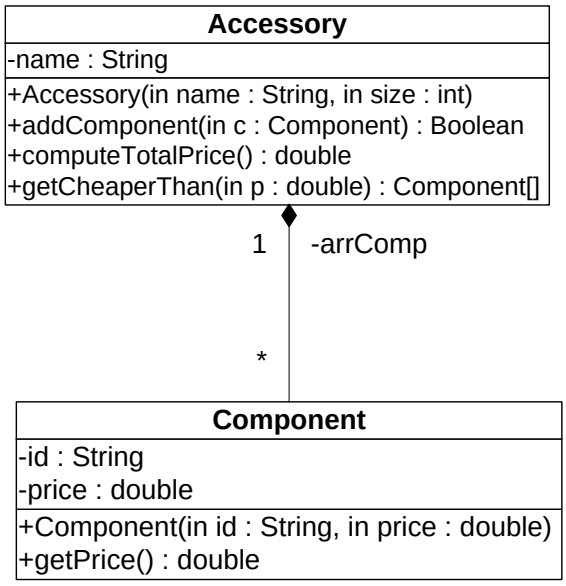
King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II Mid Term 1 Exam – Fall 2013

Exercise 1:



Accessory class***:***

- Attributes:
  - ***name***: the name of the accessory.
- Methods:
  - ***Accessory(name: String, size: int)***: constructor
  - ***addComponent(c: Component)***: this method adds the component ***c*** to the accessory. It returns true if the component ***c*** is added; false otherwise.
  - ***computeTotalPrice( )***: this method returns the total price of all the components of the accessory.
  - ***getCheaperThan(p: double***): this method returns an array of components having a price less than the given price ***p***.

Component class

- Attributes:
  - ***id***: the id of the component.
  - ***price***: the price of the component.
- Methods:
  - ***Component(id: String, price: double)***: constructor.
  - ***getPrice( )***: this method returns the price of the component.

QUESTION: Translate into Java code the class ***Component***, and the class ***Accessory***.

KSU/CCIS/CS	CSC 113	MT1 Fall 2013
Student Name:		**Section No:**
Student Id:		**Instructor:**

Answer Exercise 1:

/5

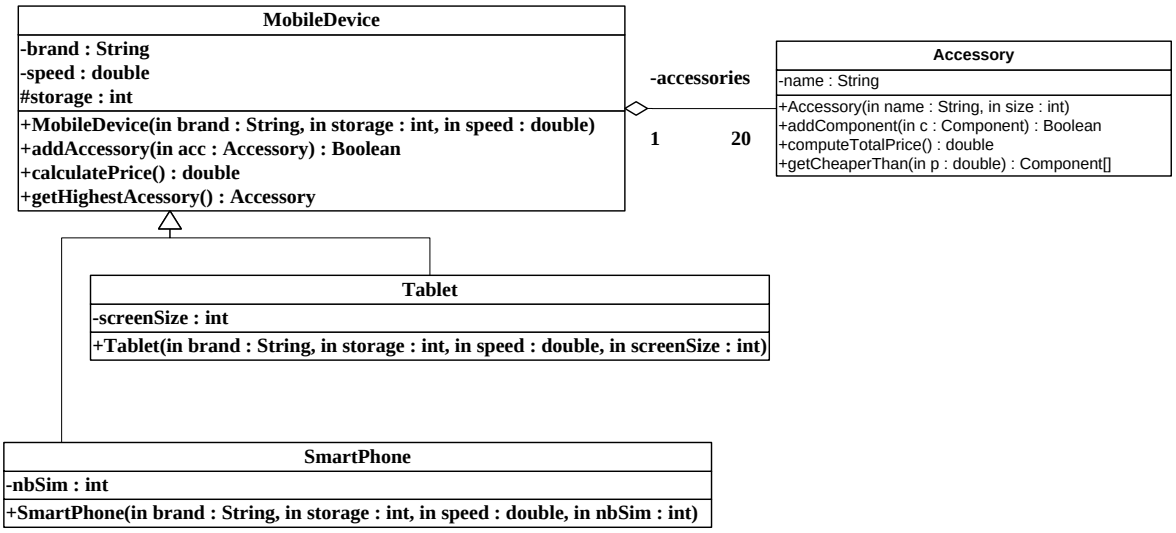
```
public class Component {
	private String ID;
	private double price;

	public Component(String id, double p) {-----1
		ID = id;
		price = p;
	}

	public Component(Component c) {--------1
		ID = c.ID;-----------------------1
		price = c.price;-----------------1
	}

	public double getPrice()  {
		return price;--------------------1
	}

}
```

/20

```
public class Accessory {
	private String name;
	private Component arrComp[];---------1
	private int nbComp;------------------1

	public Accessory(String x, int size) {
		name = x;
		arrComp = new Component[size];-------1
		nbComp = 0;--------------------------1
	}

	public boolean addComponent(Component c) {
		if (nbComp < arrComp.length) {-------------1
				arrComp[nbComp] = new Component(c);--------1
				nbComp ++;---------------------------------1
				return true;----------------------0.5
		}
		else
			return false;--------------------------0.5
	}

	public double computeTotalPrice() {
		double total = 0.0;----------------------1

		for (int i = 0; i < nbComp; i++) --------1
```

```
                total += arrComp[i].getPrice();-----1

        return total;----------------------------1
    }

    public Component[] getCheaperThan(double p) {--------1
        Component array[] = new Component[nbComp];------1
        int j = 0;-------------------------------------1

        for (int i=0; i < nbComp; i++) {----------------1
            if (arrComp[i].getPrice() < p) {----------1
                array[j] = arrComp[i];--------------1
                j++;--------------------------------1
            }
        }

        return array;----------------------------------1
    }

}
```

