House / Yard Cleaning Cost Calculator

This is a Python program that will help you figure out how much it will cost to clean your house or yard in total (in dollars). Simply enter some information into the calculator, and it will calculate the cost based on the setup supplied in the code.

Requirement Specifications

In order to run the Python code, you must have some things installed in your system. The table given below would provide all the programmes along with versions that need to be installed in order to run this *House / Yard Cleaning Cost Calculator*.

Softwares	Versions
Python	3.x
Git	2.x

As specified above, you can install Python 3.x to run the code in your local system, you also need to install the latest version (2.x) of Git to clone the code repository for the *House/Yard Cleaning Cost Calculator*.

Source Code

All the source code with the assignment description would be available on https://github.com/Sparsh-Kumar/PythonAssignment

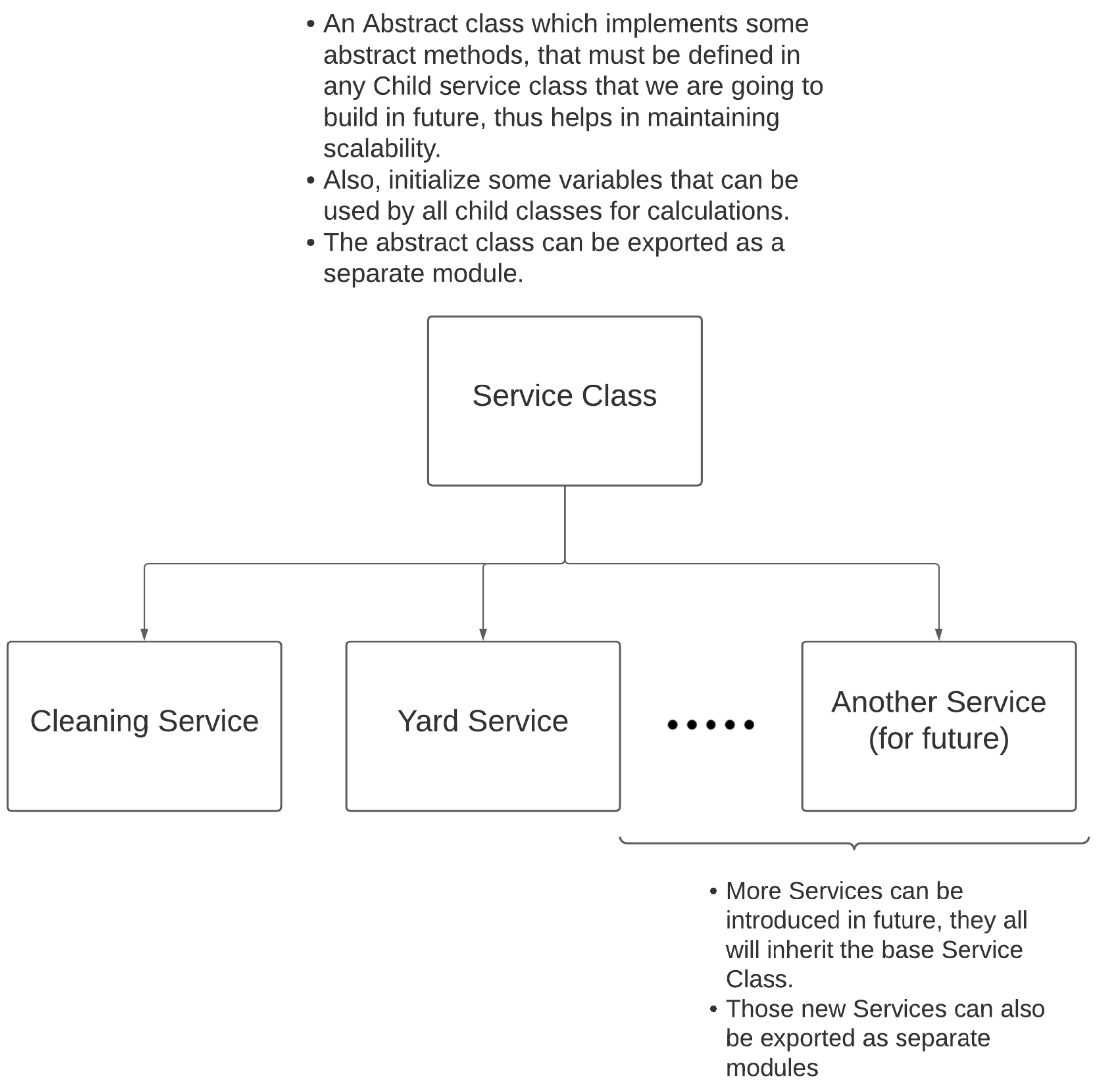
Application Implementation

Now let's discuss the deep implementation and architecture of the *House/Yard Cleaning Cost Calculator* that we have built. We have made use of the OOP (Object Oriented Programming) paradigm in order to define relationships between different components of the application.

