CMP-245 Object Oriented Programming Lab
BS Fall 2018
Lab 11

Issue Date: 06-Dec-2019
Marks: 16

Objective:

- It will help you understand the benefits that we get through inheritance relationship.
- It will also help in comparing inheritance and composition but we may also look in detail about them in next few lectures.

Challenge A:

(6)

The class 'Counter' is behaving as a counter which increment only but doesn't decrement the counter. Provide a new class named as 'FBCounter', which doesn't only increment but decrement the counter as well.

You must reuse the 'Counter' class while implementing the new class by providing two version of 'FBCounter' class:

- One using whole-part relationship
- Second using inheritance relationship

Note: You are not allowed to change anything in class 'Counter'.

```
class Counter
{
private:
    int value;
public:
    Counter(int i=0):value(i)
    {}
    void increment()
    { value++; }
    void reset()
    { value=0; }
    void startAt(int i)
    { value = i; }
    int getCounterValue()
    { return value; }
};
```

Sample Run

```
int main()
{
    FBCounter c;
    c.increment();
    c.increment();
    c.decrement();
    c.increment();
    cout<<c.getCounterValue()<<"\n";
    return 0;
}
```

Sample Output

```
2
```

Challenge B:

(10)

Package-delivery services, such as FedEx®, DHL® and UPS®, offer a number of different shipping options, each with specific costs associated. Create an inheritance hierarchy to represent various types of packages. Use Package as the base class of the hierarchy, then include classes TwoDayPackage and OvernightPackage that derive from Package. Base class Package should include data members representing the name, address, city, state and ZIP code for both the sender and the recipient of the package (idea: A person class should have this information like name, address, city, state, zip and then compose it in Package class), in addition to data members that store the weight (in ounces) and cost per ounce to ship the package. Package's constructor should initialize these data members. Ensure that the weight and cost per ounce contain positive values. Package should provide a public member

function calculateCost that returns a double indicating the cost associated with shipping the package. Package's calculateCost function should determine the cost by multiplying the weight by the cost per ounce. Derived class TwoDayPackage should inherit the functionality of base class Package, but also include a data member that represents a flat fee that the shipping company charges for two-day-delivery service. TwoDayPackage's constructor should receive a value to initialize this data member. TwoDayPackage should redefine member function calculateCost so that it computes the shipping cost by adding the flat fee to the weight-based cost calculated by base class Package's calculateCost function. Class OvernightPackage should inherit directly from class Package and contain an additional data member representing an additional fee per ounce charged for overnight-delivery service. OvernightPackage should redefine member function calculateCost so that it adds the additional fee per ounce to the standard cost per ounce before calculating the shipping cost.
Write a test program that creates objects of each type of Package and tests member function calculateCost.

"In looking for people to hire, you look for three qualities: integrity, intelligence, and energy. And, if they don't have the first, the other two will kill you."

-- Warren Buffett --

Success is not the absence of failure; it's the persistence through failure.

-- --