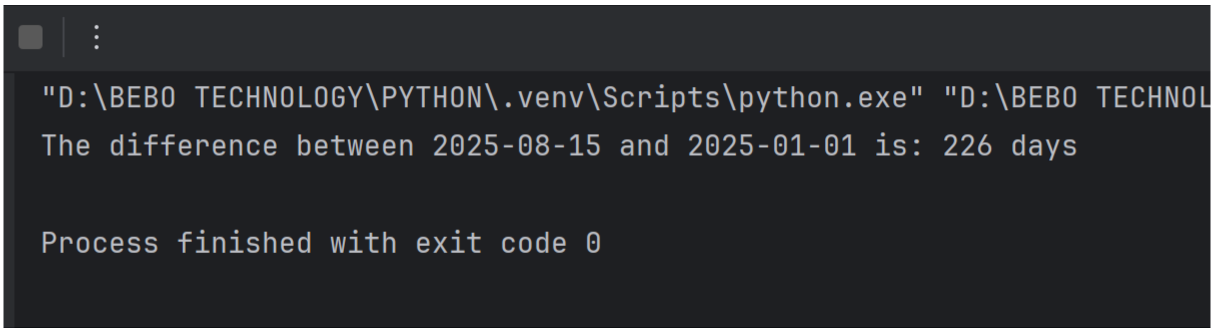
Assignment-7

1. Calculate the difference in days between two dates: 15th August 2025 and 1st January 2025.

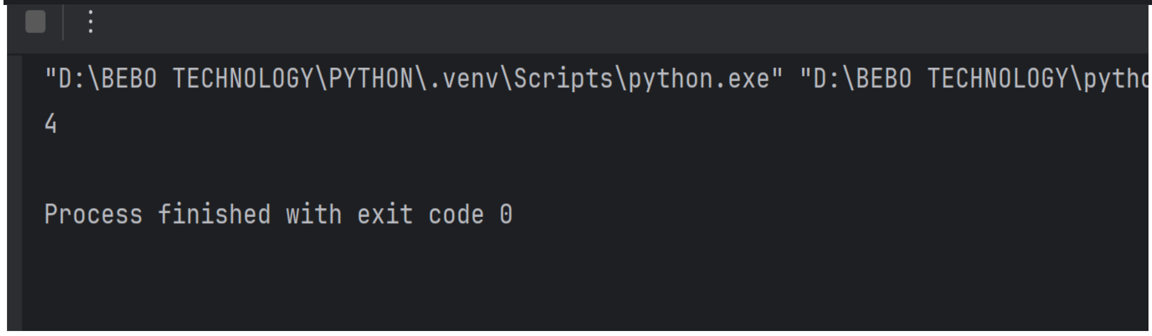
```
import datetime

date_1 = datetime.date(2025,8,15)
date_2 = datetime.date(2025,1,1)
diff = date_2-date_1
print(f"The difference between {date_1} and {date_2} is:
{abs(diff.days)} days")
```



```
"D:\BEBO TECHNOLOGY\PYTHON\.venv\Scripts\python.exe" "D:\BEBO TECHNOL
The difference between 2025-08-15 and 2025-01-01 is: 226 days

Process finished with exit code 0
```

2. Write a Counter class with a class variable count to keep track of how many objects have been created. Test this by creating multiple objects and printing the count.

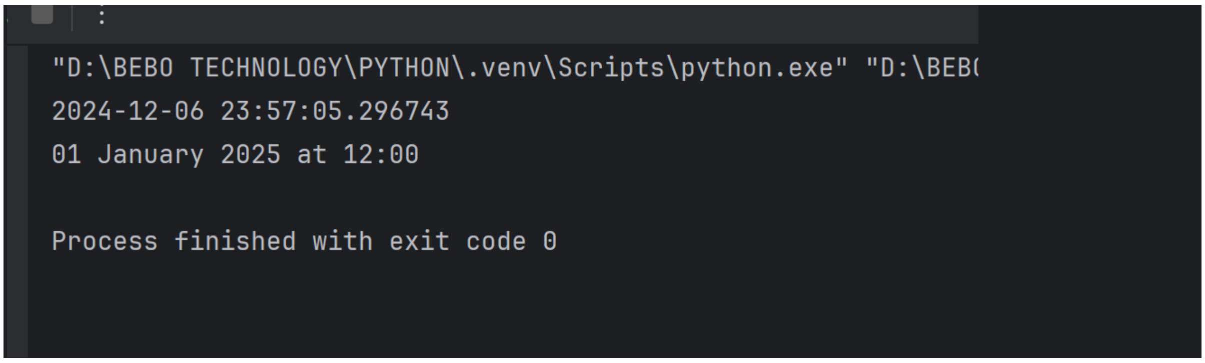
```
class Counter:
    count = 0
    def __init__(self):
         Counter.count+=1
```



```
"D:\BEBO TECHNOLOGY\PYTHON\.venv\Scripts\python.exe" "D:\BEBO TECHNOLOGY\pytho
4

Process finished with exit code 0
```

