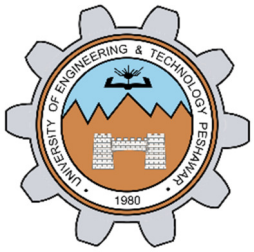
Lab Title

LAB # XX



Fall 2019

CSE208L Object Oriented Programming Lab

Submitted by: **Student Name**

Registration No. : **18PWCSE16XX**

Class Section: **A/B/C**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

Submitted to:

Engr. Sumayyea Salahuddin

October 9, 2019

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Objectives of the Lab:

Objectives of the lab are to:

- Clearly understand the purpose and advantages of OOP
- Understand the concept of a Class and Objects
- Develop a basic class containing Data Members and Member Functions
- Use access specifiers to access Class Members
- Make Simple and Overloaded Constructor
- Use the Class Objects and Member Functions to provide and extract data from Object
- Practice with Classes and Objects

Activity # 01

Title:

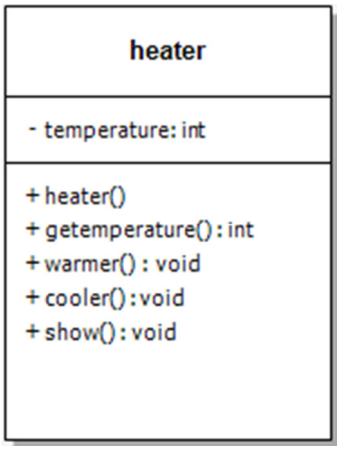
Make a class for heater and model it using temperature.

Problem analysis:

Create a class, **Heater** that contains a single integer field, **temperature**. Define a constructor that takes no parameters. The **temperature** field should be set to the value 15 in the constructor. Define the mutators **warmer** and **cooler**, whose effect is to increase or decrease the value of the temperature by 5 respectively. Define an accessor method to return the value of **temperature**. Demonstrate the use of Heater class.

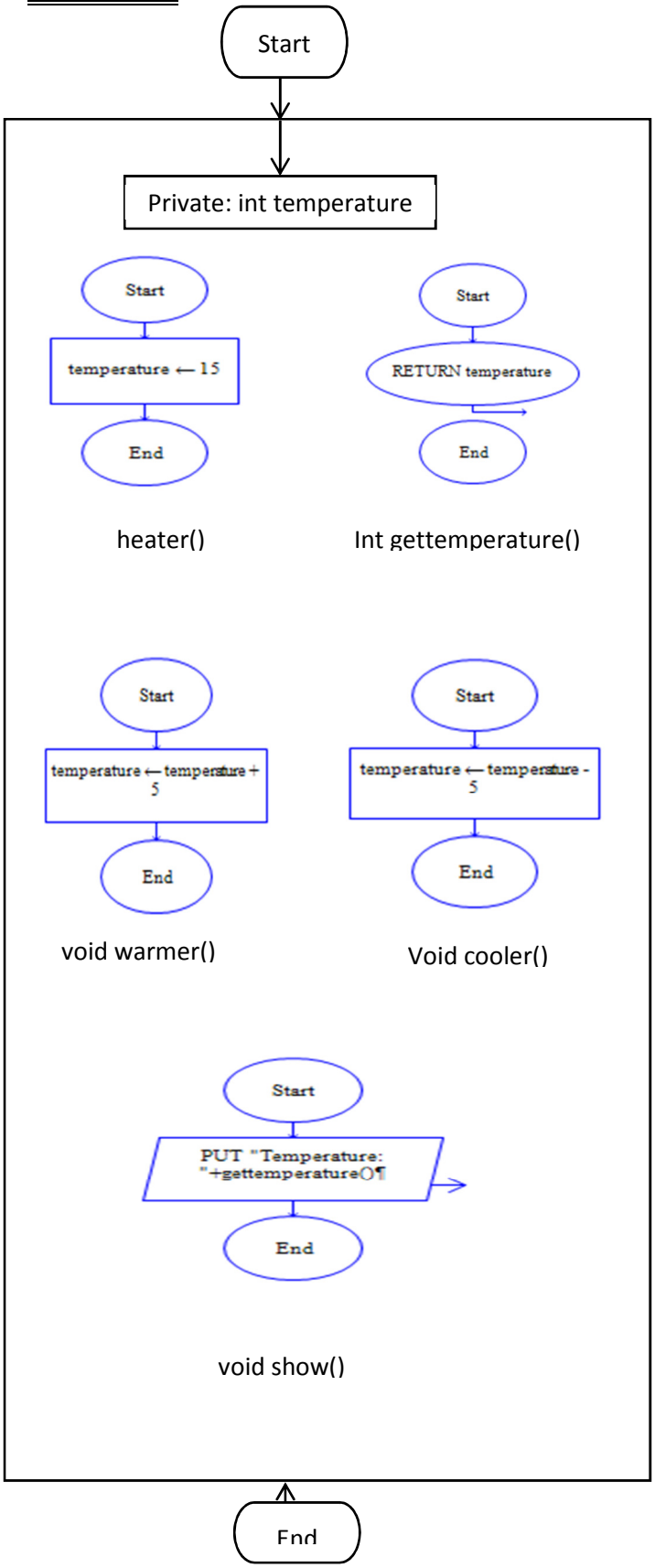
Algorithm:

