CENG 305
Object Oriented Programming With Java
Fall 2013
HOMEWORK 1

1. Regulations

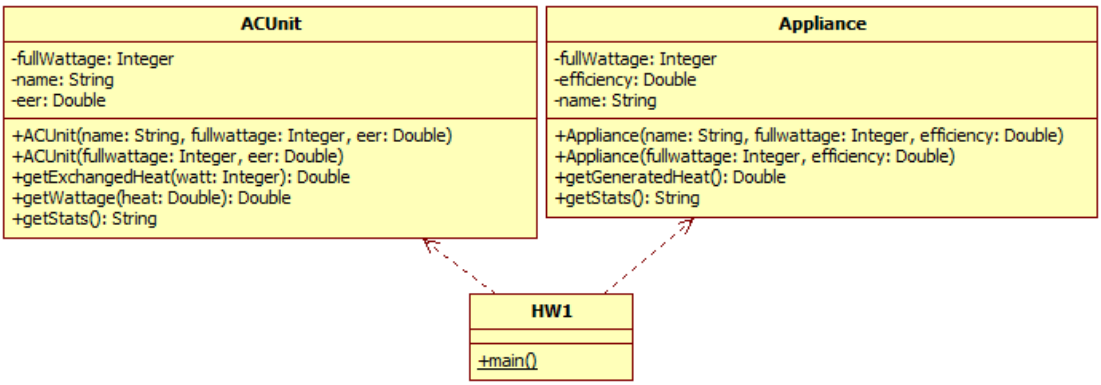
- Due Date: 24 / 11 /2013
- Programming Language: You will use Java (version 1.6) to implement the assignment.
- Submission: Submit your homework through Cow (https://cow.ceng.metu.edu.tr).You will submit a ZIP file named "e1xxxxxx.zip", which will include "Appliance.java" file, "ACUnit.java" file and a folder that has all .html files generated by javadoc.
- The homework should be done and submitted individually.
- Late Submission: You are allowed to submit your homework late with a penalty of 5*days*days.
- Cheating: The homework has to be done individually. In case of cheating, all involved (source(s) and receiver(s)) will get zero.
- Newsgroup: Check the metu.ceng.course.305 for discussions, announcements, etc. about the homework, regularly.
- Grading: Black box method will be used for evaluating and grading your assignments. Hence, please check sure that your codes can be compiled and run with the adequate output before submitting them. Furthermore, adequate indentation of the code, documentation and following submission rules are important since they will be considered in grading.

2. Specification

Data centers or server farms are at least a room full of servers and networking equipment that use electricity and generate heat. As you can imagine, a data center for Google or Facebook is comparable to a factory for its energy usage and heat generation.

An aerial view of the two of Facebook data centers in Forest City, North Carolina(left) and Lulea, Sweden(right)

Starting with this homework you will write a system that will calculate the heat generation and air conditioning needs for a simple model of a data center. Do not worry over the physics (or the units) of the calculations, they are clearly stated in the function definitions. You will start with the base classes of your system `Appliance` class and the `ACUnit` class. Your package name have to be `CENG305`



Appliance class

Attributes

fullWattage: Integer Energy needed for the appliance in an hour in watts.
name: String Name of the appliance.
efficiency: Double Efficiency of the appliance. Will be used in heat calculation for this appliance. This takes values between 0 and 1.

Operation

Appliance(name: String, fullWattage: Integer, efficiency:Double) Assigns the parameters to the appropriate attributes.
Appliance(fullWattage: Integer, efficiency: Double) Assigns the parameters to the appropriate attributes. Here, name is assigned to be 'NA'.
getGeneratedHeat(): Double This function returns the heat generated in an hour by the appliance. It is calculated as $fullWattage \times \frac{1}{efficiency} \times 3.41$ and its unit is BTU.
getStats(): String Generates and returns a string in the following format where italic words signifies the values held in corresponding attributes:

Name: *name*
Watt: *fullWattage*
Efficiency: *efficiency*

ACUnit Class

Attributes

fullWattage: Integer ***Energy*** needed for the AC unit in an hour in watts.
name: String Name of the AC unit.
eer: Integer Efficiency of the AC unit. This has a special unit defined specifically for the AC units. It is the ratio of output cooling (in BTU) to input electrical power (in Wh) at a given operating point.