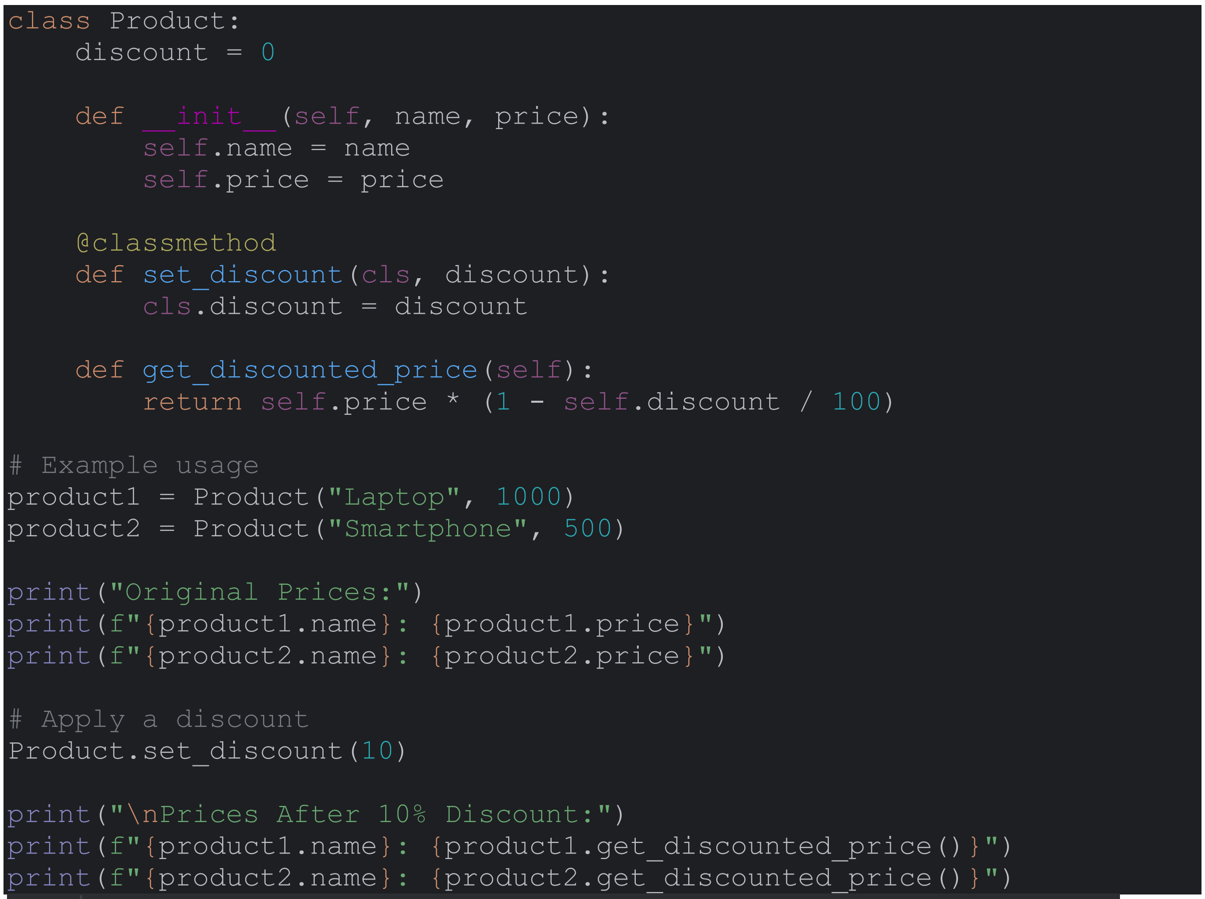
3. Write a program to print the current date and time in the format YYYY-MM-DD HH:MM:SS. Create a datetime object for 1st January 2025, 12:00 PM and print it in the format Day Month, Year at HH:MM.

```
import datetime
now = datetime.datetime.now()
print(now)

date = datetime.datetime(2025,1,1,12,0)
# print(f"{date.day}-{date.month}-{date.year} at {time.hour}:{time.mi-
nute}")
print(date.strftime("%d %B %Y at %H:%M"))
```

