1. Problem: Calculator Description: *Create a simple calculator class that can perform basic arithmetic operations (addition, subtraction, multiplication, division).*

```
class Calculator:
   def __init__(self):
      pass
   def add(self, num1, num2):
      return num1 + num2
   def subtract(self, num1, num2):
      return num1 - num2
   def multiply(self, num1, num2):
      return num1 * num2
   def divide(self, num1, num2):
      if num2 != 0:
         return num1 / num2
      else:
         return "Error: Division by zero."
# Example Usage:
calc = Calculator()
print("Addition:", calc.add(5, 3))
print("Subtraction:", calc.subtract(5, 3))
print("Multiplication:", calc.multiply(5, 3))
print("Division:", calc.divide(5, 3))
```

2. Problem: Student Management System Description: *Implement a Student class with attributes like name, roll number, and methods for displaying student details.*

```
class student:
	def __init__(self,name,rollno):
		self.name=name
		self.rollno=rollno
	def display(self):
		print(f'Name : {self.name}\nRollno : {self.rollno}')
det=student("Ishu Agrawal",24)
det.display()
```

3. Problem: Employee Management System Description: *Design an Employee class with attributes like name, employee ID, and methods for calculating salary based on hours worked.*

```
class Employee:
    def __init__(self,name,employee_ID,hour):
        self.name=name
        self.employee_ID=employee_ID
        self.hour=hour
    def display(self):
        print(f'{self.name:<17} {self.employee_ID:<17} {self.hour:<17} Rs.{self.hour*77:<17}
')
det=Employee("Ishu Agrawal",101,7)
print("NAME".ljust(17," "),"Employee_ID".ljust(17," "),"Working_hrs".ljust(17," "),"Salary".ljust(17,"
"),"\n")
det.display()
```

4. Problem: ToDo List Description: *Create a ToDo class with methods to add tasks, mark tasks as completed, and display the list of tasks.*

```
class TODO:
    def __init__(self):
        self.task_to_do=[]
    def add(self,add_task):
        self.add_task=add_task
        self.task_to_do.append(self.add_task)
    def comp(self,c_task):
        del self.task_to_do[c_task-1]
    def display(self):
        if self.task_to_do==[]:
            print("\nAll task completed")
        else:
            print("\nToday's tasks\n")
            for i in range(0,len(self.task_to_do)):
                print(f'{i+1}. {self.task_to_do[i]}')
tks=TODO()
tks.add("12:00am-7:00am  sleep")
tks.add("7:00am-7:15am  eat")
tks.add("8:00am-3:30pm  college")
tks.add("4:30pm-4:45pm  eat")
tks.add("5:00pm-7:00pm  sleep")
tks.add("7:00pm-7:15pm  eat")
tks.add("7:15pm-9:00pm  Entertainemt")
tks.add("9:00pm-9:30pm  Shop")
tks.add("9:30pm -12pm  study+Entertainment")
tks.display()
tks.comp(1)
tks.display()
```

5. Problem: Car Rental System Description: *Develop a Car class for a rental system, including attributes like model, color, and methods for calculating rental charges.*

```
class car:
    def __init__(self,model,color,travel_distance):
        self.model=model
        self.color=color
        self.travel_distance=travel_distance
    def display(self):
        print(f'{self.model:<17} {self.color:<17} {self.travel_distance}{"km":<16}
Rs.{self.travel_distance*121:<17}')
det=car("Alto 800","White",2)
print("Model".ljust(17," "),"Color".ljust(17," "),"Distance Travel".ljust(17," "),"Rent".ljust(17," "),"\n")
det.display()
```