Operation

ACUnit(name: String, wattage: Integer, eer:Double) Assigns the parameters to the appropriate attributes.
ACUnit(wattage: Integer, eer: Integer) Assigns the parameters to the appropriate attributes. Here, name is assigned to be 'NA'.
getExchangedHeat(Integer: watt): Double This function returns the heat removed from the room in an hour by the AC unit. It is calculated as $watt \times eer$ and its unit is BTU.
getWattageNeeded(heat: Double): Double This function returns the power need by the AC unit to remove *heat* amount of heat from the room. It is calculated as $heat/eer$ and its unit is watt/h.

getStats(): String Generates and returns a string in the following format where italic words signifies the values held in corresponding attributes:

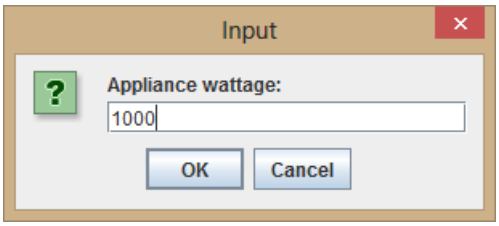
Name: *name*
Watt: *fullWattage*
EER: *eer*

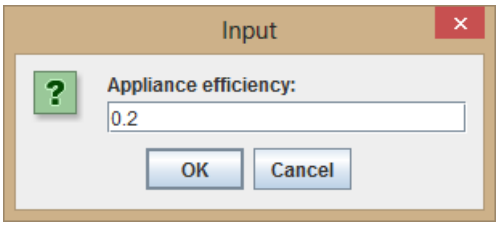
HW1 Class

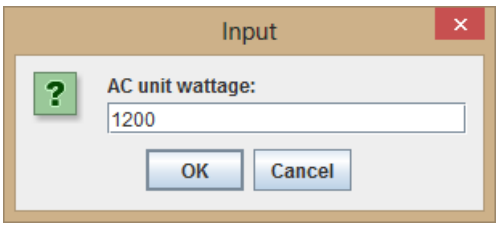
This class will only contain the main function. The details will be given in the next section.

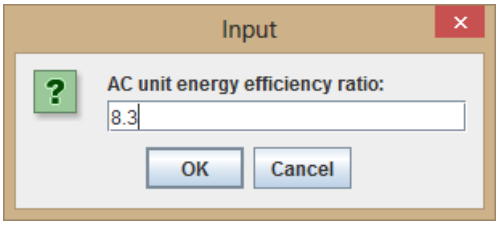
3. Program Flow

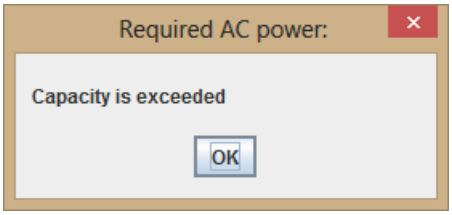
In the first example, we input the numbers for an appliance which consumes 1000 watts of power for an hour and has 0.2 efficiency. AC unit consumes 1200 watts of power and has EER rating of 8.3. Program outputs that the unit is not sufficient to keep the room temperature stable, meaning the heat generated by the appliance is higher than the capacity of the AC unit.



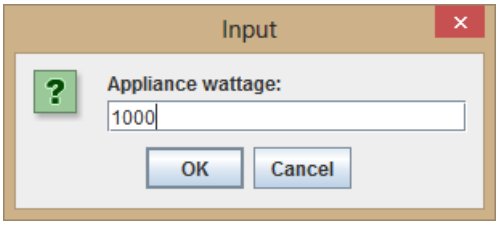


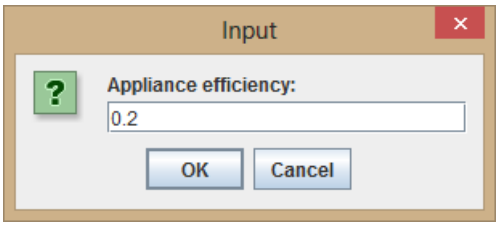


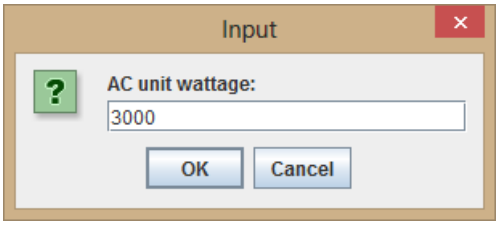


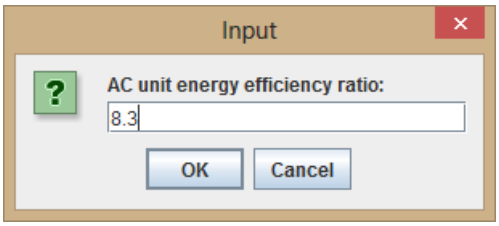


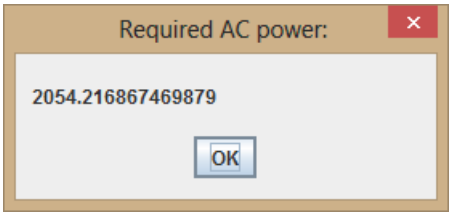
In the second example we enter the same values for the appliance but this time the AC unit is a 3000 watt unit. Program outputs the power needed by the AC unit to keep the room temperature stable.











4. Submission

You will submit a single exxxxxxx.zip file. File name have to be your user name. Only your class files ("Appliance.java" and "ACUnit.java") have to be in this .zip file. All the package, class, attribute and operation names have to be **exactly** the same with the names given in this document. If you do not follow the submission rules a penalty will be applied to your grade.