Assignment 2 Solutions

Muhammad Turab

18CS45

Muhammad Turab

Task-01 –

```python
# Task-01 - Create a class method change_total and change value of total_phones to 5.
#Output:
#0
#5

# Author : Muhammad Turab

class Mobile:

    total_phones = 0

    @classmethod
    def change_phones(cls, phones):
        cls.total_phones = phones

    def __init__(self, brand, ram, rom):
        self.brand = brand
        self.ram = ram
        self.rom = rom
        Mobile.total_phones += 1


print(Mobile.total_phones)
Mobile.change_phones(5)
print(Mobile.total_phones)
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Turab Bajeer\Desktop\Python bootcamp\Day 5>"
0
5

C:\Users\Turab Bajeer\Desktop\Python bootcamp\Day 5>
```

Task-02 -

task2.py > ...

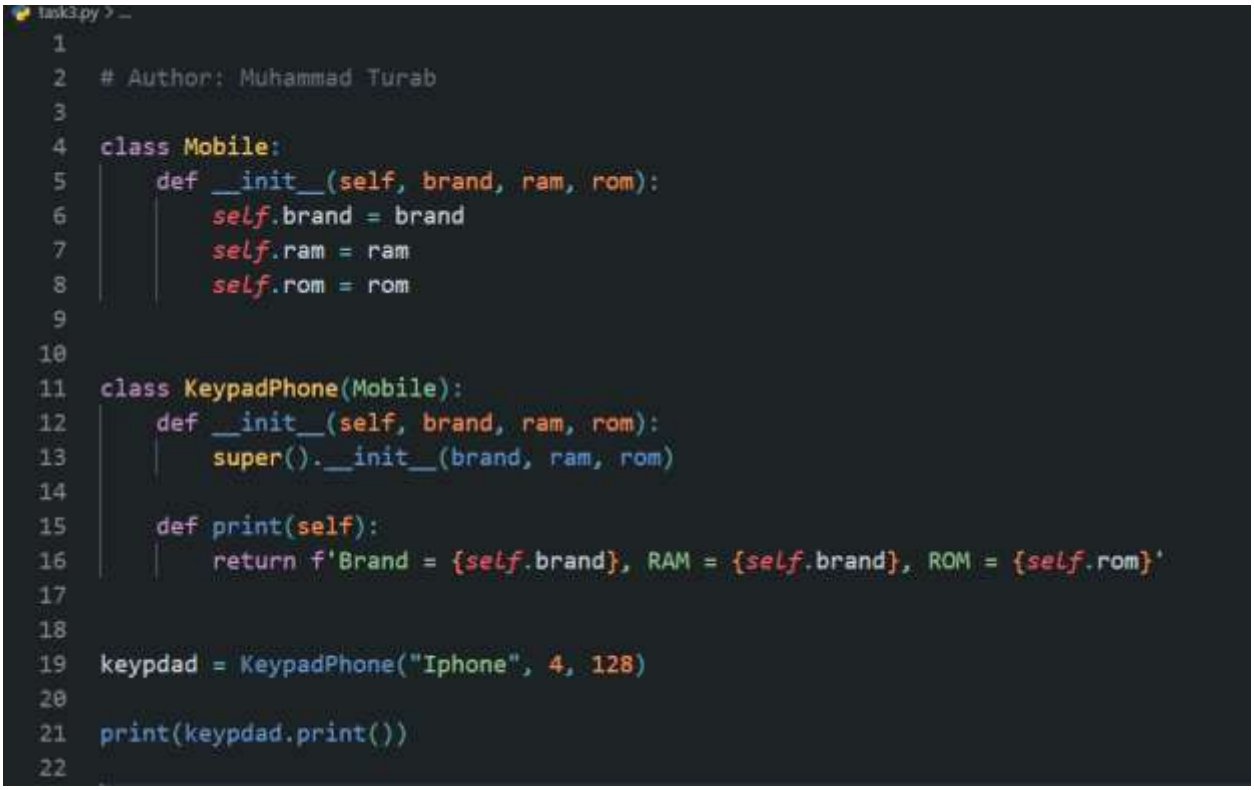
```python
# Author: Muhammad Turab
import random


class Mobile:
    def __init__(self, brand, ram, rom):
        self.brand = brand
        self.ram = ram
        self.rom = rom

    @classmethod
    def lucky_draw(cls, list):
        num = random.randint(0, len(list)-1)
        return list[num]


list = ['Nokia', 'Iphone', 'Samsung', 'Blackberry']
print(Mobile.lucky_draw(list))
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Turab Bajeer\Desktop\Python bootcamp\Day 5>"
Nokia

C:\Users\Turab Bajeer\Desktop\Python bootcamp\Day 5>"
Blackberry

C:\Users\Turab Bajeer\Desktop\Python bootcamp\Day 5>"
Iphone

C:\Users\Turab Bajeer\Desktop\Python bootcamp\Day 5>
```

