The University of the West Indies, St. Augustine
COMP 2603 Object Oriented Programming 1
Week 2, Lab 2

Objects and Classes

1. Write the code for an **AirConditioner** class with the following attributes:
   - **brand** (String) - default is null
   - **location** (String) - default is null
   - **acID** (integer) - default is 0
   - **isOn** (boolean) - default is false (AC is off)
   - **units** (String) - default is Celsius
   - **mode** (String) - default is Cool
   - **roomTemperature** (double) - default is 24C
   - **desiredTemperature** (double) - default is 18C

   Ensure that information hiding principles are not violated!

2. Write a simple no-argument constructor for the **AirConditioner** class that initialises state of the attributes of the class.

3. Write **accessor** methods for the following attributes:
   - **brand** (String)
   - **location** (String)
   - **acID** (integer)
   - **isOn** (boolean)
   - **units** (String)
   - **mode** (String)
   - **roomTemperature** (double)
   - **desiredTemperature** (double)

4. Write **mutator** methods for the following attributes:
   - **units** (String) - value is either Fahrenheit or Celsius (default)
   - **mode** (int) - input value is either 1, 2 or 3 which corresponds to Fan, Dry or Cool respectively. (Hint: research switch statements in Java for a quick implementation)

   For all methods, print error messages if the input does not conform to the ranges specified.

5. Write the following methods to control the **AirCondition** object which return **true** if successful and **false** otherwise.
   - **turnOff ( ): Boolean**
   - **turnOn( ): Boolean**

6. Write a **toString( )** method that returns the aggregated state of an **AirConditioner** object. The method signature is as follows: **public String toString( )**

7. Write a main class called **CoolingSystem** with the following structure:

```
public class CoolingSystem{
     public static void main(String[ ] args){
          //code for steps 7 and onward goes here
     }
}
```

Ensure that the **CoolingSystem** class and the **AirConditioner** class are in the same folders.

8. In the **CoolingSystem** class:

Brand	Location	acID	Turned on
Carrier	Atrium	34	No
Lennox	Lobby	93	No
Trane	Kitchen	67	Yes

(a) Create three **AirConditioner** objects with the state specified in the table above.
(b) Create an **AirConditioner** array and add the **AirConditioner** objects to the array.
(c) Iterate through the array and print the state of each **AirConditioner** object in the array by invoking the **toString( )** method on each object.

9. Write a private method with the following method signature:
private double convertCelsiusToFahrenheit (double celciusValue)
This method should convert a Celsius value to Fahrenheit.

10. Write a private method with the following method signature:
private double convertFahrenheitToCelcius(double fahrenheitValue)
This method should convert a Fahrenheit value to Celsius.

11. Write a **mutator** method for the following attribute in the **AirConditioner** class:
    - **desiredTemperature** (double) - allowable range is 15 to 30 C otherwise the default is used.

    This method should cater for the user entering a value that is in either Celsius or Fahrenheit and doing the appropriate conversion.

12. Change the details of the **AirConditioner** objects as follows by invoking the appropriate methods of the class:

acID	Turned on	Units	Mode	DesiredTemperature
34	Yes	C	Cool	19.5
93	Yes	F	Fan	16.3
67	No	C	Dry	24.0

Iterate through the array and print the state of each **AirConditioner** object in the array.

13. Modify your **AirConditioner** classes so that the **AirConditioner** constructor initializes the state variables using input parameters of your choice. Modify the instance creation in the **CoolingSystem** class as appropriate.
    a. How does this impact on your work from step 8 ?

Additional Exercise:

Write a method with the signature **public void regulateTemperature( )** in the **AirCondition** class that simulates the temperature regulation. It should test whether the room temperature is at the desired temperature. If the room temperature is higher or lower, the method should use a simple loop to decrement or increment the room temperature in steps of 0.5C. Each step should printout the following message: “Cooling to “.. along with temperature changes until the desired temperature is reached.