

Exercise 1: Rewrite your pay computation to give the employee 1.5 times the hourly rate for hours worked above 40 hours.

Enter Hours: 45

Enter Rate: 10

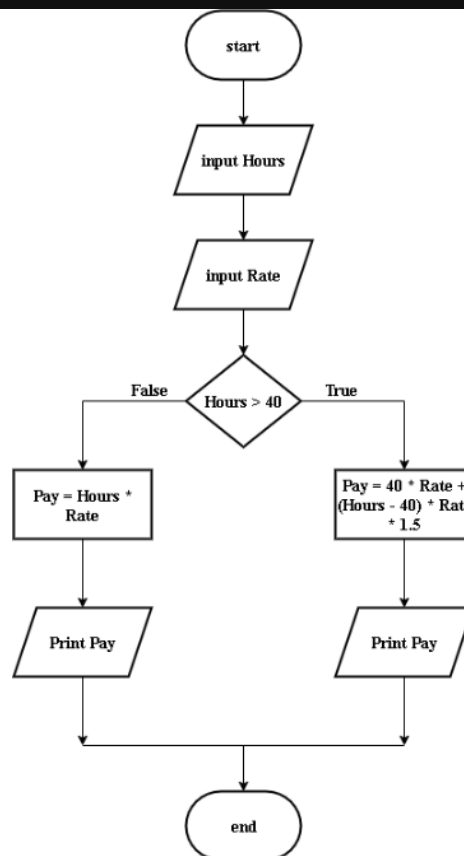
Pay: 475.0

The screenshot shows a Python IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code in the editor implements the logic for calculating pay based on hours worked and an hourly rate. The terminal shows the execution of the program with the inputs 45 for hours and 10 for rate, resulting in a pay of 475.0.

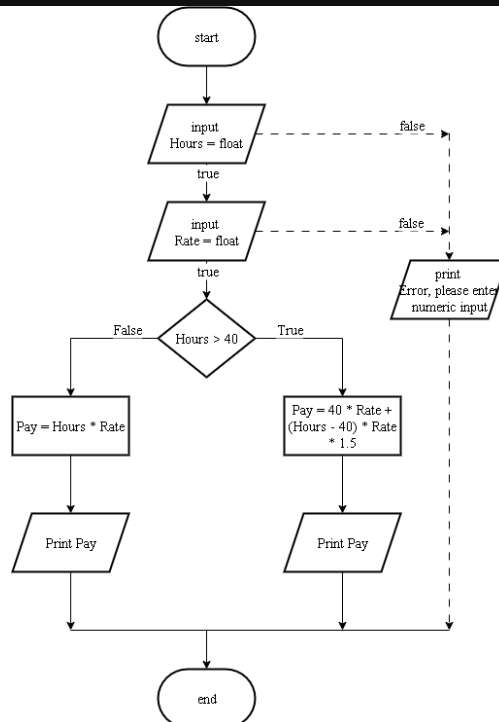
```
1 # Exercise 1
2 Hours = float(input('Enter Hours: '))
3 Rate = float(input('Enter Rate: '))
4 if Hours > 40:
5     Pay = 40 * Rate + (Hours - 40) * Rate * 1.5
6 else:
7     Pay = Hours * Rate
8 print("Pay:", Pay)
```

Terminal output:

```
PS D:\KTLT-NTTD-060206000031\NTTD-060206000031> & C:\Users\Admin\AppData\Local\Programs\Python\Python313\python.exe d:\KTLT-NTTD-060206000031\NTTD-060206000031\BT03/a.py
Enter Hours: 45
Enter Rate: 10
Pay: 475.0
PS D:\KTLT-NTTD-060206000031\NTTD-060206000031>
```



```
Enter Hours: forty
Error, please enter numeric input
```



Exercise 3: Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error message. If the score is between 0.0 and 1.0, print a grade using the following table:

Score	Grade
>= 0.9	A
>= 0.8	B
>= 0.7	C
>= 0.6	D
< 0.6	F

Enter score: 0.95 A ~~

Enter score: perfect
Bad score

Enter score: 10.0
Bad score

Enter score: 0.75
C

Enter score: 0.5
F

Run the program repeatedly as shown above to test the various different values for input.

The screenshot shows a Visual Studio Code editor window with a Python file named `c.py` open. The code implements a program to prompt for a score between 0.0 and 1.0 and print a grade based on the following table:

Score	Grade
>= 0.9	A
>= 0.8	B
>= 0.7	C
>= 0.6	D
< 0.6	F

The code is as follows:

```
1 # Exercise 3
2 try:
3     score = float(input('Enter score: '))
4     if score > 0.0 and score < 1.0:
5         if score >= 0.9:
6             print('A')
7         elif score >= 0.8:
8             print('B')
9         elif score >= 0.7:
10            print('C')
11        elif score >= 0.6:
12            print('D')
13        else:
14            print('F')
15    else:
16        print('Bad score')
17 except:
18     print('Bad score')
```

The terminal output shows the program being run three times with different inputs:

```
PS D:\KTLT-NTTD-060206000031\NTTD-060206000031> & C:/Users/Admin/AppData/Local/Programs/Python/Python313/python.exe d:/KTLT-NTTD-060206000031/NTTD-060206000031/BT03/c.py
Enter score: 0.9
A

PS D:\KTLT-NTTD-060206000031\NTTD-060206000031> & C:/Users/Admin/AppData/Local/Programs/Python/Python313/python.exe d:/KTLT-NTTD-060206000031/NTTD-060206000031/BT03/c.py
Enter score: perfect
Bad score

PS D:\KTLT-NTTD-060206000031\NTTD-060206000031> & C:/Users/Admin/AppData/Local/Programs/Python/Python313/python.exe d:/KTLT-NTTD-060206000031/NTTD-060206000031/BT03/c.py
Enter score: 10
Bad score

PS D:\KTLT-NTTD-060206000031\NTTD-060206000031>
```

