Program: 05

Aug 12, 2022

Problem Statement:

Define a class BOOK with the following specifications :

Members of the class BOOK areBook_no, Book_title, Price. total_cost():A function to calculate the total cost for N number of copies where N is passed to the function as argument. Input(): function to read Book_no, Book_title, Price Purchase() function to ask the user to input the number of copies to be purchased. It invokes total_cost() and prints the total cost to be paid by the user.

Solution:

Solution:

```
class Book:
    def __init__(self):
        self.Input()
        self.Purchase()
    def Input(self):
        self.Book_no=input("Enter Book Number: ")
        self.Book_title=input("Enter Book Title: ")
        self.Price=input("Enter Price: ")
    def Purchase(self):
        self.N=int(input("Number of copies to be purchased: "))
        self.total_cost()
    def total_cost(self):
        print("Total cost to be paid:",float(self.Price)*int(self.N))

b1=Book()
```

Program: 06

Aug 18, 2022

Problem Statement:

Write a Python class to find validity of a string of parentheses, '(', ')', '{', '}', '[' and ']. These brackets must be close in the correct order.

Solution:

Solution:

```
class validity:
    def f(str):
        a = ['()', '{}', '[]']
        while any(i in str for i in a):
            for j in a:
                str = str.replace(j, '')
        return not str
s = input("enter : ")
print(s,"is valid" if validity.f(s) else "is Not valid")
```

Program: 07

Aug 18, 2022

Problem Statement:

Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay Rs. 50/- toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected. Model this tollbooth with a class called TollBooth. The two data items are a type int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both of these to 0. A member function called payingCar() increments the car total and adds 0.50 to the cash total. Another function, called nopayCar(), increments the car total but adds nothing to the cash total. Finally, a member function called display() displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the $ key should cause the program to print out the total cars and total cash and then exit.

Solution:

```
class ToolBooth:
    def __init__(self):
        ToolBooth.cars=0
        ToolBooth.amnt=0
    def payingCar(self):
        self.cars=self.cars+1
        self.amnt=self.amnt+50
    def nopayCar(self):
        self.cars=self.cars+1
    def display(self):
        print('\nNumber of cars passed: ',self.cars)
        print('total cash collected: ',self.amnt)
toll=ToolBooth()
a=0
while(a!='end') :
    a=input("Enter 'p' for paying car, 'n' for non-paying car,'$' to display and 'end' to stop:")
    if a=='p':
        toll.payingCar()
    elif a=='n':
        toll.nopayCar()
    elif a=='$':
        toll.display()
    else:
        pass
```

Program: 08

Aug 25, 2022

Problem Statement:

Create a class called employee that contains a name and an employee number. Include a member function called getdata() to get data from the user for insertion into the object, and another function called putdata() to display the data. Assume the name has no embedded blanks. Write a main() program to exercise this class. It should create an array of type employee, and then invite the user to input data for up to 100 employees.
Finally, it should print out the data for all the employees.

Solution:

```
l1 = []
class Employee:
    def getData(self):
        self.Ename = input("Enter Employee Name : ")
        self.Eno   = int(input("Enter Employee Number : "))
        t = self.Ename ,self.Eno
        l1.append(t)
    def putData(self):
        for i in l1:
            for j in i:
                print(j,end=' ')
            print()
n = int(input("Enter number of Emloyees : "))
e = Employee()
for i in range(n):
    e.getData()
    e.putData()
```

Program: 09

Aug 25, 2022

Problem Statement:

Write a program to maintain the record of movies, one record of movie contains movie name, actor or actress, movie rating, production house, more than one record can be inserted by a operator and all records should be displayed to user only.

Solution:

```
class Movies:
    def __init__(self):
        self.name = ""
        self.actor = ""
        self.rating = ""
        self.prohouse= ""

    def insert(self, name, actor, rating, prohouse):
        self.name = name
        self.actor = actor
        self.rating = rating
        self.prohouse = prohouse

    def display(self):
        print("name : ", self.name)
        print("actor and actress : ",self.actor)
        print("rating : ",self.rating)
        print("production house : ", self.prohouse)

a = Movies()
a.insert("Avengers : Endgame ","Robert Downey Jr","10", "Marval Studios")
a.display()
```

Program: 10

Oct 07, 2022

Problem Statement:

Create a class that includes a data member that holds a serial number for each object created from a class. That is the first object will be numbered 1, the second 2 and so on. When each object is creating, its constructor can examine this count member variable to determine the appropriate serial number for the new object. Add a member function that permits an object to report its own serial number. Then the main () function that creates three objects and queries each one about its serial number. they should respond i am object 2, and so on.

Solution:

```
sn = 1
class SerialNumber():
    def __init__(self,name):
        global sn
        self.name = name
        self.sn = sn
        sn += 1
    def display(self):
        print("name : ",self.name)
        print("Serial number : ",self.sn)

a = SerialNumber("Deepak")
b = SerialNumber("Ravi")
c = SerialNumber("Rampal")

a.display()
b.display()
```

Program: 11

Oct 07, 2022

Problem Statement:

Imagine a publishing company that markets both book and compact disk version of its work. Create a class publication that stores the title and price of a publication. From this class derive two classes book which adds a page count and CD which adds a playing timein minutes. Each of these three classes should have a getData() function to get its data from the userat the keyboard and a putData() function to display the data.Write a main program to test the book and CD classes by creating instances of them.Asking the user to fill in thedata with getData() and displaying the data with putData().

Solution:

```
class Publication:
    def getData(self):
        self.title=input("Enter Title: ")
        self.price=input("Enter Price: ")
    def putData(self):
        print("\nTitle:",self.title)
        print("Price:",self.price)
class Book(Publication):
    def getData(self):
        super().getData()
        self.pg=input("Enter page count: ")
    def putData(self):
        super().putData()
        print("Page Count:",self.pg)
class CD(Publication):
    def getData(self):
        super().getData()
        self.playT=input("Enter playing time: ")
    def putData(self):
        super().putData()
        print("Playing Time:",self.playT,"min")


def main():
    cd=CD()
    b=Book()
    cd.getData()
    cd.putData()
    b.getData()
    b.putData()
main()
```