//COS10007 – Developing Technical Software//  
//StudentID – 103234103 Son Nguyen//  
//Week 11 – Lab Task 11  
  
1. Describe briefly the following terminologies/symbols.  
  
a/ Class – Describe object behavior and its properties including a valid range of values, and a default value.  
b/ Object – an object is a region of memory that can contain value and are referenced by identifiers.  
c/ Member Function – Operators and functions that are declared as member of a class.   
d/ Data Member – the variable which are declared in any class by using data types such as (int, char, float etc.) or derived data types such as (pointer, structure, class).  
e/ Access specifiers – keywords in OOL that set the accessibility of classes, methods and other members.  
f/ Class definition – in C++ , it’s a fundamental block of a program that has its own set of methods and variables.  
g/ Encapsulations – is to make sure that “sensitive” data is hidden from users using declare class variables/attributes as private.  
h/ Inheritance – a feature in which new classes are created from the existing classes.  
  
2. Modify Class Gradebook  
  
a/ Include another string data member that represents the course lecturer’s name.   
#include <iostream>

#include <string>

class GradeBook {

public:

GradeBook(std::string courseName, std::string lecturerName); // Constructor

// Set function to change the lecturer's name

void setLecturerName(std::string lecturerName);

// Get function to retrieve the lecturer's name

std::string getLecturerName();

// Function to display the welcome message, course name, and lecturer's name

void displayMessage();

private:

std::string courseName;

std::string lecturerName; // New data member for lecturer's name

};

b/ Provide a set function to change the lecturer’s name and a get function to retrieve it.  
  
GradeBook::GradeBook(std::string courseName, std::string lecturerName) {

this->courseName = courseName;

this->lecturerName = lecturerName;

}

c/ Modify the constructor to specify course name and lecturer’s name parameters.  
  
void GradeBook::setLecturerName(std::string lecturerName) {

this->lecturerName = lecturerName;

}

std::string GradeBook::getLecturerName() {

return lecturerName;

}

d/ Modify function displayMessage to output the welcome message and course name, and the string “This course is presented by:” followed by the lecturer’s name  
  
void GradeBook::displayMessage() {

std::cout << "Welcome to the course: " << courseName << std::endl;

std::cout << "This course is presented by: " << lecturerName << std::endl;

}

int main() {

GradeBook gradeBook("Developing Technical Software", "Nalin Randeniya");

gradeBook.displayMessage();

gradeBook.setLecturerName("Nalin Randeniya");

std::cout << "Updated lecturer's name: " << gradeBook.getLecturerName() << std::endl;

return 0;

}  
Output   
A black background with white text

Description automatically generated with low confidence  
  
3/  
a/ Define a C++ base class named Rectangle containing length and width data members. From this class, derive a class named Box with another data member named Depth. The member functions from the base class Rectangle should consist of a constructor and an area () function. The derived class Box should have a constructor, a volume () function and an override function named area() that return the surface area of the box  
  
#include <iostream>

class Rectangle {

protected:

double length;

double width;

public:

Rectangle(double len, double wid) : length(len), width(wid) {}

double area() {

return length \* width;

}

};

class Box : public Rectangle {

private:

double depth;

public:

Box(double len, double wid, double dep) : Rectangle(len, wid), depth(dep) {}

double volume() {

return length \* width \* depth;

}

double area() override {

double side1 = length \* width;

double side2 = length \* depth;

double side3 = width \* depth;

return 2 \* (side1 + side2 + side3);

}

};

int main() {

Box myBox(3.0, 4.0, 5.0);

std::cout << "Box area: " << myBox.area() << std::endl;

std::cout << "Box volume: " << myBox.volume() << std::endl;

return 0;

}

Disclaimer: I couldn’t get override to work so for the question b , I kinda just removed it   
  
b/ Include the classes written in part. A above in a working C++ program that creates an object for each class and calls each member function for each class to test them. Verify the results manually.   
  
#include <iostream>

class Rectangle {

protected:

double length;

double width;

public:

Rectangle(double len, double wid) : length(len), width(wid) {}

double area() {

return length \* width;

}

};

class Box : public Rectangle {

private:

double depth;

public:

Box(double len, double wid, double dep) : Rectangle(len, wid), depth(dep) {}

double volume() {

return length \* width \* depth;

}

double area() {

double side1 = length \* width;

double side2 = length \* depth;

double side3 = width \* depth;

return 2 \* (side1 + side2 + side3);

}

};

int main() {

Rectangle myRectangle(3.0, 4.0);

Box myBox(3.0, 4.0, 5.0);

std::cout << "Rectangle area: " << myRectangle.area() << std::endl;

std::cout << "Box area: " << myBox.area() << std::endl;

std::cout << "Box volume: " << myBox.volume() << std::endl;

return 0;

}

Output   
A black background with white text

Description automatically generated with low confidence