W11\_Qn1

a. Class: A class is a blueprint or template for creating objects in object-oriented programming. It specifies the attributes and methods of class objects. A class is a template for creating multiple instances of objects with similar properties.

b. Object: An object is a class instance. It represents a specific entity or item created with the class's blueprint. Objects have distinct states and behaviors that are defined by the class to which they belong to.

c. Member function: A member function is a function defined within a class that operates on the data members of that class. It contains the actions or behaviors that a class object can perform. Member functions are accessed through objects and can change the state of the object.

d. Data member: A data member is a variable or attribute associated with a class. These variables contain the data or state of the objects created by the class. The class's member functions can access and modify data members.

e. Access specifiers: They are used in class definitions to determine whether class members are accessible from outside the class. The three most common access specifiers are:

* Public members can be accessed from anywhere in the program.
* Private members can only be accessed from within the class and not from outside.
* Protected members can be accessed within the class and its derived classes.

f. Classification: A class definition is a piece of code that defines a class's data members and member functions. It specifies the structure and behavior of objects deriving from that class. The declaration of data members and member functions is included in the class definition.

g. Encapsulation: It is used to combine data and the methods that operate on that data into a class. It ensures that data is only accessed and modified through the class's defined methods. Encapsulation aids in data hiding and abstraction, allowing for greater control over the internal implementation of the class.

h. Inheritance: It is a fundamental concept that allows one class to inherit properties and behaviors from another. The class that inherits is known as the derived class or subclass, while the class from which it is inherited is known as the base class or superclass. Inheritance allows for code reuse and promotes class hierarchy, as subclasses can add to or override the functionality of the base class.

W11\_Qn3.cpp

#include <iostream>

#include <string>

using namespace std;

class GradeBook

{

public:

GradeBook(string cName, string tName)

{

setCourseName(cName);

setTeacherName(tName);

}

void setCourseName( string name )

{

courseName = name;

}

void setTeacherName(string name)

{

teacherName=name;

}

string getCourseName()

{

return courseName; // return object's courseName

}

string getTeacherName()

{

return teacherName;

}

void displayMessage()

{

cout << "Welcome to the grade book for " << getCourseName()<< "!" << endl;

cout<<"The course is presented by "<<getTeacherName()<<"!" <<endl;

}

private:

string courseName, teacherName;

};

int main()

{

GradeBook gradeBook1( "COS10007 Developing Technical Software", "Peter");

GradeBook gradeBook2( "COS10008 Foundation of Technical Programming" ,"Alex");

cout << "GradeBook1 created for course: "<<gradeBook1.getCourseName()<<endl;

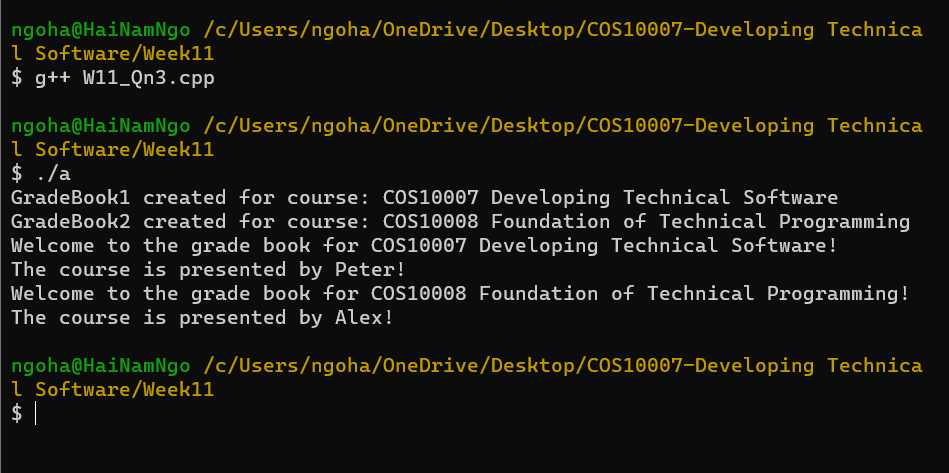
cout << "GradeBook2 created for course: "<< gradeBook2.getCourseName()<< endl;

gradeBook1.displayMessage();

gradeBook2.displayMessage();

}

OUTPUT FOR W11\_Qn3.cpp



W11\_Qn4.cpp

#include<iostream>

#include<cmath>

using namespace std;

class Rectangle

{

protected:

double length,width;

public:

Rectangle(double l=1.0, double w=1.0)

{

length=l;

width=w;

}

double area()

{

return length\*width;

}

};

class Box : public Rectangle

{

protected:

double height;

public:

Box(double l=1.0, double w=1.0, double h=1.0): Rectangle(1,w), height(h){}

double volume()

{

return height\*Rectangle::area();

}

double area()

{

return 2\*((length\*width)+(length\*height)+(width\*height));

}

};

int main()

{

Rectangle r1,r2(3,4);

Box b1, b2(3,4,5);

cout<<"The area of rectangle 1 is "<<r1.area()<<endl;

cout<<"The area of rectangle 2 is "<<r2.area()<<endl;

cout<<"The volume of Box 1 is "<<b1.volume()<<endl;

cout<<"The volume of Box 2 is "<<b2.volume()<<endl;

cout<<"The area of Box 1 is "<<b1.area()<<endl;

cout<<"The area of Box 2 is "<<b2.volume()<<endl;

}

OUTPUT FOR W11\_Qn4.cpp

A picture containing text, screenshot, font

Description automatically generated