6. Problem: Bank Transaction System Description: *Build a BankAccount class with methods for deposit, withdraw, and check balance.*

```
class BankAccount:
    def __init__(self,name):
        self.name=name
        self.balance=0
    def deposit(self,money):
        self.balance+=money
    def withdraw(self,money):
        if self.balance>=money:
            self.balance-=money
        else:
            print("!!!YOU DON'T HAVE ENOUGH BALANCE!!!")
    def display(self):
        print(f'{self.name} your Available Balance is: ',self.balance)
bank=BankAccount("Ishu Agrawal")
bank.deposit(1000)

bank.display()
bank.withdraw(500)
bank.display()
bank.withdraw(1000)
bank.display()
```

7. Problem: Shape Area Calculator Description: *Implement a Shape class with methods to calculate the area for different shapes like square, rectangle, and circle.*

```
class Shape:
    def __init__(self):
        pass
    def sq(self,side):
        print(f'Area of square with side {side} :- ',side**2)
    def crl(self,radius):
        print(f'Area of Circle with radius {radius} :- ',3.14*radius**2)
    def rec(self,lgth,bdth):
        print(f'Area of Rectangle with Length:{lgth} & Breadth:{bdth} :-',lgth*bdth)
sp=Shape()
sp.sq(12)
sp.crl(10)
sp.rec(11,12)
```

8. Problem: Time Converter Description: *Create a Time class that converts seconds to minutes, minutes to hours, and vice versa.*

```
class Time:
    def stm(self,s):
        print(f'{s} seconds is equal to {s/60} minutes')
    def mth(self,m):
        print(f'{m} minutes is equal to {m/60} hours')
    def hts(self,h):
        print(f'{h} hours is equal to {h*3600} seconds')

t=Time()
t.stm(72)
```

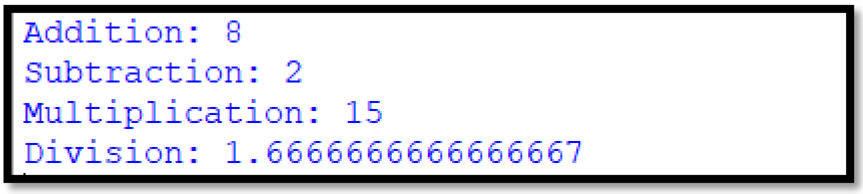
```
t.mth(72)
t.hts(1.2)
```

9. Problem: Dice Simulator Description: *Develop a Dice class that simulates rolling a six-sided die.*

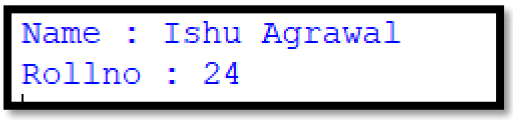
```
import random
class Dice:
    def __init__(self):
        self.num=random.randint(1,6)
        print("Your number is :",self.num)
Dice()
```

10. Problem: Library Book Management Description: *Design a LibraryBook class with attributes like title, author, and availability status. Implement methods for borrowing and returning books*

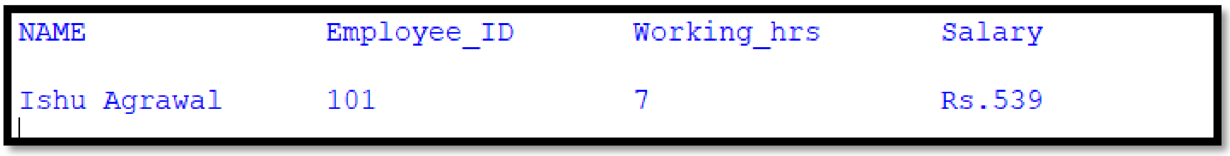
```
class LibraryBook:
    def __init__(self,title,author,stock):
        self.title=title
        self.author=author
        self.stock=stock
        self.available=stock
    def borrow(self,title,author):
        if title.lower() == self.title.lower()and self.available>0:
            self.available-=1
            print(f'You borrow {self.title} by {self.author}')
        else:
            print(f'currently {self.title} by {self.author} not Available')
    def retn(self,title,author):
        if self.available<self.stock:
            self.available+=1
            print(f'You return {self.title} by {self.author}')
        else:
            print("This book is not from our Librabry")
lb=LibraryBook("Computer Science with Python","Sumita Arora",2)
lb.borrow("Computer Science with Python","Sumita Arora")
lb.retn("Computer Science with Python","Sumita Arora")
lb.retn("Computer Science with Python","Sumita Arora")
```



