San Francisco Bay University

CS360 - Programming in C and C++
Homework Assignment #5

Due day: 4/04/2024

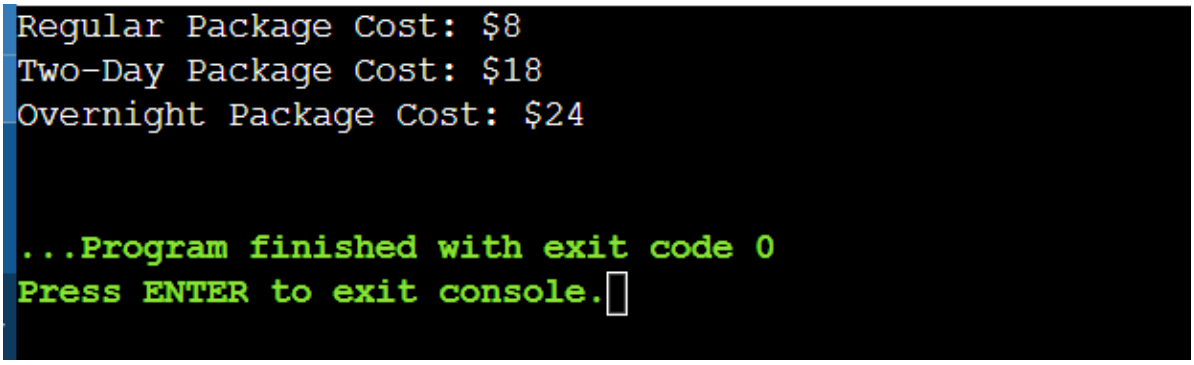
Instruction:

1. Push the answer sheets/source code to Github
2. Please follow the code style rule like programs on handout.
3. Overdue homework assignment submission can't be accepted.
4. Take academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)

1. 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 class *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, 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 *main* program that creates objects of each type of *Package* and tests member function *calculateCost*.

Here is the Output and the main code is in the cpp file



2. A supermarket chain has asked you to develop an automatic checkout system. All products are identifiable by means of a barcode and the product name. Groceries are either sold in packages or by weight. Packed goods have fixed prices. The price of groceries sold by weight is calculated by multiplying the weight by the current price per kilo.

   Develop the classes needed to represent the products first and organize them hierarchically. The *Product* class, which contains generic information on all products (barcode, name, etc.), can be used as a base class.

   a. The *Product* class contains two data members of type *long* used for storing barcodes and the product name. Define a constructor with parameters for both data members. Add default values for the parameters to provide a default constructor for the class. In addition to the access methods *setCode()* and *getCode()*, also define the methods *scanner()* and *printer()*. For test purposes, these methods will simply output product data on screen or read the data of a product from the keyboard.

   b. The next step involves developing special cases of the *Product* class. Define two classes derived from *Product*, *PrepackedFood* and *FreshFood*. In addition to the product data, the *PrepackedFood* class should contain the unit price and the *FreshFood* class should contain a weight and a price per kilo as data members.

      In both classes define a constructor with parameters providing default-values for all data members. Use both the base and member initializer.

      Define the access methods needed for the new data members. Also redefine the methods *scanner()* and *printer()* to take the new data members into consideration.

   c. Test the various classes in a *main* function that creates two objects each of the types Product, *PrepackedFood* and *FreshFood*. One object of each type

is fully initialized in the object definition. Use the default constructor to create the other object. Test the *get* and *set* methods and the *scanner()* method and display the products on screen.

Here is the Output and the main code is in the cpp file