Exercise 2:



MobileDevice class***:***

- Attributes:
  - ***brand***: the brand of the mobile device.
  - ***speed:*** the speed of the mobile device.
  - ***storage***: the storage capacity of the mobile device.
- Methods:
  - ***MobileDevice(brand: string, storage: int, speed: double)***: constructor
  - ***addAccessory(acc: Accessory):*** this method adds an accessory ***acc*** to the mobile device. It returns true if the accessory ***acc*** is added; false otherwise.
  - ***calculatePrice( ):*** this method calculates and returns the price of the mobile device. The price is calculated as follows:
    - ***For Smart Phone:***
      - ***price = 1300 + number of sim cards * 150***
    - ***For Tablet:***
      - ***price = 1300 * storage+ screen size * 50***
  - ***getHighestAccessory():***this method returns the accessory having the highest total price.

SmartPhone class

- Attributes:
  - ***nbSim***: the number of SIM cards.
- Methods:
  - ***SmartPhone (brand: string, storage: int, speed: double, nbSim: int)***: constructor.

Tablet class:

- Attributes:
  - ***screenSize***: the screen size of the tablet.
- Methods:
  - ***Tablet (brand: string, storage: int, speed: double, screenSize: int)***: constructor.

QUESTION: Translate into Java code the class ***MobileDevice***, and the class ***SmartPhone***.

KSU/CCIS/CS	CSC 113	MT1 Fall 2013
Student Name:		**Section No:**
Student Id:		**Instructor:**

Answer Exercise 2:

```
public abstract class MobileDevice {--------1          /15
      private String brand;
      private double speed;
      protected int storage;
      private Accessory accessories[];--------1
      private int nbAcc;----------------------1

      public MobileDevice(String b, int st, double sp) {
            brand = b;
            speed = sp;
            storage = st;

            accessories = new Accessory[20];---------1
            nbAcc = 0;-------------------------------1
      }

      public boolean addAccessory(Accessory acc) {
            if (nbAcc < accessories.length) {----------1
                  accessories[nbAcc] = acc;------------1
                  nbAcc ++;----------------------------1
                  return true;-------------------------0.5
            }
            else
                  return false;------------------------0.5
      }

      public abstract double calculatePrice();-----------1


      public Accessory getHighestAccessory() {
            Accessory max = accessories[0];------------------1

            for (int i = 1; i < nbAcc; i++) {----------------1
                  if (accessories[i].computeTotalPrice() >
max.computeTotalPrice())-------------------------------------1
```

```
                        max = accessories[i];----------------1
                }

                return max;-----------------------------------1
        }
}
```

```
public class SmartPhone extends MobileDevice {-----------1     /4
        private int nbSim;

        public SmartPhone(String b, int st, double sp, int nbS) {
                super(b, st, sp);---------------------------------1
                nbSim = nbS;
        }

        public double calculatePrice() {-------------------------1
                return (1300 + nbSim * 150 );----------------------1
        }

}
```

Exercise 3:

Write a class ***Application*** that contains a ***main()*** method to do the following statements in the given order:

- Create an ***Accessory*** object AC**C1** named **"headset"** that contains the following 2 components.
    - Component 1: id: "1111", price: 25.00
    - Component 2: id: "2222", price: 15.00

- Create an ***Accessory*** object AC**C2** named **"memory card"** that contains the following component.
    - Component 1: id: "3333", price: 55.0

- Create a Smart phone object **SP (brand is "Nokia", storage is 8, speed is 120, number of sim cards is: 2)** that has 2 accessories ACC1 and ACC2.

- Add the accessories to the Smart phone.

- Display the price of the smart phone.

KSU/CCIS/CS	CSC 113	MT1 Fall 2013
Student Name:		**Section No:**
Student Id:		**Instructor:**

Answer Exercise 3:

```
public class Application {          /6

	public static void main(String[] args) {
		Accessory acc1 = new Accessory("HeadSet", 2);-----------0.5
		Component c1;
		c1 = new Component("111", 25.0);------------------------0.5
		acc1.addComponent(c1);----------------------------------0.5

		c1 = new Component("222", 15.0);------------------------0.5
		acc1.addComponent(c1);----------------------------------0.5

		Accessory acc2 = new Accessory("Memory Card", 1);------0.5
		c1 = new Component("333", 55.0);------------------------0.5
		acc2.addComponent(c1);----------------------------------0.5

		SmartPhone sp = new SmartPhone("Nokia", 8, 120.0, 2);----0.5
		sp.addAccessory(acc1);----------------------------------0.5
		sp.addAccessory(acc2);----------------------------------0.5

		System.out.println("Price is :" + sp.calculatePrice());---0.5

	}

}
```