```
class calculator():

    def __init__(self):
        self.prevAns = None

    def adder(self):
        input1 = float(input("input1: "))
        truncd1 = roundHalfUp(input1)
        if almostEqual(input1,truncd1):
            input1 = truncd1
        input2 = float(input("input2: "))
        truncd2 = roundHalfUp(input2)
        if almostEqual(input1,truncd2):
            input2 = truncd2
        self.prevAns = input1 + input2
        truncdAns = int(self.prevAns)
        if almostEqual(self.prevAns,truncdAns):
            self.prevAns = int(self.prevAns)
        return self.prevAns

    def subtractor(self):
        input1 = float(input("input1: "))
        truncd1 = roundHalfUp(input1)
        if almostEqual(input1,truncd1):
            input1 = truncd1
        input2 = float(input("input2: "))
        truncd2 = roundHalfUp(input2)
        if almostEqual(input1,truncd2):
            input2 = truncd2
        self.prevAns = input1 - input2
        truncdAns = int(self.prevAns)
        if almostEqual(self.prevAns,truncdAns):
            self.prevAns = int(self.prevAns)
        return self.prevAns

    def multiplier(self):
        input1 = float(input("input1: "))
        truncd1 = roundHalfUp(input1)
        if almostEqual(input1,truncd1):
            input1 = truncd1
        input2 = float(input("input2: "))
```

```
42          truncd2 = roundHalfUp(input2)
43          if almostEqual(input1,truncd2):
44              input2 = input2
45          self.prevAns = input1 * input2
46          truncdAns = int(self.prevAns)
47          if almostEqual(self.prevAns,truncdAns):
48              self.prevAns = int(self.prevAns)
49          return self.prevAns
50
51      def clear(self):
52          self.prevAns = 0
53
54          return self.prevAns
55
56      def divider(self):
57          try:
58              input1 = float(input("input1: "))
59              input2 = float(input("input2: "))
60              truncd1 = roundHalfUp(input1)
61              if almostEqual(input2,0):
62                  raise ZeroDivisionError
63              if almostEqual(input1,truncd1):
64                  input1 = truncd1
65              truncd2 = roundHalfUp(input2)
66              if almostEqual(input1,truncd2):
67                  input2 = input2
68          except ZeroDivisionError:
69              return "Can't divide by 0!"
70          self.prevAns = input1/input2
71          truncdAns = int(self.prevAns)
72          if almostEqual(self.prevAns,truncdAns):
73              self.prevAns = int(self.prevAns)
74          return self.prevAns
75
76  class clockTime():
77
78      def __init__(self):
79          self.time = None
80          self.hours = None
81          self.minutes = None
82          self.seconds = None
```

```
 83
 84      def setHours(self):
 85          try:
 86              hours = int(input("Enter hours: "))
 87              if hours < 0:
 88                  raise ValueError
 89              if hours > 23:
 90                  raise moreThanAllowedError
 91          except (ValueError, moreThanAllowedError) as
     e:
 92              if isinstance(e, moreThanAllowedError):
 93                  print("Cannot be more than 23 Hours"
    )
 94                  return self.setHours()
 95              elif isinstance(e, ValueError):
 96                  print("Cannot have negative hours")
 97                  return self.setHours()
 98          self.hours = str(hours)
 99
100      def setMinutes(self):
101          try:
102              minutes = int(input("Enter Minutes: "))
103              if minutes < 0:
104                  raise ValueError
105              if minutes > 59:
106                  raise moreThanAllowedError
107          except (ValueError, moreThanAllowedError) as
     e:
108              if isinstance(e, moreThanAllowedError):
109                  print("Cannot be more than 59
    Minutes")
110                  return self.setMinutes()
111              elif isinstance(e, ValueError):
112                  print("Cannot have negative Minutes"
    )
113                  return self.setMinutes()
114          if minutes < 10:
115              self.minutes = "0"+str(minutes)
116          else:
117              self.minutes = str(minutes)
118
```

```
119     def setSeconds(self):
120         try:
121             seconds = int(input("Enter Seconds: "))
122             if seconds < 0:
123                 raise ValueError
124             if seconds > 59:
125                 raise moreThanAllowedError
126         except (ValueError, moreThanAllowedError) as
     e:
127             if isinstance(e, moreThanAllowedError):
128                 print("Cannot be more than 59
    Seconds")
129                 return self.setSeconds()
130             elif isinstance(e, ValueError):
131                 print("Cannot have negative Seconds"
    )
132                 return self.setSeconds()
133         if seconds < 10:
134             self.seconds = "0"+str(seconds)
135         else:
136             self.seconds = str(seconds)
137 
138     def setTime(self):
139         print("Set hours: ", end ="")
140         self.setHours()
141         print("Set minutes: ", end = "")
142         self.setMinutes()
143         print("Set Seconds: ", end = "")
144         self.setSeconds()
145 
146     def clearTime(self):
147         self.hours = None
148         self.minutes = None
149         self.seconds = None
150         return print("Successfully cleared timed")
151 
152     def showTime(self):
153         if self.hours == None or self.minutes ==
    None or self.seconds == None:
154             return "Time is not properly set yet"
155         return f'{self.hours}Hours:{self.minutes}
```