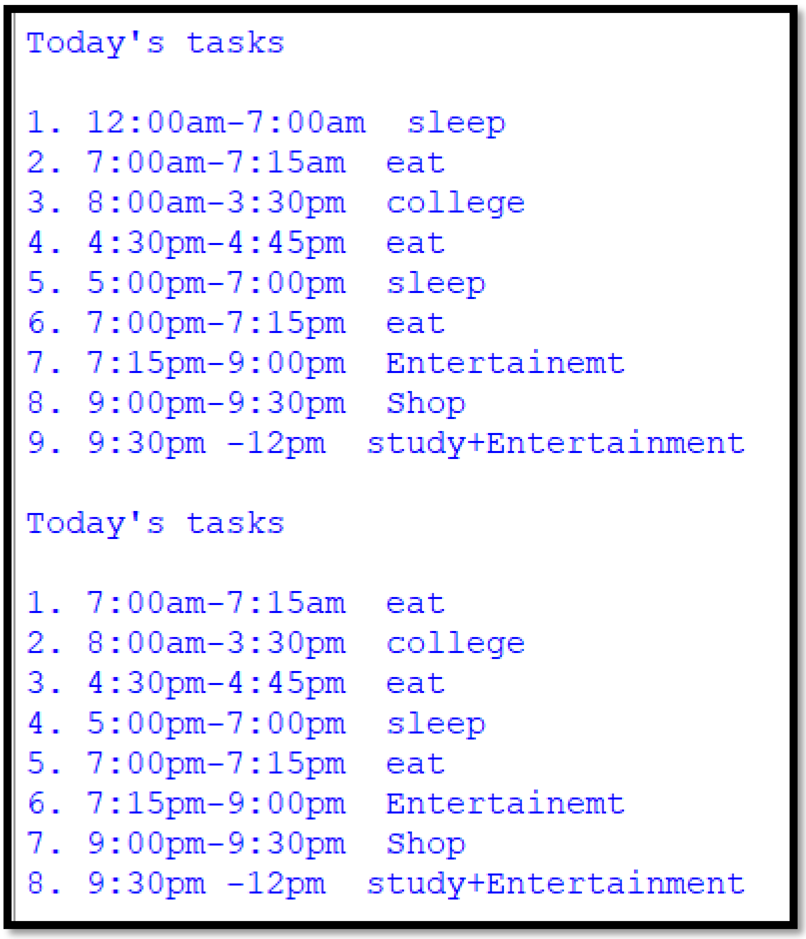
```
Addition: 8
Subtraction: 2
Multiplication: 15
Division: 1.6666666666666667
```



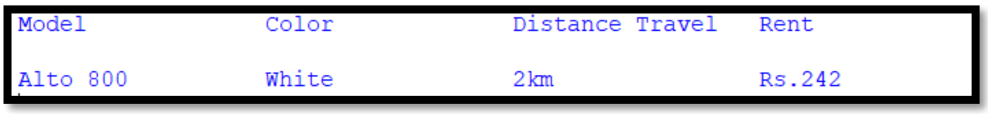
```
Name : Ishu Agrawal
Rollno : 24
```



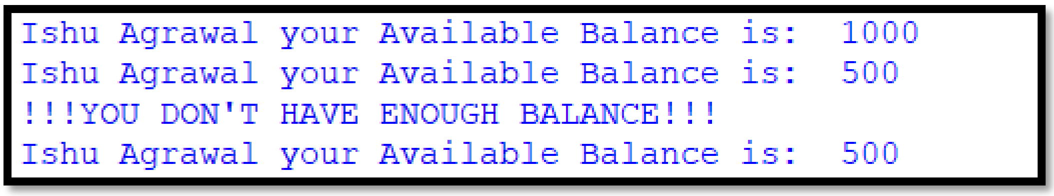
```
NAME                Employee_ID         Working_hrs         Salary

Ishu Agrawal        101                 7                   Rs.539
```