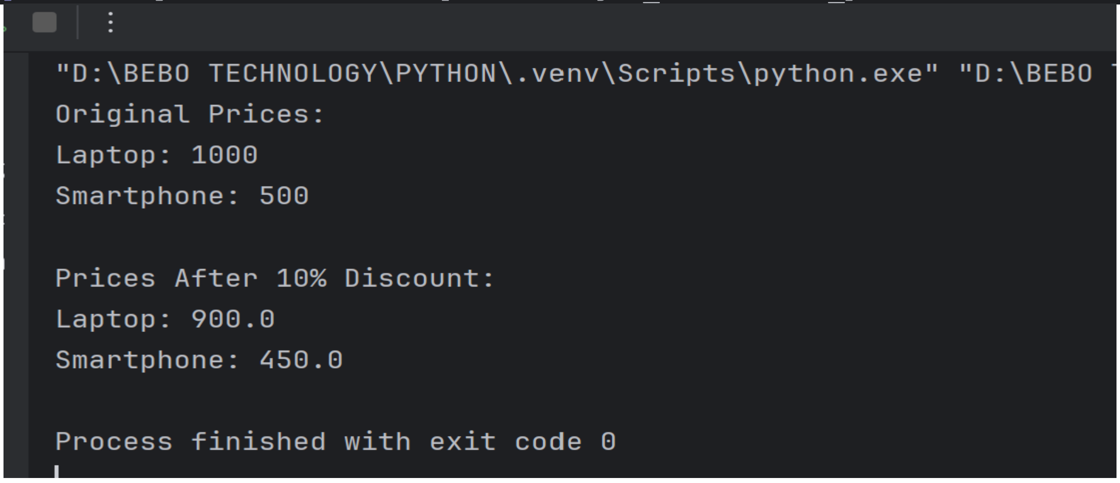


```
"D:\BEBO TECHNOLOGY\PYTHON\.venv\Scripts\python.exe" "D:\BEB
2024-12-06 23:57:05.296743
01 January 2025 at 12:00

Process finished with exit code 0
```

4. Write a Product class with instance attributes for name and price. Add a class method set_discount(cls, discount) to apply a discount to all products. Use this class method to change the price of all created products.



```python
class Product:
    discount = 0

    def __init__(self, name, price):
        self.name = name
        self.price = price

    @classmethod
    def set_discount(cls, discount):
        cls.discount = discount

    def get_discounted_price(self):
        return self.price * (1 - self.discount / 100)

# Example usage
product1 = Product("Laptop", 1000)
product2 = Product("Smartphone", 500)

print("Original Prices:")
print(f"{product1.name}: {product1.price}")
print(f"{product2.name}: {product2.price}")

# Apply a discount
Product.set_discount(10)

print("\nPrices After 10% Discount:")
print(f"{product1.name}: {product1.get_discounted_price()}")
print(f"{product2.name}: {product2.get_discounted_price()}")
```

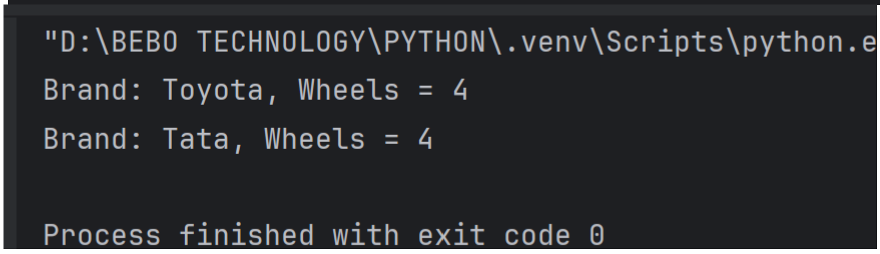


```
"D:\BEBO TECHNOLOGY\PYTHON\.venv\Scripts\python.exe" "D:\BEBO
Original Prices:
Laptop: 1000
Smartphone: 500

Prices After 10% Discount:
Laptop: 900.0
Smartphone: 450.0

Process finished with exit code 0
```

5. Create a class Car with:

- An instance variable brand
- A class variable wheels initialized to 4
- Add a method show() to print both variables.

