Handson_3_Template

February 26, 2025

1 Functions

To submit this assignment in D2l, post the link to your notebook file on your GitHub account.

1.1 4.1 Even or Odd Checker

Create a program that checks whether a number is even or odd.

1.1.1 Console:

```
Even or Odd Checker

Enter an integer: 33
This is an odd number.
```

1.1.2 Specifications:

- Store the code that gets user input and displays output in the main function.
- Store the code that checks whether the number is even or odd in a separate function.
- Assume that the user will enter a valid integer.

[28]:
```python
def checker(n):
    return n % 2 == 0

def main():
    print("Even or Odd Checker\n")
    number = int(input("Enter an integer:"))

    if checker(number):
        print("This is an even number.")
    else:
        print("This is an odd number.")

main()
```

```
Even or Odd Checker


Enter an integer: 33
This is an odd number.
```

1.2 4.2 - Feet and Meters Converter

Create a program that converts feet to meters and vice versa.

1.2.1 Console

```
Feet and Meters Converter

Conversions Menu:
a.  Feet to Meters
b.  Meters to Feet
Select a conversion (a/b): a

Enter feet: 100
30.48 meters

Would you like to perform another conversion? (y/n): y

Conversions Menu:
a.  Feet to Meters
b.  Meters to Feet
Select a conversion (a/b): b

Enter meters: 100
328.08 feet

Would you like to perform another conversion? (y/n): n

Thanks, bye!
```

1.2.2 Specifications:

- The formula for converting feet to meters is: `feet = meters / 0.3048`
- The formula for converting meters to feet is: `meters = feet * 0.3048`
- Store the code that performs the conversions in functions within a module. For example, store the code that converts feet to meters in a function in a module.
- Store the code that displays the title in its own function, and store the code that displays the menu in its own function, but store the rest of the code that gets input and displays output in a main function.
- Assume the user will enter valid data.
- The program should round results to a maximum of two decimal places.

[26]:
```
def f_to_m(n):
    return round(n * 0.3048, 2)

def m_to_f(n):
```

```
    return round(n / 0.3048, 2)

def converter():
    print("Feet and Meters Converter\n")

def menu():
    print("Conversions Menu:")
    print("a.\tFeet to Meters")
    print("b.\tMeters to Feet")
    selection = input("Select a conversion (a/b):")
    return selection

def main():
    converter()
    while True:
        choice = menu()
        print()
        if (choice == 'a'):
            f = float(input("Enter feet:"))
            meters = f_to_m(f)
            print(f"{meters} meters")

        elif (choice == 'b'):
            m = float(input("Enter meters:"))
            feet = m_to_f(m)
            print(f"{feet} feet")

        Continue = input("\nWould you like to perform another conversion? (y/n):
 ↪")
        print()
        if (Continue == 'y'):
            continue
        elif (Continue == 'n'):
            print("Thanks, bye!")
            break
main()
```

```
Feet and Meters Converter

Conversions Menu:
a.      Feet to Meters
b.      Meters to Feet

Select a conversion (a/b): a


Enter feet: 100

30.48 meters
```

```
Would you like to perform another conversion? (y/n): y


Conversions Menu:
a.      Feet to Meters
b.      Meters to Feet

Select a conversion (a/b): b


Enter meters: 100

328.08 feet


Would you like to perform another conversion? (y/n): n


Thanks, bye!
```

1.3 4.3 - Sales Tax Calculator

Create a program that uses a separate module to calculate sales tax and total after tax.

1.3.1 Console

```
Sales Tax Calculator

ENTER ITEMS (ENTER 0 TO END)
Cost of item: 35.99
Cost of item: 27.50
Cost of item: 19.59
Cost of item: 0
Total:  83.08
Sales tax:  4.98
Total after tax: 88.06

Again? (y/n): y

ENTER ITEMS (ENTER 0 TO END)
Cost of item: 152.50
Cost of item: 59.80
Cost of item: 0
Total:  212.3
Sales tax:  12.74
Total after tax: 225.04

Again? (y/n): n

Thanks, bye!
```

1.3.2 Specifications

- The sales tax rate should be 6% of the total.
- Store the sales tax rate in a module. This module should also contain functions that calculate the sales tax and the total after tax. These functions should round the results to a maximum of two decimal places. Upload the seperate sales tax module to your GitHub repo when submitting.
- Store the code that gets input and displays output in this notebook. Divide this code into functions wherever you think it would make that code easier to read and maintain.
- Assume the user will enter valid data.

