Solution for Tutorial 1 (Week 1 + 2)

OBJECTIVES

- Able to use print function for output
- Able to use string addition/concatenation
- Able to use string format
- Able to translate number into string
- Able to use escape sequences
- Able to use if-else statement
- Able to use for loop statement
- Able to use while loop statement

String format, input output

Given the following code:

```
product_code = "377B"
product_name = "Beef Liquid Stock"
product_size = "250mL"
product_price = 2.15
```

SECTION 1 – Run and understand

1) What is the output of the following statement?

```
print("product_code + product_name + product_size")
```

Copy the python code, run it, and check your answer.

EXAMPLE SOLUTION: product_code + product_name + product_size

2) What is the output of the following statement?

```
print(product_code + " product_name " + product_size)
```

Copy the python code, run it, and check your answer.

EXAMPLE SOLUTION: 377B product_name 250mL

3) What is the output of the following statement?

```
print(product_code + ", " + product_name + ", + product_size")
```

Copy the python code, run it, and check your answer.

EXAMPLE SOLUTION: 377B, Beef Liquid Stock, + product_size

4) What is the output of the following statement?

```
print(product_code + ", " + product_name + ", " + product_size)
```

Copy the python code, run it, and check your answer.

EXAMPLE SOLUTION: 377B, Beef Liquid Stock, 250mL

SECTION 2 Exercises – string format

5) Write one print statement using the above variables and string addition so that it produces the following exact output:

```
377B: Beef Liquid Stock, 250mL
```

EXAMPLE SOLUTION:

```
print(product_code + ": " + product_name + ", " + product_size)
```

6) Write one print statement using the above variables and string addition so that it produces the following exact output:

```
"Beef Liquid Stock", 250mL
```

EXAMPLE SOLUTION: `print('"' + product_name + '", ' + product_size)`

7) Write one print statement using the above variables and string addition so that it produces the following exact output:

```
Beef Liquid Stock, 250mL $2.15
```

EXAMPLE SOLUTION:

```
print(product_name + ", " + product_size + ", $" + str(product_price))
```

Write one print statement using the above variables and string format so that it produces the following exact output:

```
377B: Beef Liquid Stock, 250mL
```

EXAMPLE SOLUTION:

```
print("{0}: {1}, {2}".format(product_code, product_name, product_size))
```

8) Write one print statement using the above variables and string format so that it produces the following exact output:

```
"Beef Liquid Stock", 250mL
```

EXAMPLE SOLUTION: `print('"{0}", {1}'.format(product_name, product_size))`

9) Write one print statement using the above variables and string format so that it produces the following exact output:

```
Beef Liquid Stock, 250mL, $2.15
```

EXAMPLE SOLUTION: `print("{0}, {1}, ${2}".format(product_name, product_size, product_price))`

10) Using string format to write a program that produces the following output

```
President           Secret Service Code Name        In Office
Donald Trump                 "Mogul"                2017-present
Barack Obama                "Renegade"              2009-2017
George W. Bush              "Tumbler"               2001-2009
Bill Clinton                 "Eagle"                1993-2001
```

EXAMPLE SOLUTION:

```
print("{0:<25}{1:^40}{2:<25}".format("President", "Secret Service Code Name" , "In Office"))
print("{0:<25}{1:^40}{2:<25}".format("Donald Trump", "\"Mogul\"", "2017-present"))
print("{0:<25}{1:^40}{2:<25}".format("Barack Obama", "\"Renegade\"", "2009-2017"))
print("{0:<25}{1:^40}{2:<25}".format("George W. Bush", "\"Tumbler\"", "2001-2009"))
print("{0:<25}{1:^40}{2:<25}".format("Bill Clinton", "\"Eagle\"", "1993-2001"))
```

11) Using string format and escape sequence to write a program that produces the following output

```
Alkali metals:

Element         Symbol          Atomic number          Atomic weight
Lithium         Li                      3                      6.940
Sodium          Na                     11                     22.990
Potassium       K                      19                     39.098
Rubidium        Rb                     37                     85.468
Caesium         Cs                     55                    132.905
Francium        Fr                     87                    223.000
```

EXAMPLE SOLUTION:

```
print("Alkali metals:")
print()
# alignment: left, left, center, right (3 decimal point)
print("{0:<20}{1:<15}{2:^15}{3:>25}".format("Element", "Symbol", "Atomic number", "Atomic weight"))
print("{0:<20}{1:<15}{2:^15}{3:>25.3f}".format("Lithium", "Li", 3, 6.940))
print("{0:<20}{1:<15}{2:^15}{3:>25.3f}".format("Sodium", "Na", 11, 22.990))
print("{0:<20}{1:<15}{2:^15}{3:>25.3f}".format("Potassium", "K", 19, 39.098))
print("{0:<20}{1:<15}{2:^15}{3:>25.3f}".format("Rubidium", "Rb", 37, 85.468))
print("{0:<20}{1:<15}{2:^15}{3:>25.3f}".format("Caesium", "Cs", 55, 132.905))
print("{0:<20}{1:<15}{2:^15}{3:>25.3f}".format("Francium", "Fr", 87, 223.000))
```

