Week 6:Operator Overloading

Learning Materials: Chapter 8

Task 0

Goto page 351-356 copy the code times1.cpp and times2.cpp . Run the codes in your IDE and understand how the cast operator works. You need to explain later.

Task 1

The `StudentCollection` class represents a collection of student records.
It has two member variables:

names: This member variable is an array that stores the names of up to 10 students in the collection.
grades: This member variable is also an array and stores the corresponding grades for the students. The grades are stored in the same order as the names.

The class provides the following member functions:

operator[] (string& name) for reading: You can retrieve a student's grade by providing their name as an argument. It searches through the `names` array to find the student with the matching name and returns their grade.
operator[] (string& name) for writing: You can modify a student's grade by providing their name and the new grade. It searches for the student by name and updates their grade. You can use the same function to do both task.

addStudent (string name, double grade): You can add a new student. It takes the student's name and grade as arguments and adds them to the `names` and `grades` arrays, respectively. If the collection is already full (contains 10 students), it throws an exception.

In the `main()` function, demonstrate the usage of the `StudentCollection` class by adding students, accessing their grades, and updating grades.

Task 2

You need to implement a `Weight` class where you can convert Weight to double and double to Weight.
It has three member variables:
kgToPound: A constant double representing the conversion factor from KG to Pound
pound: An integer variable to store the number of pounds in the Weight class.
ounce: A double variable to store the decimal number in the Weight class.

The class provides the following member functions:
Weight()
Weight(double kg): converts it to pound and ounce.
Weight(int pound, double ounce): sets the Weight accordingly.
showWeight(): Displays the weight in the format pound-ounce.
A conversion operator operator double() that converts a Weight object to kg.

In the main() function, the code demonstrates the following:
Create a Weight object using a single-argument constructor and display it in pound-ounce format. Then convert that object to kg and display the result in KG. Create another Weight object using a two-argument constructor. Convert it to KG and display the result in KG.

Task 3

You will implement a Currency Conversion calculator.
Three classes are defined:
CurrencyBDT represents an amount in BDT and poisha. It has two private member variables: taka and poisha.
CurrencyDollar represents an amount in US Dollars and cents. It has two private member variables: dollars and cents.

CurrencyEuro represents an amount in Euros and cents. It also has two private member variables: euros and cents.
The classes have constructors to initialize the currency amounts. The default constructor sets the amounts to zero, and the parameterized constructor is used to specify the amount in taka/dollars/euros and poisha/cents.
The CurrencyBDT class has two conversion operators: The first operator is used to convert an amount in Taka to an equivalent amount in US Dollars. It takes into account the conversion rate (1 BDT = 0.0091 USD) and performs the necessary calculations to convert the amount, including cents. The second conversion is used to BDT to Euro (1 BDT = 0.0085 Euro)

In the main() function, create an object of CurrencyBDT to show both conversions.