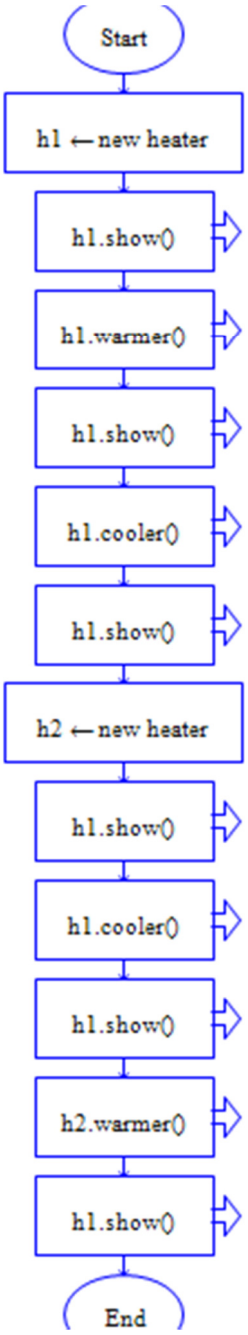
UML diagram for the above problem is given below:



- First make class heater
- Declare temperature as private integer field
- Define no argument constructor to set value of temperature to 15
- Define gettemperature method to return value of temperature
- Define warmer and cooler method to increase and decrease temperature by 5 respectively
- Define show function to display the output
- In main function, make objects of heater to demonstrate the use of heater
- Call each function one after the other and display the show function as shown in the flow chart.

Flowchart:

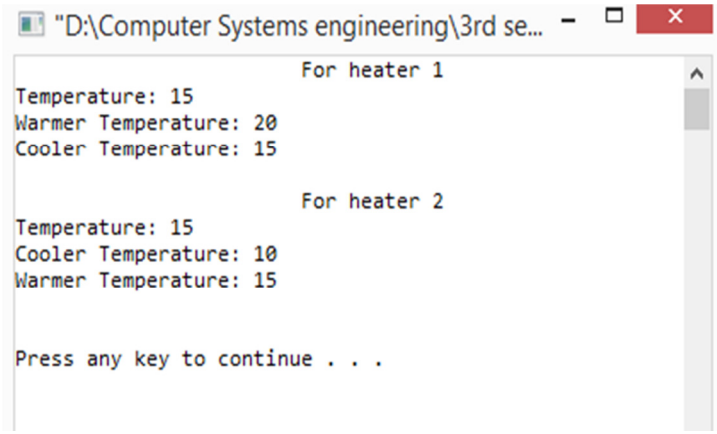

Start
Private: int temperature
Start
temperature ← 15
End
heater()
Start
RETURN temperature
End
Int gettemperature()
Start
temperature ← temperature + 5
End
void warmer()
Start
temperature ← temperature - 5
End
Void cooler()
Start
PUT "Temperature: "+gettemperature()¶
End
void show()
End


Start
h1 ← new heater
h1.show()
h1.warmer()
h1.show()
h1.cooler()
h1.show()
h2 ← new heater
h1.show()
h1.cooler()
h1.show()
h2.warmer()
h1.show()
End

In C++

Source code:

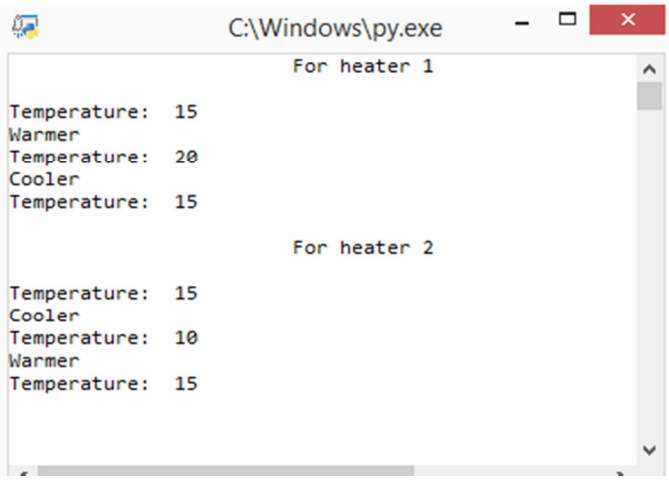
Output:



In Python

Source code:

Output:



In Java (Optional)

Source code:

Output:

```
<terminated> Activity_1_main [Java Application] C:\Program Files\Java\
                            For heater 1

Temperature: 15
Warmer Temperature: 20
Cooler Temperature: 15
                            For heater 2

Temperature: 15
Cooler Temperature: 10
Warmer Temperature: 15
```

Conclusion:

This program helps us in understanding the basic concepts of classes and objects in different languages. It acts as a base for us and helps us in preparing ourselves for the higher level of programming. We get to know about the constructor and method in OOP with the help of this program.

Registration #: _____ Name & Section: _____ Date: _____

CSE 208L – OBJECT ORIENTED PROGRAMMING LAB

LAB XX ASSESSMENT RUBRICS

Dimension	Exemplary	Acceptable	Developing	Unsatisfactory	Student Score out of 10 Marks
	10	9-7	6-5	4-0	
Submission	Report is submitted on time and in correct format.	Report is submitted on time with slight incorrect format.	Report is submitted on time in incorrect format.	Report is not submitted.	
Overall Impression of Lab Report	Report is complete, well written, and organized appropriately with additional elements that enhance it. Task titles and output screenshots are included. Purpose for each concept, input requirements and output results is noted.	Report is complete, briefly written, and organized. Lacks additional elements. Task titles and output screenshots are included. Purpose for each concept, input requirements and output results is noted.	Report is mostly complete, loosely written, and fairly organized. Basic documentation including descriptions of all concepts. Specific purpose is noted for each concept. Task titles and output screenshots are included and good formatting.	Report is incomplete, sloppy, and/or disorganized. No documentation included. No task titles, no output screenshots, poor formatting.	
Ability to Code Required Class/Classes	Able to code required class, use objects effectively, and produces desired results.	Able to code required class, use objects effectively, and produces most results.	Able to code required class, somewhat use of objects, and some results are produced.	Unable to code required class or unable to use objects.	
Compilation, Execution, and Results	Program compiles with no errors and no warnings. Executes without errors, excellent user prompts, good use of symbols, and spacing in output. Thorough and organized testing has been completed and output from test cases is included.	Program compiles with no errors and some warnings. Executes without errors. User prompts are understandable, minimum use of symbols or spacing in output. Most of the testing has been completed.	Program compiles with no errors and lots of warnings. Executes without errors. User prompts are understandable, minimum use of symbols or spacing in output. Some testing has been completed.	Program fails to compile. Does not execute due to errors. User prompts are misleading or non-existent. No testing has been d.	

Marks: _____ /4 = _____

Teacher Remarks and Signature: _____