

## ▼ Experiment 8

### AIM

Write an Object-Oriented Python program to

1. Create two Time objects
  - Current Time, which contains the current time, and
  - Bread Time, which contains the amount of time it takes for a bread maker to make bread
2. Then use add\_time to determine when the bread will be done.
3. Write the print\_time function to display the time when the bread will be done by the bread maker.

### Description

#### OOP:

Object-oriented programming (OOP) is a programming paradigm built on the concept of objects that contain both data and code to modify the data. Object-oriented programming mimics a lot of the real-world attributes of objects.

#### Class:

A class is a collection of objects. A class contains the blueprints or the prototype from which the objects are being created. It is a logical entity that contains some attributes and methods.

Syntax:

```
class class_name:
```

#### Object:

The object is an entity that has a state and behavior associated with it. It may be any real-world object like a mouse, keyboard, chair, table, pen, etc. Integers, strings, floating-point numbers, even arrays, and dictionaries, are all objects.

#### Methods:

Methods are functions defined inside the body of a class. They are used to define the behaviors of an object

## ▼ Program

```

'''
Define class Time containing
3 instance variables- hrs,mins,sec
a method print_time() to print time as hrs:mins:sec
a method correct_time() to maintain correct time format so that mins<60 and sec<60
'''

class Time:
    def __init__(self,hrs = 00,mins = 00,sec = 00):
        self.hrs = hrs
        self.mins = mins
        self.sec = sec

    def print_time(self):
        print(self.hrs, ':', self.mins, ':', self.sec)

    def correct_time(self):
        while True:
            if self.mins>60:
                self.mins-=60
                self.hrs+=1
            elif self.sec>60:
                self.sec -= 60
                self.mins += 1
            else:
                break

# Create object for Current Time and print the current time
current_time = Time(9,45,30)
current_time.correct_time()
current_time.print_time()

9 : 45 : 30

# Create object for Bread Time and print the bread time
bread_time = Time(30,53,82)
bread_time.correct_time()
bread_time.print_time()

30 : 54 : 22

'''
Define function add_time() to add two time objects and
return a time object containing total time after correcting its format
'''

def add_time(t1: Time,t2: Time):
    hours = t1.hrs+t2.hrs
    minutes = t1.mins + t2.mins

```

```
minutes = t1.minutes + t2.minutes
seconds = t1.sec + t2.sec
added_time = Time(hours,minutes,seconds)
added_time.correct_time()
return added_time

# Use add_time function to add current time and bread time. Print the Total Time
total_time =add_time(current_time, bread_time)
total_time.print_time()

40 : 39 : 52
```

## ▼ Conclusion

Hence, we have created the class Time, created instances of the class Time, current\_time and bread\_time, printed them using print\_time() method of the class Time. Then we added current\_time and bread\_time using add\_time() and printed it using print\_time()

## Evaluation

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		