Selections and Loops

3.1 Letter Grade Converter

Create a program that converts number grades to letter grades.

Console:

```
Letter Grade Converter

Enter numerical grade: 90
Letter grade: A

Continue? (y/n): y

Enter numerical grade: 88
Letter grade: A

Continue? (y/n): y

Enter numerical grade: 80
Letter grade: B

Continue? (y/n): y

Enter numerical grade: 67
Letter grade: C

Continue? (y/n): y

Enter numerical grade: 59
Letter grade: F

Continue? (y/n): n

Bye!
```

Specifications:

- Assume that the user will enter valid integers for the grades.
- The program should continue only if the user enters “y” or “Y” to continue.
- The grading criteria is as follows:

Letter	Range
A	88-100

Letter	Range
B	80-87
C	67-79
D	60-66
F	<60

In [6]:
```
### CODE HERE ###
print("Colton Kauffman handson 2")
while True:
    grade = int(input("\nPlease enter a valid grade(int): "))

    if grade >= 88 and grade <= 100:
        print("Letter Grade: A")
    elif grade < 88 and grade >= 80:
        print("Letter Grade: B")
    elif grade <= 79 and grade >= 67:
        print("Letter Grade: C")
    elif grade <= 66 and grade >= 60:
        print("Letter Grade: D")
    elif grade < 60:
        print("Letter Grade: F")
    else:
        print("Please enter a valid int")
    newgrade = input(print("Would you like to continue?"))
    if newgrade == "y" or newgrade == "Y":
        grade
    elif newgrade == "n" or newgrade =="N":
        print("Goodbye!")
        break
    else:
        print("Goodbye!")
        break
```

```
Colton Kauffman handson 2
Letter Grade: A
Would you like to continue?
Letter Grade: B
Would you like to continue?
Letter Grade: C
Would you like to continue?
Letter Grade: D
Would you like to continue?
Letter Grade: F
Would you like to continue?
Goodbye!
```

3.2 - Tip Calculator

Create a program that calculates three options for an appropriate tip to leave after a meal at a restaurant.

Console

```
Tip Calculator

Cost of meal: 52.31

15%
Tip amount:     7.85
Total amount: 60.16

20%
Tip amount:     10.46
Total amount: 62.77

25%
Tip amount:     13.08
Total amount: 65.39
```

Specifications:

- The program should calculate and display the cost of tipping at 15%, 20%, or 25%.
- Assume the user will enter valid data.
- The program should round results to a maximum of two decimal places.

In [2]:
```
### CODE HERE ###
def tip_calculator():
    meal_subtotal = float(input("Please enter the total for your meal: "))
    tip_percentages = [15,20,25]

    print("\nTip Calculator\n")
    print(f"Cost of meal: {meal_subtotal:.2f}\n")

    for tip_percentage in tip_percentages:
         tip_amount = meal_subtotal * (tip_percentage/100)
         total = meal_subtotal + tip_amount

         print(f"{tip_percentage}%")
         print(f"Tip amount: {tip_amount:.2f}")
         print(f"Total amount: {total:.2f}\n")


tip_calculator()
```

```
Tip Calculator

Cost of meal: 52.31

15%
Tip amount: 7.85
Total amount: 60.16

20%
Tip amount: 10.46
Total amount: 62.77

25%
Tip amount: 13.08
Total amount: 65.39
```

3.3 - Change Calculator

Create a program that calculates the coins needed to make change for the specified number of cents.

Console

```
Change Calculator

Enter number of cents (0-99): 99

Quarters: 3
Dimes:  2
Nickels: 0
Pennies: 4

Continue? (y/n): y

Enter number of cents (0-99): 55

Quarters: 2
Dimes:  0
Nickels: 1
Pennies: 0

Continue? (y/n): n

Bye!
```

Specifications

- The program should display the minimum number of quarters, dimes, nickels, and pennies that one needs to make up the specified number of cents.
- Assume that the user will enter a valid integer for the number of cents.

- The program should continue only if the user enters “y” or “Y” to continue.

In [1]:
```
### CODE HERE ###
while True:
    try:
        cents = int(input("Enter the number of cents(0-99): "))
        if 0 <= cents <= 99:
            quarters = cents // 25
            cents %= 25
            dimes = cents // 10
            cents %= 10
            nickels = cents // 5
            pennies = cents % 5

            print(f"\nQuarters: {quarters}")
            print(f"Dimes: {dimes}")
            print(f"Nickels: {nickels}")
            print(f"Pennies: {pennies}\n")

            choice = input("Continue? (y/n): ").lower()
            if choice != 'y':
                print("Bye!")
                break
        else:
            print("please enter number 0-99!")
    except error:
        print("Enter a valid int please!!")
```

```
Quarters: 3
Dimes: 2
Nickels: 0
Pennies: 4

Quarters: 2
Dimes: 0
Nickels: 1
Pennies: 0

Bye!
```

3.4 - Table of Powers

Create a program that displays a table of squares and cubes for the specified range of numbers.

Console

```
Table of Powers

Start number: 90
Stop number: 100
```

```
Number  Squared Cubed
======  ======= =====
90      8100    729000
91      8281    753571
92      8464    778688
93      8649    804357
94      8836    830584
95      9025    857375
96      9216    884736
97      9409    912673
98      9604    941192
99      9801    970299
100     10000   1000000
```

Specifications

- The formulas for calculating squares and cubes are:

  `square = x ** 2`

  `cube = x ** 3`

- Use tabs to align the columns.
- Assume that the user will enter valid integers.
- Make sure the user enters a start integer that's less than the stop integer. If the user enters a start integer that's greater than the stop integer, display an error message and give the user a chance to enter the integers again.

In [4]:
```
### CODE HERE ###
start = int(input("Start number: "))
stop = int(input("Stop number: "))

if start < stop:
    print("\nTable of Powers\n")
    print("Number\tSquared\tCubed")
    print("-------\t-------\t-----")

    for number in range(start, stop + 1):
        square = number ** 2
        cube = number ** 3
        print(f"{number}\t{square}\t{cube}")
else:
    print("start must be less than stop!")
```

```
Table of Powers

Number  Squared Cubed
------- ------- -----
90      8100    729000
91      8281    753571
92      8464    778688
93      8649    804357
94      8836    830584
95      9025    857375
96      9216    884736
97      9409    912673
98      9604    941192
99      9801    970299
100     10000   1000000
```

In [ ]: