**Lab4 Topic: Inheritance represents “Is a Relationship”.**

**A thought**: If it sounds right that class A is class B, for example the“Beatles” is a music

band, then declare Band as a parent or base class and Beatles as a derived class.

Syntax for Public Inheritance:

class Baseclass

{ members;

};

**class Derivedclass : public Baseclass**

{

members;

};

Syntax for Protected Inheritance:

class Baseclass

{ members;

};

**class Derivedclass : protected Baseclass**

{

members;

};

Syntax for Private Inheritance:

class Baseclass

{ members;

};

**class Derivedclass : private Baseclass**

{

members;

};

Syntax for Multiple Inheritance (a derived class inherits more than one class.)

The syntax below is for public inheritance.

class Baseclass1

{ members;

};

class Baseclass2

{ members;

};

**class Derivedclass : public Baseclass1, Baseclass2**

{

members;

};

**Inheritance Concepts and Rules**

**Concept/Rule 1**: If using **public inheritance,** public members of the base class

become public of the derived class, protected members of the

base class become protected members of the derived class, and

private members of the base class will remain private to the

base class.

**Concept/Rule 2**: If using **protected inheritance,** public and protected members

of the base class become protected of the derived class, and

private members of the base class will remain private to the

base class.

**Concept/Rule 3**: If using **private inheritance,** public and protected members

of the base class become private of the derived class, and

private members of the base class will remain private to the

base class.

**Concept/Rule 4**: When an object of the derived class is being instantiated, the

base class constructor is called before the derived class

constructor. When the object of the derived class is destroyed,

the derived class destructor is called before the base class

destructor. To call an overloaded base class constructor via

the derived class constructor use the following format:

derivedClass::derivedClass( parameter list)

**: baseClassName(** parameter list **)**

**{**

Statements;

**}**

**Concept/Rule 5**: A derived class can override a member function of its base

class by defining a derived class member function with the

same name and parameter list.

**Concept/Rule 6**: A member function of the derived class definition can contain

a call statement that calls a member function of the base class

using the following syntax:

BaseClassName**::**MemberFunctionName(parameter(s) if any.);

**Your Tasks:**

1. Compile program “Lab4Program.cpp”. Capture a screenshot of the program output.

#include <iostream>

#include <string>

using namespace std;

class Band // This is the base class/Parent class

{ public:

Band( ); // Default constructor.

Band(string);

void DisplayBand();

void SetBand();

private:

string BandName;

};

Band::Band(string n2)

{ cout << "In the overloaded constructor of Band class ." << endl;

BandName= n2;

}

void Band::SetBand()

{ fflush(stdin);

cout << "Enter a Band name " << endl;

getline(cin,BandName);

}

void Band::DisplayBand()

{

cout << "The band name is " << BandName << endl;

}

Band::Band()

{ cout << "\a\aIn the default constructor of the Band class. "<< endl;

BandName ="Rush!!";

}

class Beatles : public Band // Beattles inherits from the Band class.

{ //Beattles is a derived class.

public:

Beatles();

Beatles(string, string);

private:

string frontMan1;

};

Beatles::Beatles( string n1, string n2)

**:** Band(n2)

{ cout <<"In the Overloaded constructor of the Beatles class " << endl;

frontMan1 = n1;

}

Beatles::Beatles()

{

cout << "\aIn the default constructor of the Beatles class. " << endl;

frontMan1 = "John Lennon";

}

int main()

{

Beatles bObject; //Calls the Band class constructor first , then calls the

//default constructor of the derived class.

Beatles cObject("Paul Mccartny", "Beatle Specials");

cin.ignore();

cin.get();

return 0;

}

1. State all concepts/rules that are proven by analyzing the program output and the codes.

*Concept/Rule 1 / Concept/Rule 4*

1. Comment out the statement

**:** Band(n2)

Compile it and display the error message(s) if any.

*No error message, but the Overloaded constructor of the Beatles class is skipped and the program uses Band :: Band (string n2) which is a LOGIC ERROR.*

1. Eliminate the default constructor of the Band class. Compile it and display the error message(s) if any.

*h:\c++ class 162\the lab work\lab4\lab4\lab4programkenwagner.cpp(29) : error C2511: 'Band::Band(void)' : overloaded member function not found in 'Band'*

*h:\c++ class 162\the lab work\lab4\lab4\lab4programkenwagner.cpp(5) : see declaration of 'Band'*

*h:\c++ class 162\the lab work\lab4\lab4\lab4programkenwagner.cpp(50) : error C2512: 'Band' : no appropriate default constructor available*

1. Create a member function in the Beatles class that has the function name as DisplayBand which is the same name as the DisplayBand in the Band class. The function displays the frontMan1 name and it calls the DisplayBand() of the Band class.

void Beatles::DisplayBand()

{

//Displays the frontMan1.

// Calls the DisplayBand( ) of the Band class.

}

Compile the program and state what concept/rule has been proven.

*Concept/Rule 1 / Concept/Rule 4 / Concept/Rule 5 / Concept/Rule 6*

1. The program uses public inheritance. Test the program using protected inheritance. Does it work? Then test it using private inheritance. Does it work? Do you believe all concepts/rules have been proven?

*Both test succeeded. This proves Concept/Rule 2 / Concept/Rule 3. So at this point, ALL concepts have been proven.*

1. State some of the concepts/rules that have not been proven if any.

*None that I can find.*

1. Write a program which contains two classes: Movie and CollectionMovie. The program must illustrate all tasks that prove all concepts/rules stated in this lab.

Save your program **Assessment4ProjectYourname.cpp**. Capture a screenshot of the program output after compiled.

**What To Turn In**

Submit:

**Lab4ResultYourname.doc** and **Assessment4ProjectYourname.cpp**

Grading Rubric:

|  |  |
| --- | --- |
| Lab4 Completion of Tasks 1- 7 | 7 points |
| Assessment4ProjectYourname.cpp  done in Task #8   * The program runs. * The program proves Lab 4 Concept/Rule 1. * The program proves Lab 4 Concept/Rule 2. * The program proves Lab 4 Concept/Rule 3. * The program proves Lab 4 Concept/Rule 4. * The program proves Lab 4 Concept/Rule 5. * The program proves Lab 4 Concept/Rule 6. | 3 points  2 Points  2 points  2 points  2 points  2 points |