

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ  
MINH  
VIỆN CƠ KHÍ  
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ  
ĐỘNG HÓA  
---000---**



**BÀI TẬP E-learning Môn Kỹ Thuật Lập Trình**

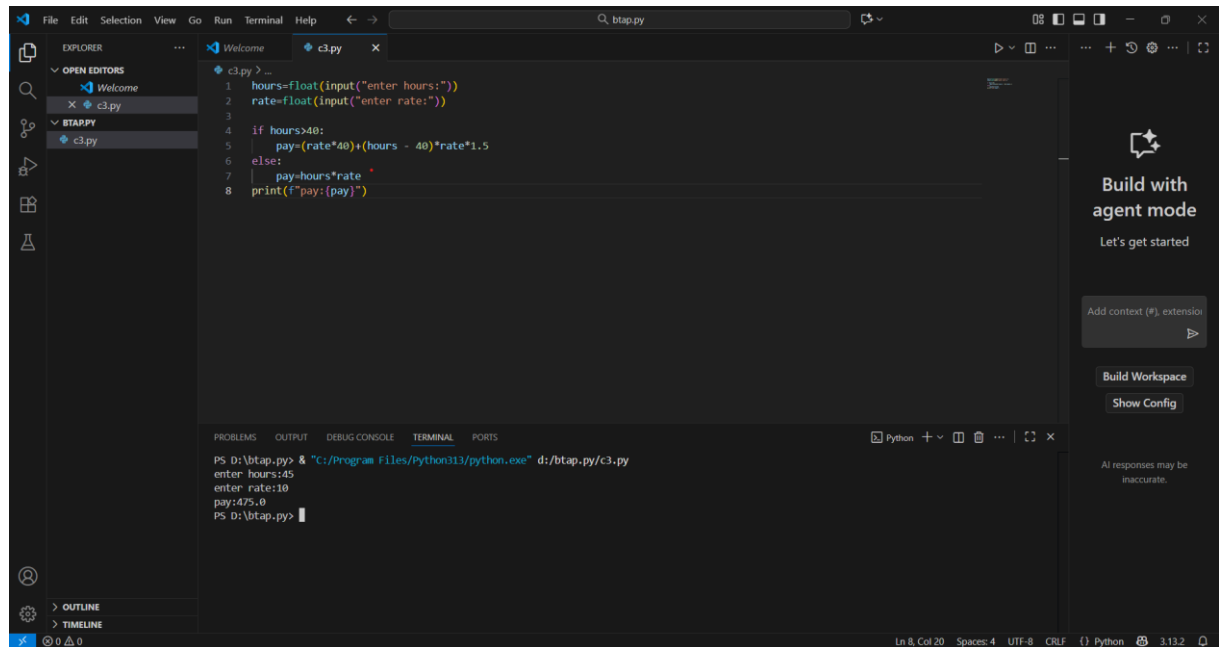
**Thành phố Hồ Chí Minh, Ngày 14 Tháng 10 năm 2025**

---

**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 Visual Studio Code editor window with a Python file named `c3.py` open. The code in the editor is as follows:

```
1 hours=float(input("enter hours:"))
2 rate=float(input("enter rate:"))
3
4 if hours>40:
5     pay=(rate*40)+(hours - 40)*rate*1.5
6 else:
7     pay=hours*rate
8 print(f"pay:{pay}")
```

Below the editor, the TERMINAL panel shows the execution of the script:

```
PS D:\btap.py> & "C:\Program Files\Python313\python.exe" d:\btap.py/c3.py
enter hours:45
enter rate:10
pay:475.0
PS D:\btap.py>
```

The status bar at the bottom indicates the file is at Line 8, Column 20, using UTF-8 encoding with CRLF line endings.

**Exercise 2:** Rewrite your pay program using try and except so that your program handles non-numeric input gracefully by printing a message and exiting the program. The following shows two executions of the program:

Enter Hours: 20

Enter Rate: nine

Error, please enter numeric input

Enter Hours: forty

Error, please enter numeric input

```
1 try:
2     hours=float(input("enter hours:"))
3     rate=float(input("enter rate:"))
4     if hours>40:
5         pay=(hours-40)*rate*1.5 + (40*rate)
6     else:
7         pay=40*rate
8     print(f"pay:{pay}")
9 except:
10    print("error, please enter numeric input")
11
12
13
```

```
PS D:\btap.py> & "C:/Program Files/Python313/python.exe" d:/btap.py/c3.py
enter hours:20
enter rate:nine
error, please enter numeric input
PS D:\btap.py> & "C:/Program Files/Python313/python.exe" d:/btap.py/c3.py
enter hours:forty
error, please enter numeric input
PS D:\btap.py>
```

**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 image shows a Visual Studio Code editor window with a Python file named `c3.py` open. The file contains a `while` loop that prompts the user to enter a score. The code is as follows:

```
1 while True:
2     try:
3         Score=float(input("enter Score:"))
4         if Score<0.0 or Score>1.0:
5             print("ngoi pham vi")
6             continue
7         break
8     except:
9         print("bad Score")
10 if Score >=0.9:
11     grade ='A'
12 elif Score >=0.8:
13     grade ='B'
14 elif Score >=0.7:
15     grade ='C'
16 elif Score >=0.6:
17     grade ='D'
18 else:
19     grade='F'
20
21 print(grade)
```

The terminal output shows the execution of the script. It starts with a `SyntaxError: invalid syntax` message, followed by the command `PS D:\btap.py> & "C:/Program Files/Python313/python.exe" d:/btap.py/c3.py`. The terminal then shows the user entering scores and the program outputting grades:

```
enter Score:0.9A~
bad Score
enter Score:hoan hao
bad Score
enter Score:10,0
bad Score
enter Score:0.75C
bad Score
enter Score:0.5F
bad Score
enter Score:0.9
A
```

The status bar at the bottom indicates the current line and column (Ln 18, Col 6), the encoding (UTF-8), the line ending (CRLF), the language (Python), and the file size (3,132 bytes).