**Task 1: Create a class Point having X and Y axis then perform an operator overloading**

**(Overload all relation operator).**

In [41]:



1

**class** Point:

2

**def** \_\_init\_\_(self, x,y):

3

self.x **=** x

4

self.y**=**y

5

**def** \_\_lt\_\_(self, other):

6

**if** self.x **>** other.x:

7

**return** **True**

8

**if** self.y **>** other.y:

9

**return** **True**

10

**else**:

11

**return** **False**

12

**else**:

13

**return** **False**

14

​

15

**def** \_\_gt\_\_(self, other):

16

**if** self.x **>** other.x:

17

**return** **True**

18

**if** self.y **>** other.y:

19

**return** **True**

20

**else**:

21

**return** **False**

22

**else**:

23

**return** **False**

24

**def** \_\_ge\_\_(self, other):

25

**if** self.x **>=** other.x:

26

**return** **True**

27

**if** self.y **>** other.y:

28

**return** **True**

29

**else**:

30

**return** **False**

31

**else**:

32

**return** **False**

33

**def** \_\_le\_\_(self, other):

34

**if** self.x **<=** other.x:

35

**return** **True**

36

**if** self.y **>** other.y:

37

**return** **True**

38

**else**:

39

**return** **False**

40

**else**:

41

**return** **False**

42

**def** \_\_eq\_\_(self, other):

43

**if** self.x **==** other.x:

44

**return** **True**

45

**if** self.y **>** other.y:

46

**return** **True**

47

**else**:

48

**return** **False**

49

**else**:

50

**return** **False**

51

**def** \_\_ne\_\_(self, other):

52

**if** self.x **!=** other.x:

53

**return** **True**

54

**if** self.y **>** other.y:

55

**return** **True**

56

**else**:

57

**return** **False**

58

**else**:

59

**return** **False**

60

c1 **=** Point(90,40)

61

c2 **=** Point(56,9)

62

print("Is c1 greater than c2: ",c1 **>** c2)

63

print("Is c1 greater or equal to c2: ",c1 **>=** c2)

64

print("Is c1 less than or equal to c2: ",c1 **<=** c2)

65

print("Is c1 less than c2: ",c1 **<** c2)

66

print("Is c1 equal to c2: ",c1 **==** c2)

67

print("Is c1 not equal to c2: ",c1 **!=** c2)

Is c1 greater than c2: True

Is c1 greater or equal to c2: True

Is c1 less than or equal to c2: False

Is c1 less than c2: True

Is c1 equal to c2: False

Is c1 not equal to c2: True

**Task 2:**

**Find out one real world example of interface and implement all abstract method by using python code.**

In [44]:



1

**from** abc **import** ABC, abstractmethod

2

**class** Payment(ABC):

3

**def** print\_slip(self, amount):

4

print('Purchase of amount ', amount)

5

@abstractmethod

6

**def** payment(self, amount):

7

**pass**

8

​

9

**class** CreditCardPayment(Payment):

10

**def** payment(self, amount):

11

print('Credit card payment of ', amount)

12

​

13

**class** MobileWalletPayment(Payment):

14

**def** payment(self, amount):

15

print('Mobile wallet payment of ', amount)

16

​

17

obj **=** CreditCardPayment()

18

obj.payment(100)

19

obj.print\_slip(100)

20

obj **=** MobileWalletPayment()

21

obj.payment(200)

22

obj.print\_slip(200)

Credit card payment of 100

Purchase of amount 100

Mobile wallet payment of 200

Purchase of amount 200