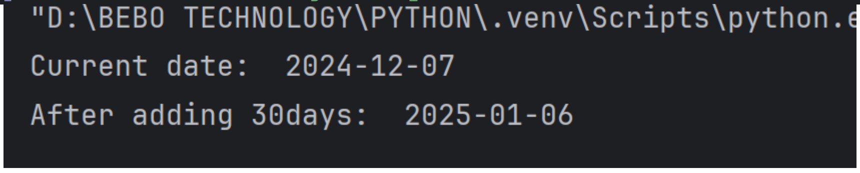
```python
class Car:
    wheels = 4

    def __init__(self,brand):
        self.brand = brand

    def show(self):
        print(f"Brand: {self.brand}, Wheels = {self.wheels}")

car1 = Car("Toyota")
car1.show()
car2 = Car("Tata")
car2.show()
```



```
"D:\BEBO TECHNOLOGY\PYTHON\.venv\Scripts\python.e
Brand: Toyota, Wheels = 4
Brand: Tata, Wheels = 4

Process finished with exit code 0
```

6. Write a program to add 30 days to the current date and print the result.

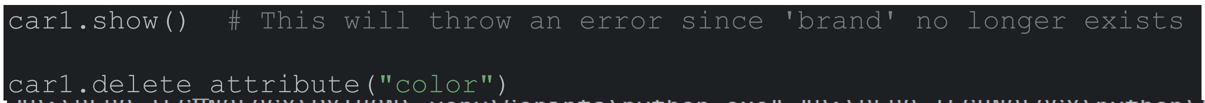
```python
import datetime

now = datetime.datetime.now()
add = now+datetime.timedelta(days=30)
print("After adding 30days: ",add.date())
```



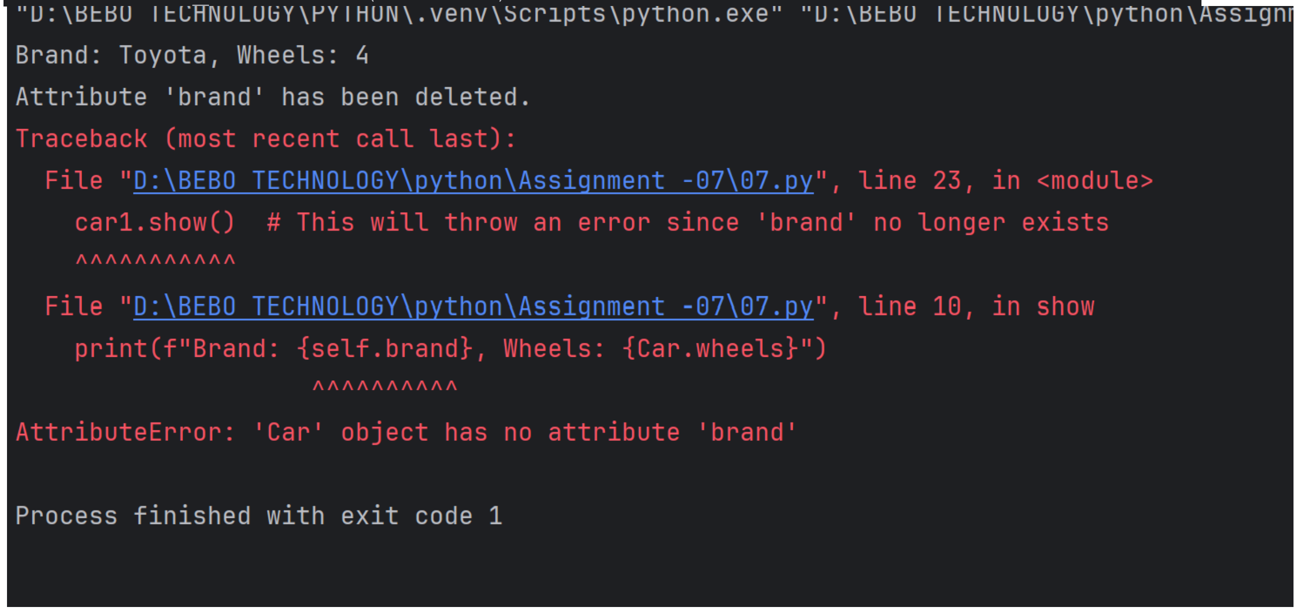
```
"D:\BEBO TECHNOLOGY\PYTHON\.venv\Scripts\python.e
Current date:  2024-12-07
After adding 30days:  2025-01-06
```