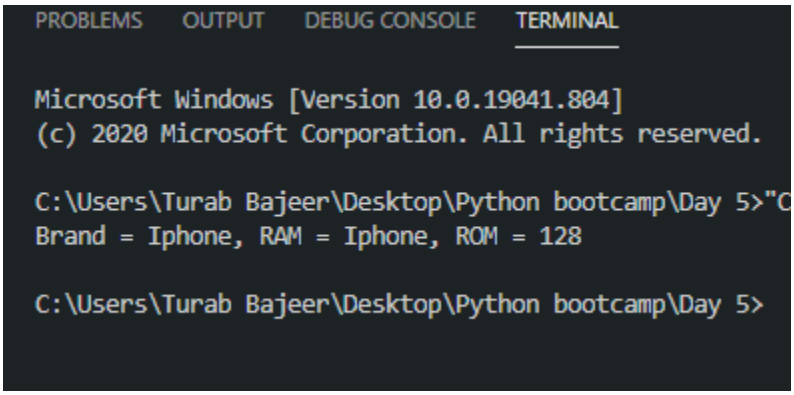
Task-03

Code:



```python
# Author: Muhammad Turab

class Mobile:
    def __init__(self, brand, ram, rom):
        self.brand = brand
        self.ram = ram
        self.rom = rom


class KeypadPhone(Mobile):
    def __init__(self, brand, ram, rom):
        super().__init__(brand, ram, rom)

    def print(self):
        return f'Brand = {self.brand}, RAM = {self.brand}, ROM = {self.rom}'


keypdad = KeypadPhone("Iphone", 4, 128)

print(keypdad.print())
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Turab Bajeer\Desktop\Python bootcamp\Day 5>"C
Brand = Iphone, RAM = Iphone, ROM = 128

C:\Users\Turab Bajeer\Desktop\Python bootcamp\Day 5>
```

Task 4-

```python

# Author: Muhammad Turab

class Mobile:
    def __init__(self, brand, ram, rom):
        self.brand = brand
        self.ram = ram
        self.rom = rom


class KeypadPhone(Mobile):
    def __init__(self, brand, ram, rom, touch=False):
        super().__init__(brand, ram, rom)
        self.touch = touch

    def print(self):
        return f'Brand = {self.brand}, RAM = {self.brand}, ROM = {self.rom}, Touch = {self.touch}'


keypdad = KeypadPhone("Iphone", 4, 128)

print(keypdad.print())

```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Turab Bajeer\Desktop\Python bootcamp\Day 5>"C:/Users/Turab E
Brand = Iphone, RAM = Iphone, ROM = 128, Touch = False
```

Task 5: Define three types of methods and difference between them.

Ans:

Three types of methods static methods, class methods and instance methods.

1. Regular (instance) methods need a class instance and can access the instance through self. They can read and modify an objects state freely.

2. Class methods, marked with the @classmethod decorator, don't need a class instance. They can't access the instance (self) but they have access to the class itself via cls.

3. Static methods, marked with the @staticmethod decorator, don't have access to cls or self. They work like regular functions but belong to the class's namespace.

Task 6: Why inheritance is useful?

Ans:

It makes easier to create and maintain an application. Inheritance also provides an opportunity to reuse the code functionality and fast implementation time.