**Lab3 Topic: Object Composition represents “Has a Relationship”.**

**A thought**: One class has a relationship with another class. If it sounds right that class A can should contain class B, then declare a private member of class B inside Class A.

For example, there are two classes: Date and Student, it makes sense that a student has

a degree completion date and an entryDate.

Syntax: class A

{ members;

};

class B

{

members;

**A aObject;**

};

Example:

class Date

{ public:

//member functions ;

private:

int day, month, year;

};

class Student

{ public:

//members functions;

private:

**Date completionDate;**

**Date entryDate;**

} ;

**Object Composition Concepts and Rules**

**Concept/Rule 1**: If there are two classes that are related, the class that has

another class as a member is a larger class. Based on the code

above , the Student class is larger than the Date class.

**Concept/Rule 2**: An object that is a member of the larger class (member object)

can only access its class member functions. Based on the code

above, the objects completionDate and entryDate can only access

the Date class member functions.

**Concept/Rule 3**: When an object of the larger class is instantiated with

arguments passed to an overloaded class constructor, the object

of the smaller class must be instantiated first using following

syntax:

**LargerClass OverloadedConstructor (list of parameters)**

**: memberObject( list of parameter )**

**{**

**Statements;**

**}**

**Concept/Rule 4**: A member object calls its class member functions using the

following syntax:

**memberObject.itsClassMemberFunction();**

**Your Tasks:**

1. Compile the following program “Lab3Program.cpp”. Capture a screenshot of the program output.

#include <iostream>

#include <string>

using namespace std;

//Date class definition

class Date

{ public:

Date(short d= 1, short m = 1, short y = 1900);//Default constructor with default values.

Date(Date &);// copy constructor.

void printDate();// function prototype;

private:

short day, month, year;

};

// Date Member funtion definitions

Date::Date( short d,short m, short y)

{ cout <<"In the date default constructor." << endl;

day = d;

month = m;

year = y;

}

Date::Date(Date &inDateObject)

{ cout <<"In the date copy constructor." << endl;

day = inDateObject.day;

month = inDateObject.month;

year = inDateObject.year;

}

void Date::printDate()

{ cout << day << "/"<<month <<"/" << year << endl;

}

//Student class definition.

class Student

{ public:

Student(string, string , Date, Date); //default constructor with default values.

private:

Date completionDate;

Date entryDate;

string name;

string sid;

};

//Student member function definitions

Student::Student(string n, string s, Date d1, Date d2)

: completionDate(d1), entryDate(d2)

{

cout <<" In the student constructor ."<< endl;

name = n;

sid = s;

cout <<"Student's name and SID are " << name << " " << sid << endl;

cout <<"Completion date: " ;

completionDate.printDate();

cout <<"Entry date: " ;

entryDate.printDate();

}

int main( )

{

Date startDateObject(20,9,2005);

Date endDateObject(6,15,2008);

Student("John Jackson","8300-56-1999",startDateObject,endDateObject );

system("pause"); // can be replaced with cin.ignore(); cin.get();

}

1. Analyze the statements in the main ( ) function, what happens when the following statements executes?

Date startDateObject(20,9,2005);

Date endDateObject(6,15,2008);

What happens when the following statement executes?

Student("John Jackson","8300-56-1999",startDateObject,endDateObject );

1. Comment out the statement

**:** completionDate(d1), entryDate(d2)

Then compile the program to see an error message.

**What does the error message say?**

1. What Concepts/Rules have been proven from the examining the program? Identify the codes to support your analysis.
2. Write a program that demonstrates your understandings of the stated concepts and rules about Object Composition. Create two classes for the program: Person and Job.

Suggestion: Come up with no more than 2 private members of each class. It is up to you to implement member functions on how many and what they do. The program should not be too long. Make it short and to the point. Save your program as **Assessment3ProjectYourname.cpp**. Capture a screenshot of the program output after compiled.

**What To Turn In**

Submit:

**Lab3ResultYourname.doc** and **Assessment3ProjectYourname.cpp**

Grading Rubric:

|  |  |
| --- | --- |
| Lab3 Completion of Tasks #1- #3 | 6 points |
| Lab 3 Completion of Task #4 | 4 points |
| Assessment3ProjectYourname.cpp  done in Task #5   * The program runs. * The program proves Lab 3 Concept/Rule 1. * The program proves Lab 3 Concept/Rule 2. * The program proves Lab 3 Concept/Rule 3. * The program proves Lab 3 Concept/Rule 4. | 2 points  2 Points  2 points  2 points  2 points |