

Assignment 1

Late homework assignments will not be accepted, unless you have a valid written excuse (medical, etc.). You must do this assignment alone. No team work or "talking with your friends" will be accepted. No copying from the Internet. Cheating means zero.

Create a new Eclipse workspace named "**Assignment1_1234567890**" on the desktop of your computer (replace **1234567890** with your student ID number). For each question below, create a new project in that workspace. Call each project by its question number: "**Question1**", "**Question2**", etc. Answer all the questions below. At the end of the assignment, create a ZIP archive of the whole workspace folder. The resulting ZIP file must be called "**Assignment1_1234567890.zip**" (replace **1234567890** with your student ID number). Upload the ZIP file on iSpace.

Question 1

You are a software engineer who has to write the software of an air conditioner (AC). The AC has three modes: cool mode, heat mode, and dry mode. The software is designed to be able to turn on / off the AC, set the mode and the temperature.

Write a **AirConditioner** class with the following UML specification:

```
+-----+
|               AirConditioner               |
+-----+
| - powerOn: boolean                          |
| - mode: int                                |
| - temperature: int                          |
+-----+
| + AirConditioner()                          |
| + turnOn(): void                            |
| + turnOff(): void                           |
| + getPower(): boolean                       |
| + setMode(int mode): void                   |
| + getMode(): int                            |
| + setTemperature(int temp): void             |
| + getTemperature(): int                     |
| + testAirConditioner(): void                |
+-----+
```

where:

- **powerOn** is a private instance variable describing whether the power of the AC is on or not. While true means power on, false means power off.
- **mode** is a private instance variable describing the current mode of the AC. To be specific, the codes for the cool mode, the heat mode and the dry mode are 0, 1, 2, respectively. Any other code is invalid.
- **temperature** is a private instance variable describing the temperature in degree Celsius that is set for the AC.
- **AirConditioner** is a public constructor that creates an **AirConditioner** object. When an AC is created, its power is off, mode is cool, and temperature is 16.
- **turnOn** is a public method that turns the AC on.
- **turnOff** is a public method that turns the AC off.

- **getPower** is a public method that returns the status of the power.
- **setMode** is a public method that sets the mode of the AC to *mode*.
 - If the AC is power off, this function should print "**Please turn on the AC first**" and return.
 - If the code is invalid, **setMode** should print "**Invalid mode**" and return.
- **getMode** is a public method that returns the current mode of the AC.
- **setTemperature** is a public method that set the temperature of the AC to *temp*.
 - If the AC is power off, this function should print "**Please turn on the AC first**" and return.
 - If the AC is power on but is on the dry mode, **setTemperature** should print "**Cannot set the temperature in the dry mode**" and return.
 - The valid range of temperature is from 16 to 30 degrees. If *temp* is invalid, **setTemperature** should print "**Invalid temperature**" and return.
- **getTemperature** is a public method that returns the current temperature of the AC.
- **testAirConditioner** is a public static method that tests all the code in your **AirConditioner** class. Test all your methods from the simplest one first to the most complicated one last.

Once you have written the **AirConditioner** class, you can test it by adding the following code in a separate class:

```
public class Start {
    public static void main(String[] args) {
        AirConditioner.testAirConditioner();
    }
}
```

This code calls the static **testAirConditioner** method of the **AirConditioner** class, which should then run all your tests.

Here are a few extra instructions:

- Give meaningful names to your variables so we can easily know what each variable is used for in your program.
- Put comments in your code (in English!) to explain WHAT your code is doing and also to explain HOW your program is doing it.
- Make sure all your code is properly indented (formatted). Your code should be beautiful to read.

Failure to follow these instructions will result in you losing points.