Now we are going to introduce 3 classes in our code.

- **Service** - This class is basically an Abstract Class which is responsible for defining a well defined structure of what each Service class should look like.This class is a Parent class of all types of Service classes. It also initializes some values which are being used by Child Classes (Cleaning Service & Yard Service).
- **CleaningService** - This class is a child class of Service class, this class is responsible for calculating the overall cost involved in cleaning of Houses after applying discount (in % age) according to the conditions specified in the code file.
- **YardService** -This class is a child class of Service class, this class is responsible for calculating the overall cost involved in mowing, edging or shrub pruning of a Yard after applying discount (in %age) according to the conditions specified in the code file.

  The UML diagram specifying the relationship between these 3 classes is given below.


• An Abstract class which implements some abstract methods, that must be defined in any Child service class that we are going to build in future, thus helps in maintaining scalability.
• Also, initialize some variables that can be used by all child classes for calculations.
• The abstract class can be exported as a separate module.
Service Class
Cleaning Service
Yard Service
• • • • •
Another Service (for future)
• More Services can be introduced in future, they all will inherit the base Service Class.
• Those new Services can also be exported as separate modules

As you can see from the above diagram, all other types of new service that we will introduce in future will also inherit the Base Service Class.

Service Class

Now let's take a look at the code specified in the service class. We are also using a Python module named abc, this module helps us to implement abstract class functionality in Python.

```python
from abc import ABC, abstractmethod

# Service Abstract Class.
# All variables and methods defined in this class are available to child classes.
class Service ( ABC ):
	def __init__ (self, customerAge: int) -> None:
		self.customerAge = customerAge
		self.SENIOR_CITIZEN_AGE = 40
	def isSeniorCitizen (self) -> bool:
		return self.customerAge >= self.SENIOR_CITIZEN_AGE
	@abstractmethod
	def calculateDiscount (self) -> float:
		pass;
	@abstractmethod
	def getCost (self) -> float:
		pass;
	def __del__ (self) -> None: # Destructor
		pass;
```

Here on initialization, we are making two variables named ***customerAge*** and ***SENIOR_CITIZEN_AGE.*** As this class would be the Parent class of all other service classes, therefore these values can be inherited by the Child classes and can be used for computational purposes. For example the Child class can make use of ***isSeniorCitizen()*** to check whether to apply senior citizen discounts on that particular customer or not.

Here this class is also restricting all child classes to implement ***calculateDiscount()*** and ***getCost()*** as they must be present in all classes to compute discounts and overall cost.

CleaningService Class

The ***CleaningService*** class is responsible for calculating the overall amount needed for the cleaning of the House / Rooms. The structure of the CleaningService class is given below.

```python
# Cleaning Service is a type of Service
class CleaningService ( Service ):
         def __init__ (self, customerAge: int, numberOfRooms: int, fullCleaning:
bool) -> None:
             Service.__init__ (self, customerAge)
             self.numberOfRooms = numberOfRooms
             self.fullCleaning = fullCleaning
             if fullCleaning:
                 self.costOfCleaningOneRoom = 20 # In Dollars
             else:
                 self.costOfCleaningOneRoom = 10 # In Dollars
             self.seniorCitizenDiscountOnCleaning = 2
         def calculateDiscount (self) -> float:
             totalDiscount = self.getDiscountBasedOnHouseRooms()
             if self.isSeniorCitizen():
                 totalDiscount = totalDiscount +
self.seniorCitizenDiscountOnCleaning
             return totalDiscount
         def getDiscountBasedOnHouseRooms (self) -> float:
             discount = 0
             if self.numberOfRooms in range (3, 8):
                 discount = 1 # In percentage
             elif self.numberOfRooms in range (8, 14):
                 discount = 1.5 # In percentage
             elif self.numberOfRooms > 13:
                 discount = 2.5 # In percentage
             return discount
         def getCost (self) -> float:
             totalDiscount = self.calculateDiscount()
             totalCost = self.costOfCleaningOneRoom * self.numberOfRooms;
             return ((100 - totalDiscount)/100) * totalCost
         def __del__ (self) -> None:
             pass;
```

As you can see from the above implementation of CleaningService class, it has 4 methods.