12) Using string format to write a program that produces the following output

```
1 x  1 =   1
2 x  2 =   4
3 x  3 =   9
4 x  4 =  16
5 x  5 =  25
```

```
 6 x  6 =  36
 7 x  7 =  49
 8 x  8 =  64
 9 x  9 =  81
10 x 10 = 100
```

EXAMPLE SOLUTION:

```
print("{0:>2} x {1:>2} = {2:>3}".format(1, 1, 1*1))
print("{0:>2} x {1:>2} = {2:>3}".format(2, 2, 2*2))
print("{0:>2} x {1:>2} = {2:>3}".format(3, 3, 3*3))
print("{0:>2} x {1:>2} = {2:>3}".format(4, 4, 4*4))
print("{0:>2} x {1:>2} = {2:>3}".format(5, 5, 5*5))
print("{0:>2} x {1:>2} = {2:>3}".format(6, 6, 6*6))
print("{0:>2} x {1:>2} = {2:>3}".format(7, 7, 7*7))
print("{0:>2} x {1:>2} = {2:>3}".format(8, 8, 8*8))
print("{0:>2} x {1:>2} = {2:>3}".format(9, 9, 9*9))
print("{0:>2} x {1:>2} = {2:>3}".format(10, 10, 10*10))
```

IF-ELSE STATEMENT

1. Write a program to calculate the cost based on the following pricing.

Number of items	**Cost**
1-50	$3 per item Postage: $10
More than 50	$2 per item Postage: free

The program should display a receipt as in the following examples:

Example 1:

```
Enter the number of items: 10


Receipt:
10 items x $3 =
$30 Postage: $10
Total: $40
```

Example 2:

```
Enter the number of items: 100


Receipt:
100 items x $2 =
$200 Postage: $0
Total: $200
```

EXAMPLE SOLUTION:

```
# ask user to enter number of items
user_input = input("Enter the number of items: ")
item_count = int(user_input)

print()

# determine cost per item and postage based on the number of items
if (item_count <= 50):
  cost_per_item = 3
  postage = 10
else:
```

```
  cost_per_item = 2
  postage = 0

# calculate item cost
item_cost = item_count * cost_per_item

# calculate total cost
total_cost = item_cost + postage

# display receipt
print("Receipt:")
print("{0} items x ${1} = ${2}".format(item_count, cost_per_item, item_cost))
print("Postage: ${0}".format(postage))
print("Total: ${0}".format(total_cost))
```

2. Write a program to calculate the cost based on the following pricing.

Number of items	**Cost**
1-50	$3 per item Postage: Standard post: $10 Registered post: $15 Express post: $20
More than 50	$2 per item Postage: Standard post: free Registered post: $10 Express post: $17

The program should display a receipt as in the following examples:
Example 1:

```
Enter the number of items: 10
Enter shipping method (s/r/e): r

Receipt:
10 items x $3 =
$30 Registered
post: $15 Total:
$45
```

Example 2:

```
Enter the number of items: 100

Enter shipping method (s/r/e): s


Receipt:
100 items x $2 =
$200 Standard post:
$0 Total: $200
```

EXAMPLE SOLUTION:

```
# ask user to enter number of items
user_input = input("Enter the number of items: ")
item_count = int(user_input)

# ask user to enter shipping method
shipping_option = input("Enter shipping method (s/r/e): ")

print()

# determine cost per item based on the number of items
if (item_count <= 50):
#{
  cost_per_item = 3
#}
else:
#{
  cost_per_item = 2
#}
# determine postage and postage description
if (shipping_option == "s"):
#{
  postage_desc = "Standard post"

  if (item_count <= 50):
  #{
    postage = 10
  #}
  else:
  #{
    postage = 0
  #}
#}
elif (shipping_option == "r"):
#{
  postage_desc = "Registered post"

  if (item_count <= 50):
  #{
    postage = 15
  #}
  else:
  #{
    postage = 10
  #}
#}
```

```
elif (shipping_option == "e"):
#{
  postage_desc = "Express post"

  if (item_count <= 50):
  #{
    postage = 20
  #}
  else:
  #{
    postage = 17
  #}
#}
```

3. Write a program to ask the user to enter four integers and then display the minimum and maximum number.

The program should work as in the following examples:

```
Enter the first integer: 10
Enter the second integer: 2
Enter the third integer: 15
Enter the fourth integer: 9

The minimum number is 2 and the maximum number is 15.
```

Here is another example:

```
Enter the first integer: 5
Enter the second integer: 5
Enter the third integer: 5
Enter the fourth integer: 5

The minimum number is 5 and the maximum number is 5.
```