```
155 Minutes:{self.seconds}Seconds.'
156
157 class moreThanAllowedError(ValueError):
158     pass
159
160 def almostEqual(val1, val2, epsilon=10**-7):
161     return (abs(val1 - val2) < epsilon)
162
163 import decimal
164 def roundHalfUp(d): #helper function
165     rounding = decimal.ROUND_HALF_UP
166     return int(decimal.Decimal(d).to_integral_value(
    rounding=rounding))
167
168 print("Question 1:")
169 print("_________________")
170 parser = calculator()
171 print("For adder method: ")
172 print(parser.adder())
173 print("For subtractor method: ")
174 print(parser.subtractor())
175 print("For multiplier method: ")
176 print(parser.multiplier())
177 print("For divider method: ")
178 print(parser.divider())
179 print("For ZeroDivision error handling on divider
    method: ")
180 print(parser.divider())
181 print("For clear method: ")
182 print("Before: ")
183 print(parser.prevAns)
184 print("After: ")
185 parser.clear()
186 print(parser.prevAns)
187 print("\n")
188 print("Question 2: ")
189 print("_________________")
190 clock = clockTime()
191 print("For error Handling on undisplayable times")
192 print(clock.showTime())
193 print("For setHours method")
```

```
clock.setHours()
print("For setMinutes method")
clock.setMinutes()
print("For setSeconds method")
clock.setSeconds()
print("Displaying time:")
print(clock.showTime())
print("For setTime method")
clock.setTime()
print(clock.showTime())
print("For out of range settings")
clock.setTime()
print(clock.showTime())

```

```
"C:\Users\Desktop\Desktop\Random garbage\Python_Programming\python.exe" E:\Java\coding\IntelliJ\CSC1109Lab11\Lab11.py
Question 1:
_________________
For adder method:
input1: 5
input2: 10
15
For subtractor method:
input1: 5
input2: 10
-5
For multiplier method:
input1: 5
input2: 10
50
For divider method:
input1: 5
input2: 10
0.5
For ZeroDivision error handling on divider method:
input1: 5
input2: 0
Can't divide by 0!
For clear method:
Before:
0.5
After:
0


Question 2:
_________________
For error Handling on undisplayable times
Time is not properly set yet
For setHours method
Enter hours: 3
For setMinutes method
Enter Minutes: 5
For setSeconds method
```

```
40 Enter Seconds: 3
41 Displaying time:
42 3Hours:05Minutes:03Seconds.
43 For setTime method
44 Set hours: Enter hours: 23
45 Set minutes: Enter Minutes: 5
46 Set Seconds: Enter Seconds: 23
47 23Hours:05Minutes:23Seconds.
48 For out of range settings
49 Set hours: Enter hours: -1
50 Cannot have negative hours
51 Enter hours: 24
52 Cannot be more than 23 Hours
53 Enter hours: 20
54 Set minutes: Enter Minutes: -1
55 Cannot have negative Minutes
56 Enter Minutes: 60
57 Cannot be more than 59 Minutes
58 Enter Minutes: 44
59 Set Seconds: Enter Seconds: -1
60 Cannot have negative Seconds
61 Enter Seconds: 60
62 Cannot be more than 59 Seconds
63 Enter Seconds: 59
64 20Hours:44Minutes:59Seconds.
65
66 Process finished with exit code 0
67
```