Aim: Write python programs to implement **operator overloading.**

Write a python program to overload *, +, < and > operators to make it act on objects. Create class Employee with attributes name and daily salary. Create class Attendance with attributes name and number of working days. Total salary of the employee = daily salary * no of days worked.

Hint:

1. Multiply the Employee object data with Attendance object data.
2. For two different employees, compare the daily salary of first employee with second and return True or False accordingly.
3. Add multiple employee object data and find sum of salaries.

Objective of the Experiment:

1. Understanding **operator overloading** as example of polymorphism in python.

Source code for the implementation:

(Write only important functions)

Post Labs:

```
class Spell:
    def __init__(self, incantation, name):
        self.name = name
        self.incantation = incantation
    def __str__(self):
        return self.name + ' ' + self.incantation + '\n' + self.get_description()

    def get_description(self):
        return 'No description'

    def execute(self):
        print(self.incantation)

class Accio(Spell):
    def __init__(self):
        Spell.__init__(self, 'Accio', 'Summoning Charm')
class Confundo(Spell):
    def __init__(self):
        Spell.__init__(self, 'Confundo', 'Confundus Charm')
    def get_description(self):
        return 'Causes the victim to become confused and befuddled.'
def study_spell(spell):
      print(spell)
spell = Accio()
spell.execute()
study_spell(spell)
study_spell(Confundo())
```

1. What are the parent and child classes here?
2. What does the code print out? (Try figuring it out without running it in Python)
3. Which get description method is called when ‘study spell(Confundo())’ is executed? Why?
4. What do we need to do so that ‘print Accio()’ will print the appropriate description (‘This charm summons an object to the caster, potentially over a significant distance’)? Write down the code that we need to add and/or change.