# **Synopsis**

This laboratory session is intended to enable you understand and apply:

*1. virtual functions*

*2. static functions*

*3. and polymorphism*.

If you have not done yet, watch the lecture before you attempt this seminar (or do both in parallel).

*Answers are provided at the end of this file.*

# Exercises

1. This problem is related to the example program described on the Lecture Slide 6. If we don’t use the ‘virtual’ keyword in the Base class, what difference do you see in the output of the program?

2. Use the following UML Diagram to demonstrate how the concept of virtual base classes is used to remove the multiple inheritance related ambiguity.



3. Imagine a publishing company that markets both ***book*** and ***audiocassette*** versions of its works. Create a class called ***publication*** that stores the title (a string) and price (type float) of a publication. From this class derive two classes: ***book***, which adds a page count (type int); and ***tape***, which adds a playing time in minutes (type float). Each of the three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display the data.

Write a main() program that creates an array of pointers to publication (similar to the example in Lecture Slide 12). In a loop, ask the user for data about a particular book or tape, and use new to create an object of type book or tape to hold the data. Put the pointer to the object in the array. When the user has finished entering the data for all books and tapes, display the resulting data for all the books and tapes entered, using a for loop and a single statement such as

pubarr[j]->putdata();

to display the data from each object in the array.

# Answers

**1.** Refer to Lecture Slide 6

**2.** See Lecture Slides 13 and 14 on how to implement above UML diagram and how to use virtual base classes.

**3.**

#include <iostream>

#include <string>

using namespace std;

**class publication**

**{**

private:

string title;

float price;

public:

virtual void getdata()

{

cout << “\nEnter title: “; cin >> title;

cout << “Enter price: “; cin >> price;

}

virtual void putdata()

{

cout << “\n\nTitle: “ << title;

cout << “\nPrice: “ << price;

}

**};**

**class book : public publication**

{

private:

int pages;

public:

void getdata()

{

publication::getdata();

cout << “Enter number of pages: “; cin >> pages;

}

void putdata()

{

publication::putdata();

cout << “\nPages: “ << pages;

}

**};**

**class tape : public publication**

**{**

private:

float time;

public:

void getdata()

{

publication::getdata();

cout << “Enter playing time: “; cin >> time;

}

void putdata()

{

publication::putdata();

cout << “\nPlaying time: “ << time;

}

};

**int main()**

**{**

publication\* pubarr[100]; //array of ptrs to pubs

int n = 0; //number of pubs in array

char choice; //user’s choice

do {

cout << “\nEnter data for book or tape (b/t)? “;

cin >> choice;

if( choice==’b’ ) //make book object

pubarr[n] = new book; // put in array

else //make tape object

pubarr[n] = new tape; // put in array

pubarr[n++]->getdata(); //get data for object

cout << “ Enter another (y/n)? “; //another pub?

cin >> choice;

}

while( choice ==’y’); //cycle until not ‘y’

for(int j=0; j<n; j++) //cycle thru all pubs

pubarr[j]->putdata(); //print data for pub

cout << endl;

return 0;

**}**