

```

1 class Time:
2     def __init__(self, *args):
3         """
4         Time object constructor
5         :param args: variable number of arguments
6         1 parameter, "hours:mins" string expected
7         2 parameters, hours, mins numerical values expected
8         :return:
9         Time object
10        """
11        if len(args)>2:
12            raise ValueError("{} is not a valid input paramter, see: {}".
13                               format(str(args),Time.__init__.__doc__))
14        if len(args) == 1:
15            if not isinstance(args[0],str): # string input assumed
16                raise ValueError("when 1 parameter given, string is expected,"
17                                   " not {}, see {}".
18                                   format(str(args[0]),Time.__init__.__doc__))
19            strvalues = args[0].split(":")
20            values = [int(x) for x in strvalues]
21        else:
22            values = args
23        self._h = values[0]
24        self._m = values[1]
25
26        def __sub__(self,other):
27            diff = abs(self) - abs(other)
28            assert diff >= 0, "Time difference should non-negative, which is not for {}-{}".\
29                               format(str(self),str(other))
30            return mins_to_time(diff)
31
32        def in_minutes(self):
33            return self._h*60 + self._m
34
35        def __abs__(self):
36            return self.in_minutes()
37
38        def __str__(self):
39            return "{:02d}:{:02d}".format(self._h,self._m)
40
41        def __lt__(self,other):
42            return abs(self) < abs(other)
43        def __le__(self,other):
44            return abs(self) <= abs(other)
45        def __eq__(self,other):
46            return abs(self) == abs(other)
47        def __ne__(self,other):
48            return abs(self) != abs(other)
49        def __gt__(self,other):
50            return abs(self) > abs(other)
51        def __ge__(self,other):
52            return abs(self) >= abs(other)
53
54    def mins_to_time(mins):
55        h = mins//60
56        m = mins-h*60
57        return Time(h,m)
58
59    if __name__ == "__main__":
60        t1 = Time(3,4)
61        t2 = Time("05:56")
62        # t4 = Time([3,4,5])
63        try:
64            t3 = t1-t2
65        except AssertionError as err:

```

```
66     print("{}".format(err))
67     print("Warning: time difference negative, flipping operands time")
68     t3 = t2-t1
69     print(t1,t2,t3,abs(t3))
70
71
```