Week3 Set1

ANUKRITI SINGH RA1911003010270

Q1

Create a Bus child class that inherits from the Vehicle class. The default fare charge of any vehicle is seating capacity * 100. If Vehicle is Bus instance, we need to add an extra 10% on full fare as a maintenance charge. So total fare for bus instance will become the **final amount = total fare + 10% of the total fare**.

Note: The bus seating capacity is 50. so the final fare amount should be 5500. You need to override the fare() method of a Vehicle class in Bus class.

Use the following code for your parent Vehicle class. We need to access the parent class from inside a method of a child class.

```
class Vehicle:
     def __init__(self, name, mileage, capacity):
         self.name = name
         self.mileage = mileage
         self.capacity = capacity
     def fare(self):
         return self.capacity * 100
class Bus(Vehicle):
     pass
School_bus = Bus("School Volvo", 12, 50)
print("Total Bus fare is:", School_bus.fare())
```

Expected Output: Total Bus fare is: 5500.0

In [16]:
```
class Vehicle:
    def __init__(self, name, mileage, capacity):
        self.name = name
        self.mileage = mileage
        self.capacity = capacity

    def fare(self):
        totalFare = self.capacity*100
        return totalFare*1.1


class Bus(Vehicle):
    pass


School_bus = Bus("School Volvo", 12, 50)
print('Total Bus fase is: {:.1f}'.format(School_bus.fare()))
```

```
Total Bus fase is: 5500.0
```

Q2

Consider the Book definition given in Example

Here are some questions to test your understanding of what it does:

1. How would you print out the author attribute of the pynut instance (at the interpreter, after running the file)?

   ```
   print('The name of the author is ' + pynut.authorfirst + ' ' + pynut.authorlast)
   ```

2. If you type print beauty.write bib entry() at the interpreter (after running the file), what will happen?

   ```
   print(beauty.write_bib_entry())
   RESULT:
   Dubay, Thoumas, The Evidential Power of Beauty, San Francisco, Ignatius Press, 1999
   ```

3. How would you change the publication year for the beauty book to "2010"?

   ```
   beauty.year = 2010
   ```

Q3

Define a class named CaseString with the following methods:

- the constructor takes an initial string as a parameter
- set_upper: sets string mode (False = original case, True = upper case)
- set_string: sets the encapsulated string
- get_string: returns the string with the proper case

In [11]:
```
class CaseString:
    def __init__(self, string):
        self.string = string
    def set_upper(self):
        if self.string.isupper():
            return True
        else:
            return False
    def set_string(self, newString):
        self.string = newString
        print('New string is updated successfully with the following value: ')
        print(self.string)
    def get_string(self):
        return self.string.title()

string = 'THIS IS A STRING'
s1 = CaseString(string)

s1.set_string('This is a new String')

print('\nWriting the string in proper case')
print(s1.get_string())
```

```
New string is updated successfully with the following value:
This is a new String

Writing the string in proper case
This Is A New String
```

Q4

Define a class named Circle which can be constructed by a radius. The Circle class has two methods for computing perimeter and area, respectively.

In [12]:
```
class Circle:
    def __init__(self, r):
        self.radius = r
    def perimeter(self):
        return 'Perimeter is: ' + str(2*3.14*self.radius)
    def area(self):
        return 'Area is: ' + str(3.14*(self.radius**2))

radius = int(input('Enter the length of the radius: '))
c1 = Circle(radius)
print(c1.perimeter())
print(c1.area())
```

```
Enter the length of the radius: 10
Perimeter is: 62.800000000000004
Area is: 314.0
10
```

Q5

Define a class named Shape and its subclass Square. The Square class has a constructor which takes a length as argument. Both classes have an area function which can print the area of the shape where Shape's area is 0 by default.

In [ ]:
```
class Shape:
    def area(self):
        self.areaVar = 0
        print('The area of the shape is: ' + str(self.areaVar))

class Square(Shape):
    def __init__(self, length):
        self.length = length
    def area(self):
        self.areaVar = self.length**2
        print('The area of the Square is: ' + str(self.areaVar))
```