Faculty of Arts and Sciences
Department of Computer Science
CMPS 212 – Intermediate Programming with Objects
Asst 1 – Due on June 23rd @ midnight

Problem 1

Create the following classes in Java:

Ornament Class:

It is Christmas season! You are eager to decorate your Christmas tree with various types of decorations. One of the favorite types of decorations of ornaments are the sphere like colored balls. In order to do that, you have to create an *Ornament* class that describes an ornament object. An *ornament* is one that has a color ("Red", "Green", or "Gold"), a size ('S', 'M', 'L'), and a price (a double) in $. Add to your class the necessary constructors and destructor (if deemed necessary), setters and getters, a toString function and a counter that keeps track of how many Ornament objects have been created.

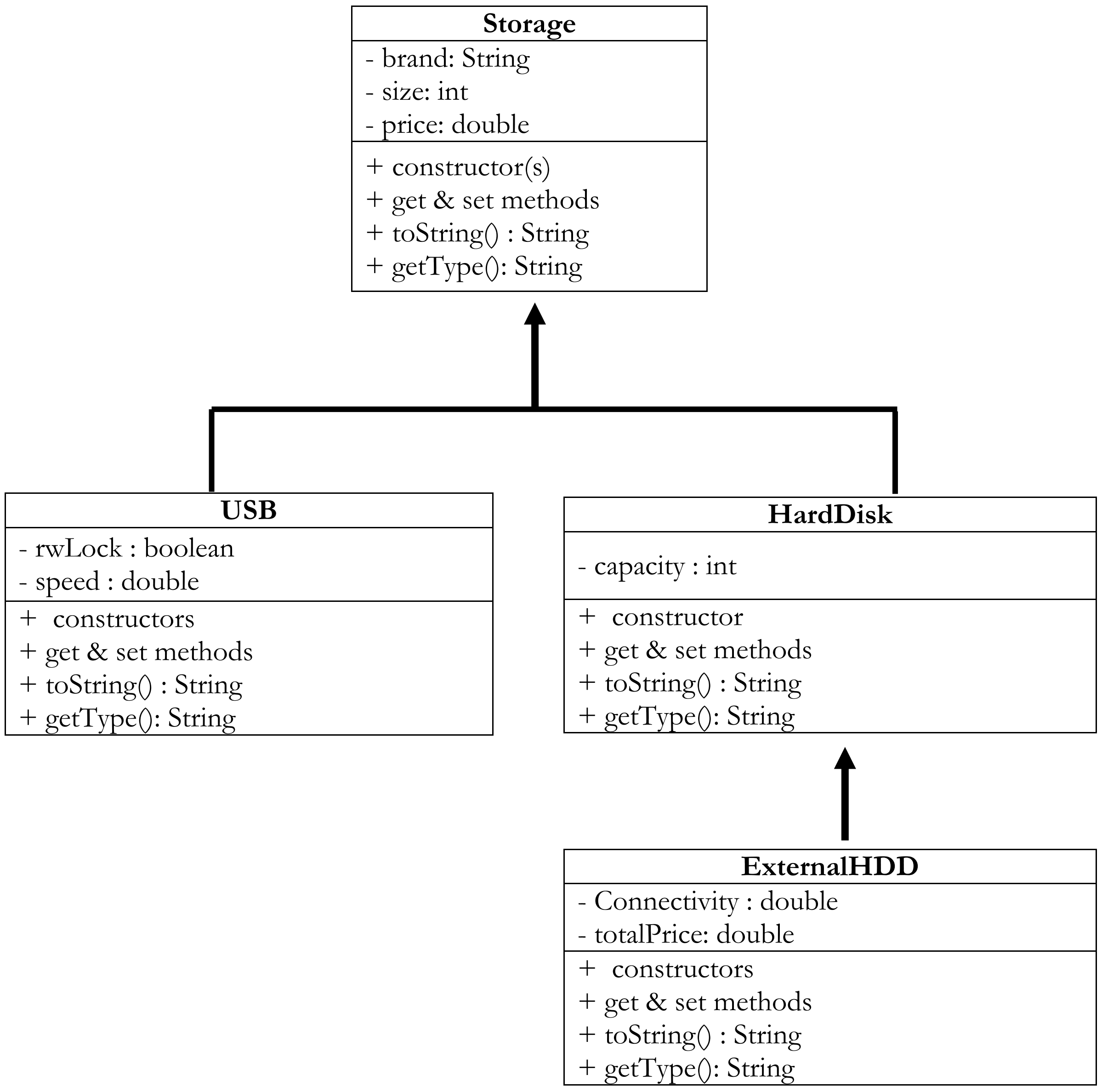
Light Class:
Lights are the key element in illuminating your Christmas tree. Usually lights come on wire strands. Your class should allow the user to choose a color for the wire ("Green" or "White"), a length for the wire strand (in meters), whether the lights are LED or not, and a price (double) in $. Again, add the necessary constructors and destructor (if deemed necessary), setters and getters and a toString function. Allow call cascading in all of your setter functions.