[40]:
```
tax = 0.06

def T(Sum):
    Total = round(Sum * tax, 2)
    return Total

def Tat(Sum):
    Sales_tax = T(Sum)
    Total_after_tax = round(Sum + Sales_tax, 2)
    return Total_after_tax

def calculator():
    print("Sales Tax Calculator\n")

def S():
    print("ENTER ITEMS (ENTER 0 TO END)")
    Sum = 0
    while True:
        Coi = float(input("Cost of item: "))
        if Coi == 0:
            break
        Sum += Coi
    return Sum

def Results(Sum):
    sales_tax = T(Sum)
    total_after_tax = Tat(Sum)
    print(f"Total:  {Sum}")
    print(f"Sales tax:\t{sales_tax}")
    print(f"Total after tax: {total_after_tax}\n")

def main():
    calculator()
    while True:
        Sum = S()
        Results(Sum)
```

```
        Continue = input("Again? (y/n): ")
        if Continue == 'y':
            print()
            continue
        elif Continue == 'n':
            print("\nThanks, bye!")
            break
        print()

main()
```

```
Sales Tax Calculator

ENTER ITEMS (ENTER 0 TO END)

Cost of item:  35.99
Cost of item:  27.50
Cost of item:  19.59
Cost of item:  0

Total:  83.08
Sales tax:      4.98
Total after tax: 88.06


Again? (y/n):  y


ENTER ITEMS (ENTER 0 TO END)

Cost of item:  152.50
Cost of item:  59.80
Cost of item:  0

Total:  212.3
Sales tax:      12.74
Total after tax: 225.04


Again? (y/n):  n


Thanks, bye!
```

1.4 4.4 - Prime Number Checker

Create a program that checks whether a number is a prime number and displays the total number of factors if it is not a prime number.

1.4.1 Console

```
Prime Number Checker

Please enter an integer between 1 and 5000: 1
Invalid integer. Please try again.
Please enter an integer between 1 and 5000: 2
2 is a prime number.

Try again? (y/n): y

Please enter an integer between 1 and 5000: 3
3 is a prime number.

Try again? (y/n): y

Please enter an integer between 1 and 5000: 4
4 is NOT a prime number.
It has 3 factors.

Try again? (y/n): y

Please enter an integer between 1 and 5000: 6
6 is NOT a prime number.
It has 4 factors.

Try again? (y/n): n

Bye!
```

1.4.2 Specifications

- A prime number is only divisible by two factors (1 and itself). For example, 7 is a prime number because it is only divisible by 1 and 7.
- If the number is not a prime number, the program should display its number of factors. For example, 6 has four factors (1, 2, 3, and 6).
- Store the code that gets a valid integer for this program in its own function.
- Store the code that calculates the number of factors for a number in its own function.
- Store the rest of the code that gets input and displays output in the main function.

[52]:
```python
def integer():
    while True:
        n = int(input("Please enter an integer between 1 and 5000:"))
        if (n == 1) or (n > 5000):
            print("Invalid integer. Please try again.")
        else:
            return n
```

```python
def calculate(n):
    count = 0
    for i in range(2, n):
        if n % i == 0:
            count += 1
    return count

def main():
    print("Prime Number Checker\n")
    while True:
        n = integer()
        k = calculate(n)

        if (k == 0):
            print(f"{n} is a prime number.\n")
        else:
            print(f"{n} is NOT a prime number.")
            print(f"It has {k} factors.\n")

        Continue = input("Try again? (y/n): ")
        if (Continue == 'y'):
            print()
            continue
        elif (Continue == 'n'):
            print("\nBye!")
            break
main()
```

```
Prime Number Checker

Please enter an integer between 1 and 5000: 1

Invalid integer. Please try again.

Please enter an integer between 1 and 5000: 2

2 is a prime number.

Try again? (y/n):  y


Please enter an integer between 1 and 5000: 3

3 is a prime number.

Try again? (y/n):  y
```

```
Please enter an integer between 1 and 5000: 4

4 is NOT a prime number.
It has 1 factors.


Try again? (y/n):  y


Please enter an integer between 1 and 5000: 6

6 is NOT a prime number.
It has 2 factors.


Try again? (y/n):  n


Bye!
```

[ ]: