**LAB # 01**



**Fall 2022**

**CSE208L Object Oriented Programming Lab**

Submitted by:

Shahzad Bangash(21PWCSE1980)

Suleman Shah(21PWCSE1983)

Ali Asghar(21PWCSE2059)

Class Section: **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 13, 2022

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

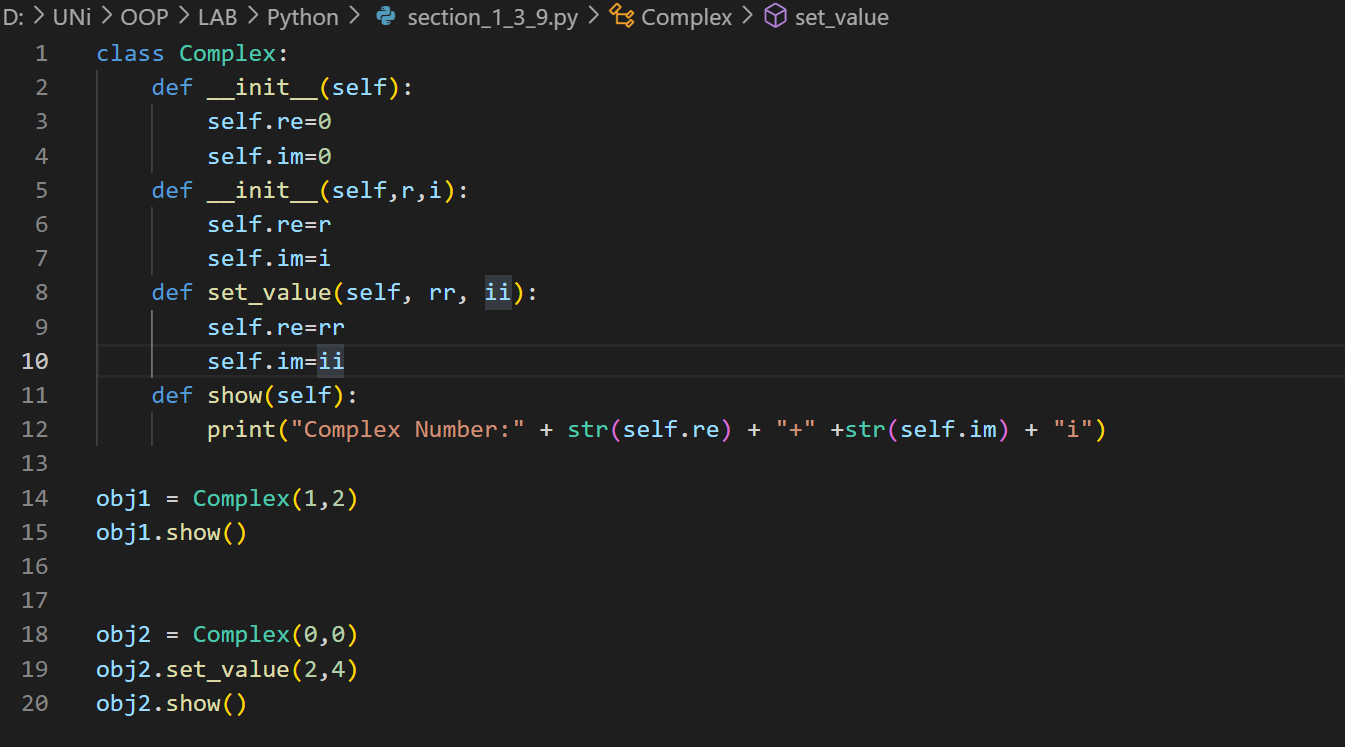
# Make Simple and Overloaded Constructor

# Use the Class Objects and Member Functions to provide and extract data from Object

# Practice with Classes and Objects

# Example # 01(Section 1.3.9)

**Code:**



**Output:**



**Algorithm:**

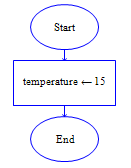
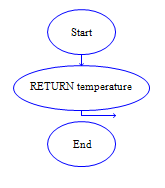
UML diagram for the above problem is given below:

* First create a class heater
* Declare temperature as private integer field
* Define no argument constructor to set value of temperature to 15
* Define access\_temperature 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 thenuse of heater
* Call each function one after the other and display the show function as shown in the flow chart.

**Flowchart:**

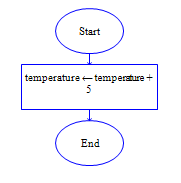
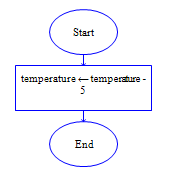
Start

Private: int temperature



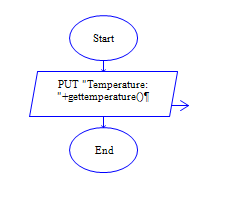
Int access\_temperature()

heater()



Void cooler()

void warmer()



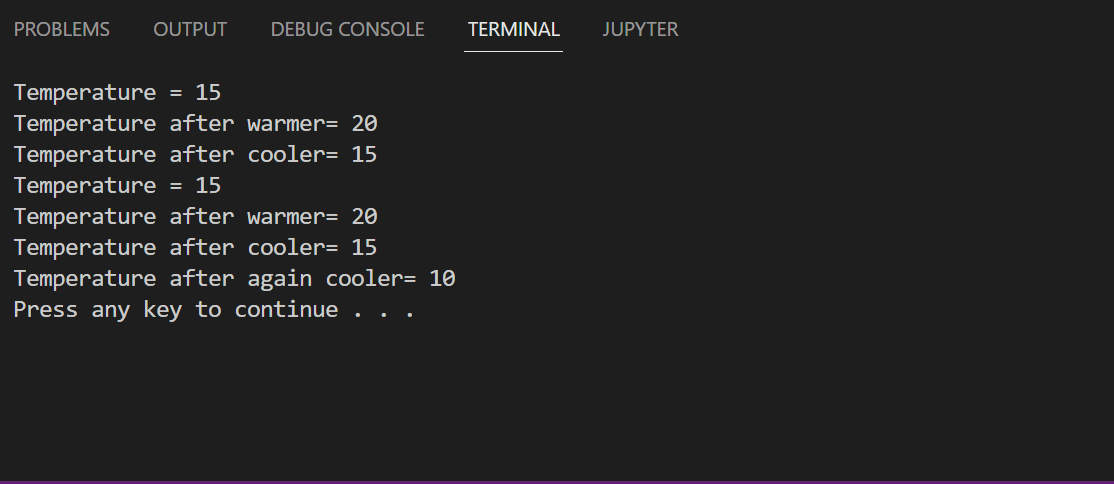
void show()

End

**In C++**

**Source code:**

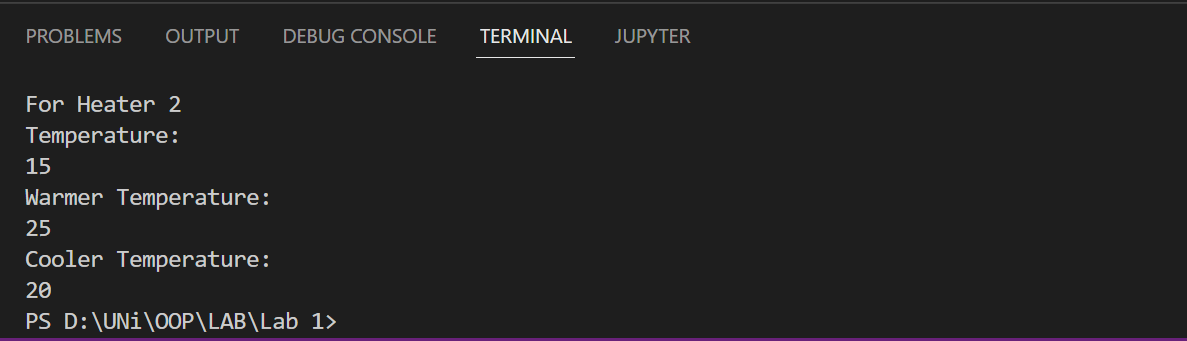
**Output:**

****

**In Python**

**Source code:**

**Output:**



**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.

# Activity # 02

**Title:**

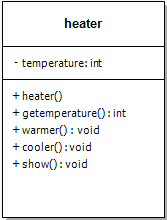
Make a Class for Point and model it using x and y-coordinates.

**Problem analysis:**

Create a class, **Point** that contains two integer fields, **x** and **y**. Define a constructor that takes no parameters. After that create another constructor with two parameters. Define the functions **Setx** and **Sety**, whose effect is to set the value of the x and y respectively and **Getx** and **Gety** returns the values of x and y. Define an accessor method to return the value of **Point**. Demonstrate the use of Point class.

**Algorithm:**

UML diagram for the above problem is given below:

* First make class Point

**Point**

* Declare x and y as private integer field
* Define a no argument constructor.

-x:int, y:int

* Define a constructor with two parameters.
* Define Setx and Sety method to set the values of x and y respectively.

+Point()

+Point(int a, int b)

+Setx(int a): void

+Sety(int a): void

+Getx(): int

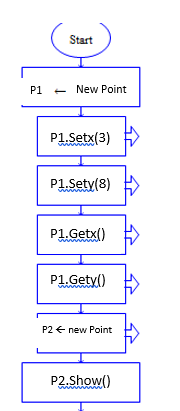
+Gety(): int

+Show(): void

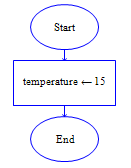
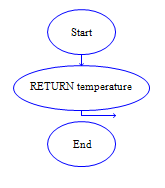
* Define Getx and Gety method to return the values of x and y respectively.
* Define show function to display the output
* In main function, make objects of Point to demonstrate theeuse of Point.
* Call each function one after the other and display the show function as shown in the flow chart.

**Flowchart:**

Start



Private: int xCordinate,yCordinate

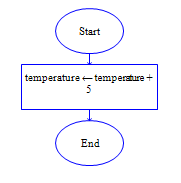
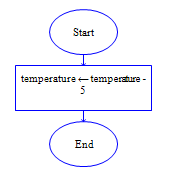


X=x, y=b

X=?, y=?

Point(int x,int y)

Point()



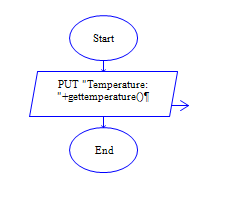
X=x

Y=y

End

Void setY(int y)

void setX(int x)



Print “(x,y)”

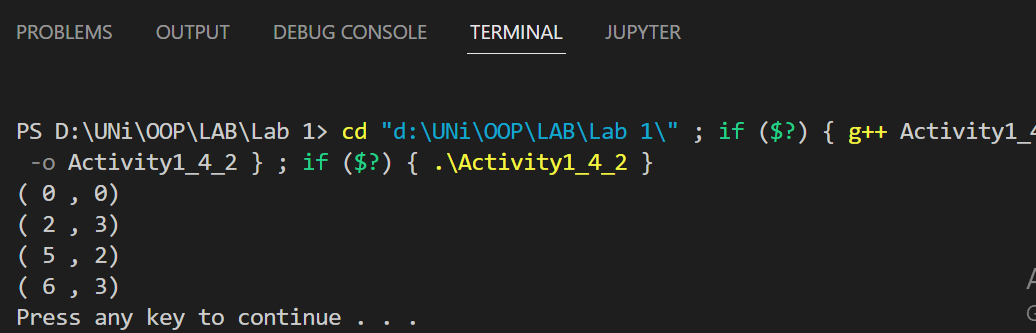
void display()

End

**In C++**

**Source code:**

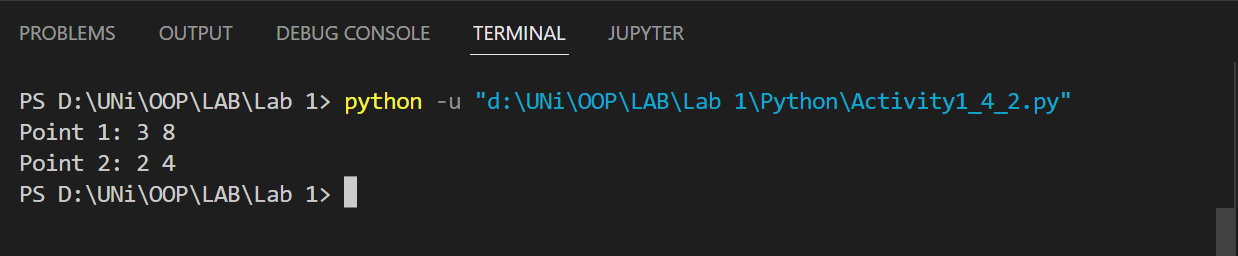
**Output:**

****

**In Python**

**Source code:**

**Output:**

****

**Conclusion:**

This program helps us in understanding the basic concepts of classes and objects in different languages. It also helps us understand **Constructor Overloading**.

# Activity # 03

**Title:**

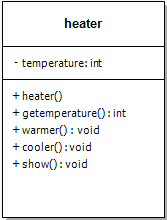
Make a Class for BankAccount and model it using **balance**.

**Problem analysis:**

Create a class, **BankAccount** that contains a single integer field, **balance**. Define a constructor that takes no parameters and sets balance to 1000. Define the function **deposit**, whose effect is to add some value to **balance** and **withdraw** which deduces some value from **balance**. Define an accessor method to show the value of **balance**. Demonstrate the use of BankAccount class.

**Algorithm:**

UML diagram for the above problem is given below:

* First make class BankAccount

**BankAccount**

* Declare balance as private integer field
* Define a no argument constructor.

-balance:int

* Define deposit method to add some value to balance.
* Define withdraw method to deduce some value from balance.

+BankAccount()

+deposit(int a): void

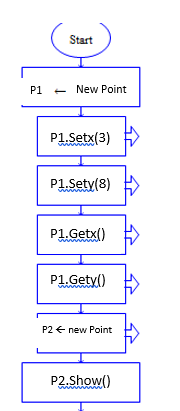
+withdraw(int a): void

+CurrentBalance(): void

* Define CurrentBalance function to display the balance
* In main function, make object of BankAccount to demonstrate theeuse of BankAccount.
* Call each function one after the other and display the CurrentBalance function as shown in the flow chart.

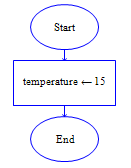
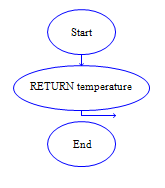
**Flowchart:**

Start



Private: int balance

Acc 🡨 new BankAccount



Acc.CurrentBalance

Balance+=a

Balance=1000

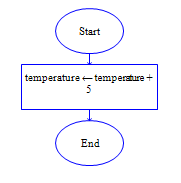
acc.deposit(500)

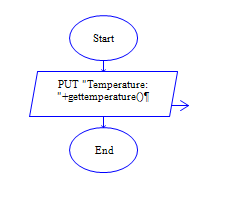
Acc.CurrentBalance

BankAccount()

Void deposit(int a)

Acc.withdraw(1000)





acc.deposit(500)

Acc.CurrentBalance

Print balance

Balance-=a

End

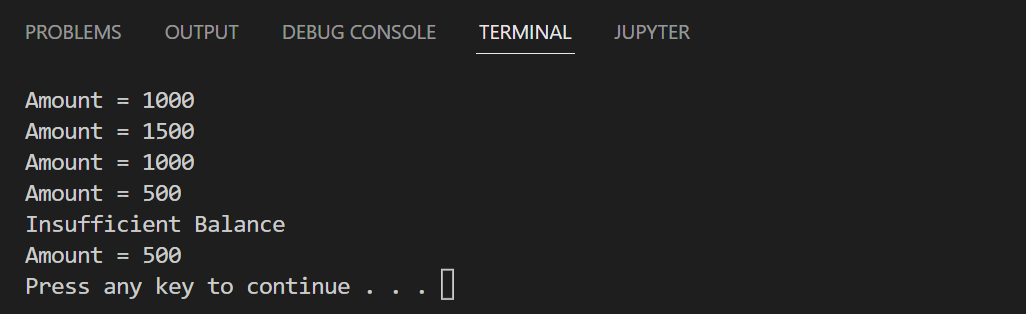
void withdraw(int a)

void CurrentBalance()

End

**In C++**

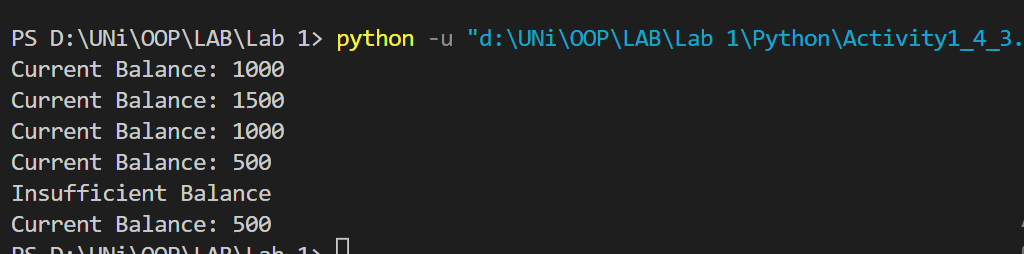
**Source code:**

**Output:** ****

**In Python**

**Source code:**

**Output:**



**Conclusion:**

This program helps us in understanding the basic concepts of classes and objects in different languages. It also helps us understand **Creating Functions in Class**.