- **Constructor** - While initializing the instance of ***CleaningService*** class, this method would get called. This method in turn called the constructor method of Parent Class i.e ***ServiceClass***. It also initialized some values like ***numberOfRooms***, ***fullCleaning, seniorCitizenDiscountOnCleaning***.
- **calculateDiscount()** - This method is responsible for calculating the overall discount that is available for a particular customer. The discount that is available for customers is based on 2 things.

1. The Number of rooms for which the cleaning is required, the number of rooms basically decides how much discount is applicable for that particular customer.
2. The customer is a Senior Citizen or not, if the customer is a Senior Citizen then an additional discount is applicable for that customer.

NOTE - All Discounts are in % age value.
The discount values for both the above specified cases are already included in the code.

- **getDiscountBasedOnHouseRooms()** - As we have already discussed, the number of rooms basically decides how much discount is applicable for that particular customer.
  Currently the configured value in the application is
    - If 3 <= numberOfRooms <=7, discount = 1 %
    - If 8 <= numberOfRooms <=13, discount = 1.5 %
    - If 14 <= numberOfRooms, discount = 2.5 %

  You can easily customize these values.
- **getCost()** - This is the method which is responsible for calculating the overall cost involved. It subtracts the discounted amount from the full amount and returns the final amount value.
- **Destructor** - This is the Destructor method of the class, it is called whenever the instance of a class gets destroyed.

YardService Class

The ***YardService*** class is responsible for calculating the overall amount required for mowing, edging and shrub pruning of the yard. The structure of the YardService class is given below.

```
class YardService ( Service ):
      def __init__ (self, customerAge: int, yardLength: int, yardWidth: int,
yardService: str = None, numberOfShrubs: int = 0) -> None:
             Service.__init__ (self, customerAge)
             self.yardLength = yardLength # In Foot
             self.yardWidth = yardWidth # In Foot
```

```
        self.yardService = yardService # String value (1, 2 or 3)
        self.numberOfShrubs = numberOfShrubs
        self.costOfMowing = 10 # In Dollars
        self.costOfEdging = 20 # In Dollars
        self.costOfPruning = 20 # In Dollars
        self.seniorCitizenDiscountOnYardService = 3
    def calculateDiscount (self) -> float:
        totalDiscount = 0
        if self.isSeniorCitizen():
            totalDiscount = self.seniorCitizenDiscountOnYardService
        return totalDiscount
    def calculateSquareFoot (self) -> float:
        return self.yardLength * self.yardWidth
    def calculateLinearFoot (self) -> float:
        return self.yardLength
    def getCost (self) -> float:
        totalDiscount = self.calculateDiscount()
        totalCost = 0
        if self.yardService == '1':
            totalCost = self.costOfMowing * self.calculateSquareFoot()
        elif self.yardService == '2':
            totalCost = self.costOfEdging * self.calculateLinearFoot()
        elif self.yardService == '3':
            totalCost = self.costOfPruning * self.numberOfShrubs
        return ((100 - totalDiscount)/100) * totalCost
    def __del__ (self) -> None:
        pass;
```

As you can see from the above implementation of the YardService class, it has 6 methods.

- **Constructor** - While initializing the instance of ***YardService*** class, this method would get called. This method in turn called the constructor method of Parent Class i.e ***ServiceClass*** and also initialized some values like ***yardLength***, ***yardWidth***, ***yardService***, ***numberOfShrubs***, ***costOfMowing***, ***costOfEdging***, ***costOfPruning***, ***seniorCitizenDiscountOnYardService***.
- **calculateDiscount()** - This method is responsible for calculating the overall discount that is available for a particular customer. The discount is applicable if the customer is a Senior Citizen.

- **calculateSquareFoot()** - This method is used to calculate the square foot value of the yard by making use of ***yardLength*** and ***yardWidth***. The **squareFoot** value of the yard = ***yardLength*** * ***yardWidth***.
- **calculateLinearFoot()** - The method is used to calculate the linear foot value of the yard, the **linearFoot** value of the yard = ***yardLength***.
- **getCost()** - This is the method which is responsible for calculating the overall cost involved. It subtracts the discounted amount from the full amount and returns the final amount value.
- **Destructor** - This is the Destructor method of the class, it is called whenever the instance of a class gets destroyed.