ChristmasTree class
Your Christmas tree is characterized by its height (in centimeters), its type ("Pine", "Cedar", and "Fir"), its ornaments (an array of ornaments whose default size is 25 but should also allow the user to determine that), and its Light. Yet again, add the necessary constructors and destructor (if deemed necessary), its setter and getters, and the toString function.

Driver class
Your driver should create an array of Christmas trees and allow the user to fill it. It should allow the following:

A. Sort the Christmas tree in ascending order of height. Use an enhanced for loop to show the result.
B. Change the length of the Light of each Christmas tree and its color according to the following (use call cascading):
    a. If the Christmas tree type is "Pine" then its height should be 10 and its color should be "Green".
    b. If the Christmas tree type is "Cedar" then its height should be 15 and its color should be "White".
    c. If the Christmas tree type is "Fir" then its height should be 20 and its color should be "Gree".
C. Print the total number of Ornament objects created in your driver.
D. Show how many ornament objects are of Small in size.
E. Print the details of the most expensive Christmas trees.

Problem 2
Implement it using Java the above hierarchy of classes.



The **Storage** class is an abstract class that has the following members:
Attributes (declared as private)*:*

- brand: a String to store the name of the brand.
- size: an integer that denotes the capacity in GB
- price: a double

Methods:

- Constructors
- get & set methods
- toString( ): a method that returns a description of the Storage class
- getType(): an **abstract** method that returns the type of storage

The **USB** is a concrete class derived from the Storage class. It has the following:
Attributes (declared as private)*:*

- rwLock: a Boolean that tells whether the USB key has a read/write lock or not
- speed: an integer that represents the speed of the USB key

Methods:

- Constructors
- get & set methods (allow call cascading in the setters)
- toString( ): a method that returns a description of the USB class.
- getType(): a method that returns the String "Solid-State storage device"

The **HardDisk** is a concrete class derived from the Storage. It has the following:
Attributes (declared as private)*:*

- SSD: a boolean that tells whether the HDD is of type SSD or Magnetic (true denotes the former and false denotes the latter)

Methods:

- Constructor.
- get & set methods
- toString( ): a method that returns a description of the **HardDisk**.
- getType(): this function returns the String "Magnetic Storage device" if SSD is true and "SSD Storage device" otherwise.

The **ExternalHDD** is a concrete class derived from the HardDisk. It has the following:
Attributes (declared as private)*:*

- connectivity: an integer representing the type of connectivity supported (1 for USB and 2 for Firewire)
- totalPrice: a double representing the TOTAL price of ALL ExternalHDD objects created

Methods:

- Constructor.
- get & set methods
- toString( ): a method that returns a description of the **ExternalHDD**.
- getType(): this function returns the String "External Magnetic Storage device".

Create a driver program called **StorageDriver** that does the following:

a) Create an array of **Storage** objects of size N. Fill it from the user. Mind that the user should choose which type of object he/she would like to create.
b) Print the total price of all objects of type **ExternalHDD**.
c) Decrease the price of the all **Storage** objects whose size is greater than 500MB by %20.
d) Sort the array in ascending order of size. Print the result using an enhanced for loop.

Problem 3

Modify the two previous exercises by introducing an Interface called *setup* that is implemented by both, the *ChristmasTree* and the *Storage* classes. Note that a Christmas Tree is setup by being assembled together (the equivalent of returning a message that reads "Assembling in progress…") and a Storage object is setup by being installed (the equivalent of returning a message that reads "Arriving at some port…")