7. Extend the Car class to include a method delete_attribute(attr_name) that checks if the attribute exists before deleting it. Print an appropriate message if the attribute does not exist.

```python
class Car:
    wheels = 4
    def __init__(self, brand):
        self.brand = brand

    def show(self):
        print(f"Brand: {self.brand}, Wheels: {Car.wheels}")

    def delete_attribute(self, attr_name):
        if hasattr(self, attr_name):
            delattr(self, attr_name)
            print(f"Attribute '{attr_name}' has been deleted.")
        else:
            print(f"Attribute '{attr_name}' does not exist.")

car1 = Car("Toyota")
car1.show()

car1.delete_attribute("brand")
```

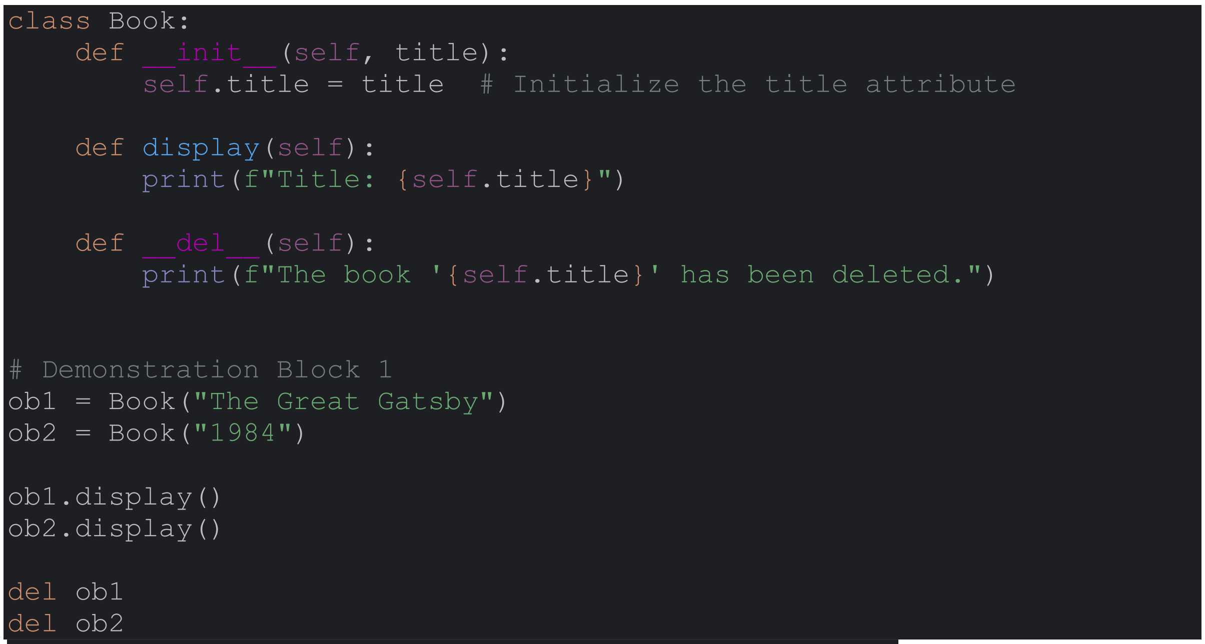


```
car1.show()  # This will throw an error since 'brand' no longer exists

car1.delete_attribute("color")
```