Main Method

The main method would be the point of execution for the programme, it will take user input and initialize the appropriate class based on user choices. The whole structure of the Main method is given below.

```
def getServiceType() -> None:
       serviceType = input('Please Enter 1 for Cleaning Service and 2 for Yard Service.\n')
       if serviceType not in ['1', '2']:
              return getServiceType();
       return serviceType

def main():
       serviceType = getServiceType();
       serviceInstance = None
       if serviceType == '1':
              numberOfRooms = int (input('Enter the number of rooms in the house.\n'))
              cleaningType = input('Please enter 1 for Full Cleaning and 2 for Light
Cleaning.\n')
              customerAge = int (input('Please enter your age.\n'))
              if cleaningType == '1':
                     fullCleaning = True
              else:
                     fullCleaning = False
              serviceInstance = CleaningService(customerAge, numberOfRooms, fullCleaning)
       elif serviceType == '2':
              customerAge = int (input('Please enter your age.\n'))
              yardLength = int (input('Enter the length of the yard (in ft.).\n'))
              yardWidth = int (input('Enter the width of the yard (in ft.).\n'))
              yardService = input ('Enter 1 for mowing, 2 for edging, 3 for shrub
pruning.\n')
```

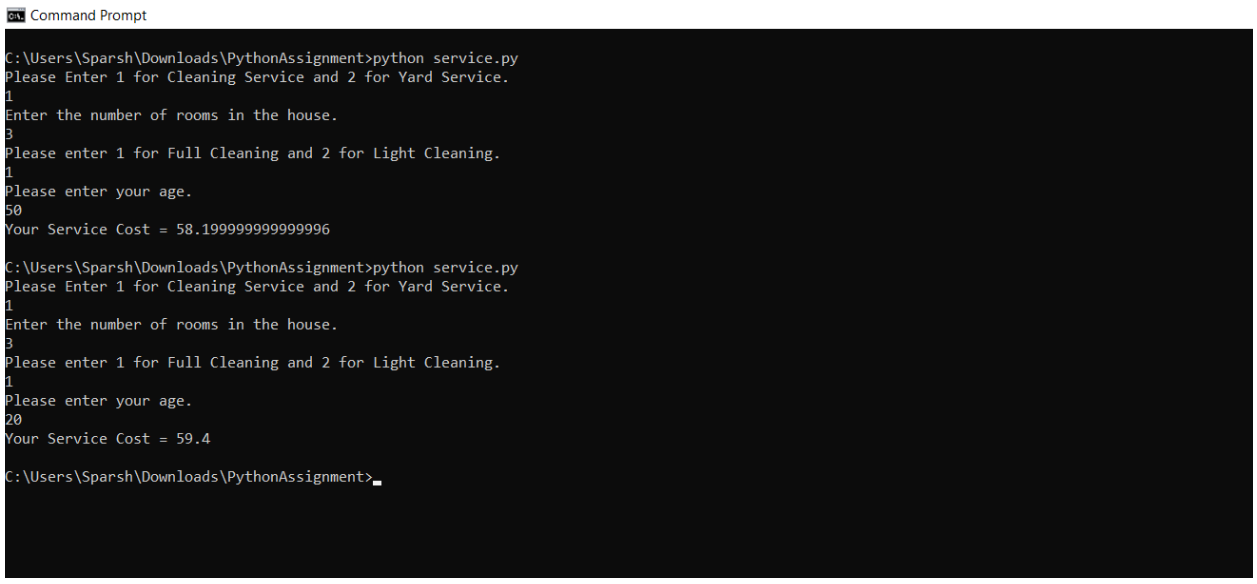
```
            numberOfShrubs = 0
            if yardService == '3':
                  numberOfShrubs = int (input('Please enter number of shrubs for shrub
pruning.\n'))
            serviceInstance = YardService(customerAge, yardLength, yardWidth,
yardService, numberOfShrubs)
      serviceCost = serviceInstance.getCost()
      print (f'Your Service Cost = {serviceCost}')
```

As you can see from the above code, the Main method is calling getServiceType() which is responsible for repeatedly prompting the user for input until a valid input has been received.
Based on the options that the user selects on prompt, the Main method would initialize the object of the appropriate class and start the execution.

NOTE - While calculating the cost of YardService, we are making an assumption that the Yard would always be rectangular or square shaped.