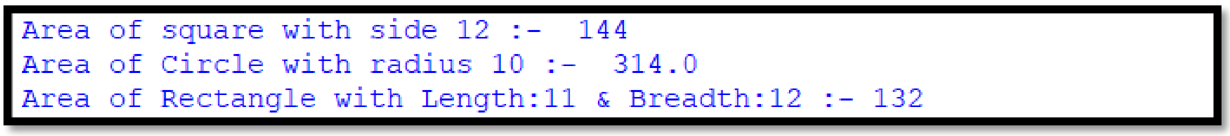
```
Today's tasks

1. 12:00am-7:00am  sleep
2. 7:00am-7:15am  eat
3. 8:00am-3:30pm  college
4. 4:30pm-4:45pm  eat
5. 5:00pm-7:00pm  sleep
6. 7:00pm-7:15pm  eat
7. 7:15pm-9:00pm  Entertainemt
8. 9:00pm-9:30pm  Shop
9. 9:30pm -12pm  study+Entertainment

Today's tasks

1. 7:00am-7:15am  eat
2. 8:00am-3:30pm  college
3. 4:30pm-4:45pm  eat
4. 5:00pm-7:00pm  sleep
5. 7:00pm-7:15pm  eat
6. 7:15pm-9:00pm  Entertainemt
7. 9:00pm-9:30pm  Shop
8. 9:30pm -12pm  study+Entertainment
```



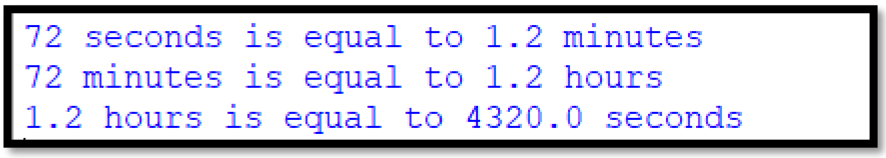
```
Model               Color               Distance Travel     Rent

Alto 800            White               2km                 Rs.242
```



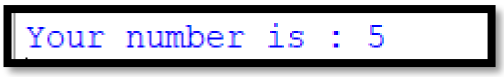
```
Ishu Agrawal your Available Balance is:  1000
Ishu Agrawal your Available Balance is:  500
!!!YOU DON'T HAVE ENOUGH BALANCE!!!
Ishu Agrawal your Available Balance is:  500
```



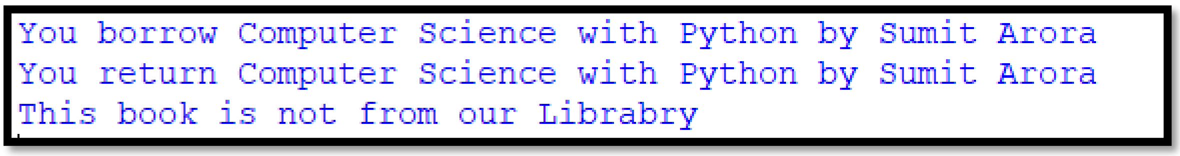
```
Area of square with side 12 :-  144
Area of Circle with radius 10 :-  314.0
Area of Rectangle with Length:11 & Breadth:12 :- 132
```



```
72 seconds is equal to 1.2 minutes
72 minutes is equal to 1.2 hours
1.2 hours is equal to 4320.0 seconds
```



```
Your number is : 5
```



```
You borrow Computer Science with Python by Sumit Arora
You return Computer Science with Python by Sumit Arora
This book is not from our Librabry
```