

Package Inheritance – Abstract Sealed

Due Date: Friday, March 18, 8:30 am

Marks: 25 (rubric)

Problem Description: Shipping companies such as Fedex and UPS offer different shipping options, each with specific costs associated. You must create an inheritance hierarchy to represent different types of packages.

Use the same code window for all classes. **Marks will be deducted if access modifiers are not included.**

Note: when coding the get and set, you must declare the field also within the class.

1. Create a console application called **LastnameLab6**. Create the class called **Client** using the UML diagram:

Client
+ Name: string + Address: string + City: string //initialize city to city where you were born, init at declaration
+ Client() // does nothing + Client(name:string, address:string) //set both to data sent + ToString():string //keyword override, return all data, , see screenshots

2. Create an abstract class **Package** using the UML diagram with explanation in comments:

abstract Package
* TrackingNumber: int // each tracking number goes up by 10 (only accessible base and derived) + From: Client // uninitialized + To: Client // uninitialized + Weight: decimal // get and set, validate for 450 – 5000, set to max if invalid (units: grams) + Price: decimal // abstract with get only (units: per gram) - static next: int = 1000
+ Package() // set the TrackingNumber using next (goes up by 10 each time), initialize From and To to new Client, set the To data to your name and 123 Elm Street as the address + Package(from: Client, to: Client, weight: decimal) //call default constructor, set all data to data sent + CalculateCost():decimal // abstract, no code body + ToString():string // keyword override , return all data formatted, type of Package, see screenshots

3. Create two sealed classes **Weekly** and **Speedy** that derive from **Package**.

sealed Weekly derived from Package
+ Discount: decimal //code get and set, validate between 10% and 25%, set to max if invalid + Price: decimal // keyword override, return .04 (represents price per gram shipped)
+ Weekly() // does nothing (calls Package default on its own) + Weekly(from: Client, to: Client, weight: decimal, discount: decimal) //call base custom send data, set discount + CalculateCost():decimal // cost is price per gram multiplied by weight in grams. // Deduct discount amount from total cost + ToString():string // keyword override, call base ToString and add discount, see screenshots
sealed Speedy derived from Package
+ Extra: decimal //code get and set, validate between 0.01 and 0.04, set to max if invalid (/ gram) + DeliveryFee: decimal //code get only, return 10.00 + Price: decimal // keyword override, return .12 (represents price per gram shipped)
+ Speedy() // does nothing (calls Package default on its own) + Speedy(from: Client, to: Client, weight: decimal, extra: decimal) //call base custom send data, set extra + CalculateCost():decimal // cost is (price per gram plus extra per gram) multiplied by //weight in grams. Add the delivery fee. + ToString():string // keyword override, call base ToString, add extra and delivery fee, see screenshots

4. Code the following in main: (label your comments on each 4a., 4b., etc.)
 - a. Attempt to create a **Package** called **p1**. Code and comment issues. Leave comment for marking.
 - b. Create a list of packages called **deliveries** in main set to an empty list. Create a method called **DisplayPackages** in the **Program** class that will display all packages. Send the list of packages to the method.
 - c. Create a **Weekly** delivery using default constructor. Add to the list. Change the **From** information to name as "NBCC", address as "100 Grand Avenue", City to "Saint John". Set the Weight to 0 to test **set** of Weight. Set the discount to 0 to test **set** of Discount. Both should get reset. See screenshots.
 - d. Create a **Weekly** delivery using custom constructor sending in From: "Bob Barker", "2 Apple Road"; To: "Mario Luigi", "80 Orange Street", weight of 450.25 and discount of 0.15 (all valid data). Add to the list.
 - e. Create a **Speedy** delivery using default constructor. Add to the list. Change the **From** information to name as "Greco", address as "8 Pizza Slices", City to "Fredericton". Set the Weight to 5000.2 to test **set** of Weight. Set the extra cost to 0 to test **set** of Extra. Both should get reset. See screenshots.
 - f. Create a **Speedy** delivery using custom constructor sending in From: "Haresh Komali", "17 Grape Juice"; To: "Amazon", "22 Cloud", weight of 899.9 and extra cost of 0.035 (all valid data). Add to the list.
 - g. Call **DisplayPackages** sending in the list of packages (only called once at end). All formatting of display is done in **ToString()** methods. Update **DisplayPackages** to sum the total cost of all and display on screen.
5. Notice that **Package**, **Weekly** and **Speedy** are all performing the same validation check within a **set** of a property. Fix this so that there is a method found in the **Package** class that is called from all **set** properties when validation is needed. Be careful with data sent and returned and the access modifier of this new method.
6. Change **DisplayPackage** to **also** write to a file named **Lastname.txt**. Add a header in the file with your name and today's date. Read two strings from the user: path and filename. Validate the path exists. If not display an error. If it does, write all data to the file. Include the total of all packages in file.

Submission: (3 files)

1. Memo: include memo discussing problems and solutions as well as approximate time to complete. Copy all code here. Highlight class definitions and method signatures in **yellow**.
2. Zip entire program and submit.
3. Text file created.

Screenshots:

Data from output screen:

Enter path including drive (for example c:/data): c:/files

Enter filename only (for example report.txt): last.txt

c:/files/last.txt created or Error-path does not exist, file not created

Type	Number	From
Weekly	1000	NBCC, 100 Grand Avenue, Saint John
Ship To	----->	Your name, 123 Elm Street, Halifax
		Weight Price Cost
		5,000.00 \$0.04 \$150.00

1000 has discount: 25.0%

Type	Number	From
Weekly	1010	Bob Barker, 2 Apple Road, Halifax
Ship To	----->	Mario Luigi, 80 Orange Street, Halifax
		Weight Price Cost
		450.25 \$0.04 \$15.31

```

1010 has discount: 15.0%
*****
Type      Number  From
Speedy    1020    Greco, 8 Pizza Slices, Fredericton
Ship To ---->    Your name, 123 Elm Street, Halifax
                  Weight      Price      Cost
                  5,000.00    $0.12    $810.00
*****
1020 has extra cost per gram: $0.040 and delivery fee of $10.00
*****
Type      Number  From
Speedy    1030    Haresh Komali, 17 Grape Juice, Halifax
Ship To ---->    Amazon, 22 Cloud, Halifax
                  Weight      Price      Cost
                  899.90    $0.12    $149.48
*****
1030 has extra cost per gram: $0.035 and delivery fee of $10.00

Total cost of all packages is $1,124.79

```

Picture of last.txt

```

last.txt - Notepad
File Edit Format View Help
Your Name, Date: 3/8/2022
*****
Type      Number  From
Weekly    1000    NBCC, 100 Grand Avenue, Saint John
Ship To ---->    Your name, 123 Elm Street, Halifax
                  Weight      Price      Cost
                  5,000.00    $0.04    $150.00
*****
1000 has discount: 25.0%

*****
Type      Number  From
Weekly    1010    Bob Barker, 2 Apple Road, Halifax
Ship To ---->    Mario Luigi, 80 Orange Street, Halifax
                  Weight      Price      Cost
                  450.25    $0.04    $15.31
*****
1010 has discount: 15.0%
Ln 7, Col 59  100%  Windows (CRLF)  UTF-8

```