AIM:

Write a python program to implement the class diagram given below.

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

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

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())

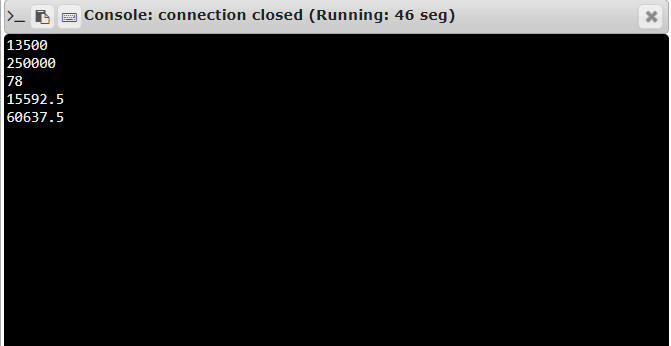
a=Silk(silk)

print(a.calculate\_price())

b=Cotton(cotton,discount)

print(b.calculate\_price())

OUTPUT:



LINK:

[http://103.53.53.18/mod/vpl/forms/edit.php?id=328&userid=1700#](http://103.53.53.18/mod/vpl/forms/edit.php?id=328&userid=1700)

RESULT:

Thus, the program was executed successfully.