Final Result

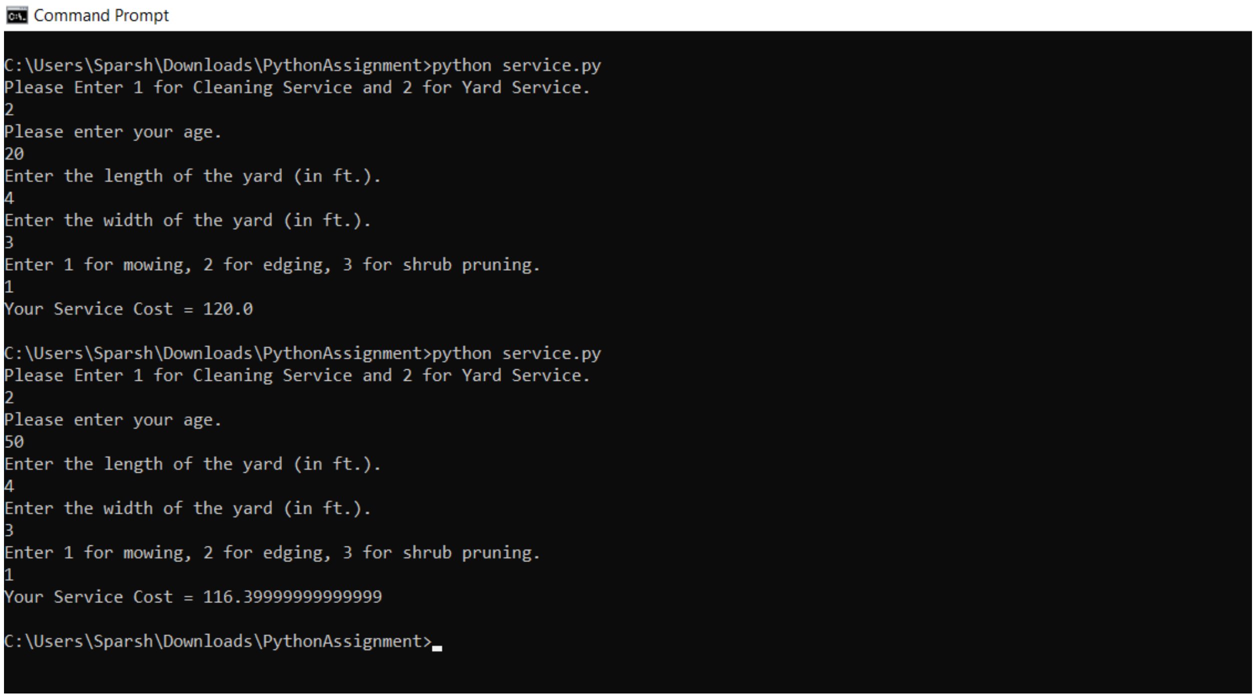
The screenshots of the execution of the resulting code is given below.


Command Prompt

```
C:\Users\Sparsh\Downloads\PythonAssignment>python service.py
Please Enter 1 for Cleaning Service and 2 for Yard Service.
1
Enter the number of rooms in the house.
3
Please enter 1 for Full Cleaning and 2 for Light Cleaning.
1
Please enter your age.
50
Your Service Cost = 58.199999999999996

C:\Users\Sparsh\Downloads\PythonAssignment>python service.py
Please Enter 1 for Cleaning Service and 2 for Yard Service.
1
Enter the number of rooms in the house.
3
Please enter 1 for Full Cleaning and 2 for Light Cleaning.
1
Please enter your age.
20
Your Service Cost = 59.4

C:\Users\Sparsh\Downloads\PythonAssignment>
```


Command Prompt

```
C:\Users\Sparsh\Downloads\PythonAssignment>python service.py
Please Enter 1 for Cleaning Service and 2 for Yard Service.
2
Please enter your age.
20
Enter the length of the yard (in ft.).
4
Enter the width of the yard (in ft.).
3
Enter 1 for mowing, 2 for edging, 3 for shrub pruning.
1
Your Service Cost = 120.0

C:\Users\Sparsh\Downloads\PythonAssignment>python service.py
Please Enter 1 for Cleaning Service and 2 for Yard Service.
2
Please enter your age.
50
Enter the length of the yard (in ft.).
4
Enter the width of the yard (in ft.).
3
Enter 1 for mowing, 2 for edging, 3 for shrub pruning.
1
Your Service Cost = 116.39999999999999

C:\Users\Sparsh\Downloads\PythonAssignment>
```