Exercise:13

Date:20/11/2020

AIM:

To write a Python program to implement the class diagram.

PROGRAM:

```
class Apparel:
  counter=100
  def __init__(self,price,item_type):
    apparel.counter+=1
    self.__item_id=item_type[0]+str(Apparel.counter)
    self.__price=price
    self.__item_type=item_type
  def calculate_price(self):
    self.__price+=self.__price*0.05
  def get_item_id(self):
    return self.__item_id
```
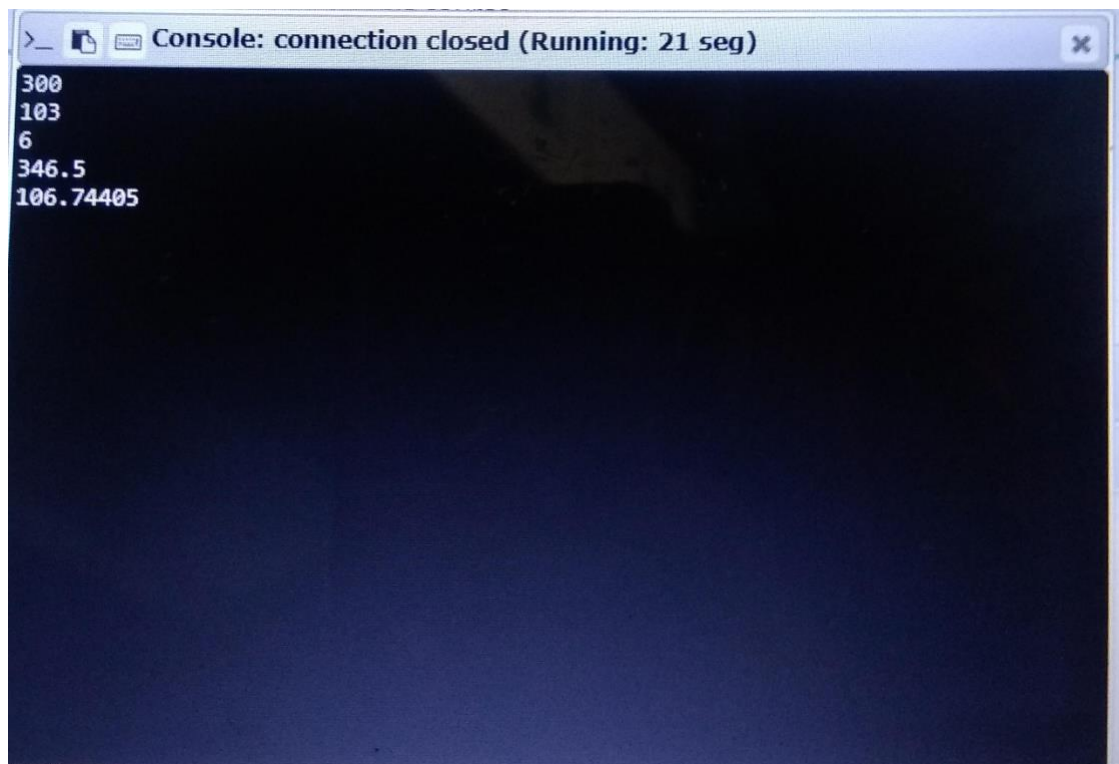
```python
    def get_price(self):
        return self.__price
    def get_item_type(self):
        return self.__item_type
def set_price(self,price):
    self.__price=price
    return self.__price
class Cotton(Apparel):
def __init__(self,price,discount):
    super().__init__(price,'Cotton')
    self.__discount=discount
    def calculate_price(self):
        super().calculate_price()
        price=self.get_price()
        price-=price*(self.__discount/100)
        price+=price*0.05
        self.set_price(price)
        return price
    def get_discount(self):
        return self.__discount
```

```python
class Silk(Apparel):
    def __init__(self,price):
        super().__init__(price,'Silk')
        self.__points=None
    def calculate_price(self):
        super().calculate_price()
        if(self.get_price()>10000):
            self.__points=10
        else:
            self.__points=3
        return self.set_price(self.get_price()+(self.get_price()*0.1))
    def get_points(self):
        return self.__points
silk=int(input())
cotton=int(input())
discount=int(input())
a=Silk(silk)
print(a.calculate_price())
b=Cotton(cotton,discount)
print(b.calculate_price())
```

MOODLE LINK:

OUTPUT:


```
>_  [] [] Console: connection closed (Running: 21 seg)          ×
300
103
6
346.5
106.74405
```

RESULT:

     The program to implement the class diagram is executed and the output is verified.