout<<aObj.A\_val;**

cout<<B\_val; }

void C:: print()

{

bObj.print();

cout<<C\_val;

}