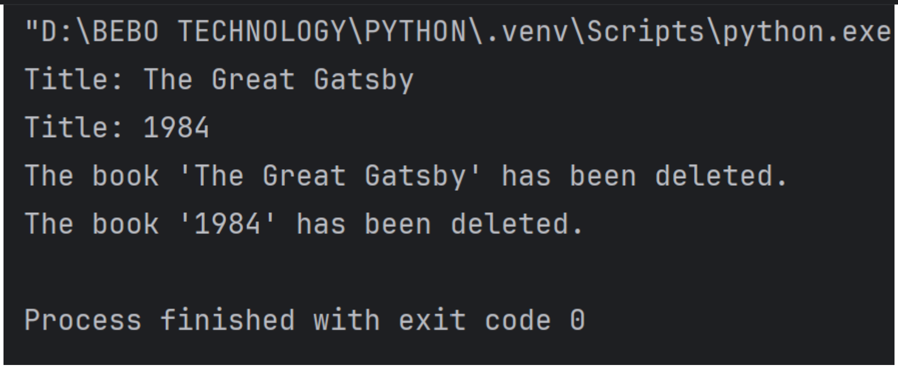


```
"D:\BEBO TECHNOLOGY\PYTHON\.venv\Scripts\python.exe" "D:\BEBO TECHNOLOGY\python\Assignm
Brand: Toyota, Wheels: 4
Attribute 'brand' has been deleted.
Traceback (most recent call last):
  File "D:\BEBO TECHNOLOGY\python\Assignment -07\07.py", line 23, in <module>
    car1.show()  # This will throw an error since 'brand' no longer exists
    ^^^^^^^^^^^
  File "D:\BEBO TECHNOLOGY\python\Assignment -07\07.py", line 10, in show
    print(f"Brand: {self.brand}, Wheels: {Car.wheels}")
                    ^^^^^^^^^^
AttributeError: 'Car' object has no attribute 'brand'

Process finished with exit code 1
```

8. Create a class Book with a constructor that initializes the title. Override the del method to print a message when the object is deleted. Create and delete a Book object to demonstrate this.



```
class Book:
    def __init__(self, title):
        self.title = title  # Initialize the title attribute

    def display(self):
        print(f"Title: {self.title}")

    def __del__(self):
        print(f"The book '{self.title}' has been deleted.")


# Demonstration Block 1
ob1 = Book("The Great Gatsby")
ob2 = Book("1984")

ob1.display()
ob2.display()

del ob1
del ob2
```



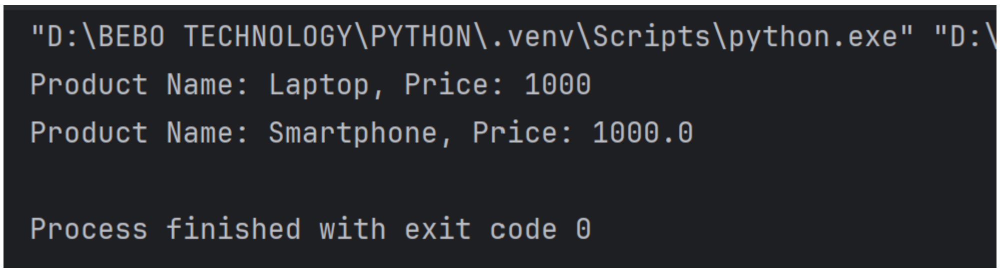
```
"D:\BEBO TECHNOLOGY\PYTHON\.venv\Scripts\python.exe
Title: The Great Gatsby
Title: 1984
The book 'The Great Gatsby' has been deleted.
The book '1984' has been deleted.

Process finished with exit code 0
```

9. Create a class Product with:

- A constructor that takes name and price.

- A class method from_discounted_price(name, discounted_price, discount_percentage) that initializes a product based on the discounted price.
- Demonstrate the use of both constructors.

```
class Product:
    def __init__(self, name, price):
        self.name = name
        self.price = price

    @classmethod
    def from_discounted_price(cls, name, discounted_price, discount_per-
centage):
        original_price = discounted_price / (1 - discount_percentage /
100)
        return cls(name, original_price)

    def display(self):
        print(f"Product Name: {self.name}, Price: {self.price}")

product1 = Product("Laptop", 1000)
product1.display()

product2 = Product.from_discounted_price("Smartphone", 800, 20)
product2.display()
```



```
"D:\BEBO TECHNOLOGY\PYTHON\.venv\Scripts\python.exe" "D:\
Product Name: Laptop, Price: 1000
Product Name: Smartphone, Price: 1000.0

Process finished with exit code 0
```