EXAMPLE SOLUTION:

```
# ask user to enter 4 integers
user_input = input("Enter the first integer: ")
number1 = int(user_input)

user_input = input("Enter the second integer: ")
number2 = int(user_input)

user_input = input("Enter the third integer: ")
number3 = int(user_input)

user_input = input("Enter the fourth integer: ")
number4 = int(user_input)

print()

# determine the max
number_max = number1

# if max < number2 then reset it
if (number_max < number2):
#{
   number_max = number2
#}

# if max < number3 then reset it
if (number_max < number3):
#{
   number_max = number3
#}

# if max < number4 then reset it
if (number_max < number4):
#{
  number_max = number4
#}


# determine the min
number_min = number1

# if min > number2 then reset it
if (number_min > number2):
#{
  number_min = number2
#}

# if min < number3 then reset it
if (number_min > number3):
#{
  number_min = number3
#}

# if min > number4 then reset it
if (number_min > number4):
#{
  number_min = number4
```

```
#}

# display max, min result
print("The minimum number is {0} and the maximum number is
{1}.".format(number_min, number_max))
```

FOR LOOP

1. Write a program to display equations using `for` loop statement and string format

```
 1 +  1 =  2
 2 +  2 =  4
 3 +  3 =  6
 4 +  4 =  8
 5 +  5 = 10
 6 +  6 = 12
 7 +  7 = 14
 8 +  8 = 16
 9 +  9 = 18
10 + 10 = 20
```

EXAMPLE SOLUTION:

```
for i in range(1, 11):
#{
  print("{0:>2} + {1:>2} = {2:>2}".format(i, i, i+i))
#}
```

2. Write a program to display the following sequence of numbers using `for` loop statement.

```
1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10.

1 : 2 : 3 : 4 : 5 : 6 : 7 : 8 : 9 : 10
```

EXAMPLE SOLUTION:

```
# 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10.
for i in range(1, 11):
#{
  # determine trailing
  if (i < 10):
  #{
    trailing = " - "
  #}
  else:
  #{
    trailing = "."
  #}
```

```
# 1 : 2 : 3 : 4 : 5 : 6 : 7 : 8 : 9 : 10
for i in range(1, 11):
#{
  # determine trailing
  if (i < 10):
  #{
    trailing = " : "
  #}
  else:
  #{
    trailing = ""
  #}

  # display number
  print(i, end="")

  # display trailing
  print(trailing, end="")
#}

  # display number
  print(i, end="")

  # display trailing
  print(trailing, end="")
#}
```

3. Write a program to display the following sequence of numbers using `for` loop statement.

12, 14, 16, 18, 20.

1.2 * 1.4 * 1.6 * 1.8 * 2.0.

1; 3; 5; 7; 9.

EXAMPLE SOLUTION:

```
# 12, 14, 16, 18, 20.
# five numbers altogether
for i in range(1, 6):
#{
  # determine the number
  number = 10 + i * 2

  # determine the trailing
  if (i < 5):
  #{
    trailing = ", "
  #}
  else:
  #{
    trailing = "."
```

```
  #}

# display number
  print(number, end=trailing)
#}

# 1.2 *1. 4 *1. 6 *1.8 * 2.0.
# five numbers altogether
for i in range(1, 6):
#{
  # determine the number
  number = 1+ i * 2

  # determine the trailing
  if (i < 5):
  #{
    trailing = " * "
  #}
  else:
  #{
    trailing = "."
  #}

  # display number
  print(number, end="")

  # display trailing
  print(trailing, end="")
#}

# 1; 3; 5; 7; 9.
# five numbers altogether
for i in range(1, 6):
#{
  # determine the number
  number = i * 2 - 1

  # determine the trailing
  if (i < 5):
  #{
    trailing = "; "
  #}
  else:
  #{
    trailing = "."
  #}

  # display number
  print(number, end="")

  # display trailing
  print(trailing, end="")
#}
```

4. Write a program to display the following sequence of numbers using `for` loop statement.

```
0
02
024
0246
02468
0246810
024681012
02468101214
0246810121416
024681012141618
02468101214161820
```

EXAMPLE SOLUTION:

```
# print 11 lines
for line in range(0, 11):
#{
  # on each line, print numbers from 1 to line number
  for number in range(1, line+1):
  #{
    print(number*2, end="")
  #}

  # print end of line
  print()
#}
```

WHILE LOOP

1. Write a program to display equations using `while` loop statement and string format

```
 2 +  2 =  4
 4 +  4 =  8
 6 +  6 = 12
 8 +  8 = 16
10 + 10 = 20
```

EXAMPLE SOLUTION:

```
# display equations for even numbers from 2-10
even = 2
while (even <= 10):
#{
  print("{0:>2} + {1:>2} = {2:>2}".format(even, even, even+even))
  even = even + 2
#}
```

2. Write a program to display the following sequence of numbers using `while` loop statement.

```
 1.1 : 2.2 : 3.3 : 4.4 : 5.5 : 6.6 : 7.7 : 8.8 : 9.9 : 11.0
```

EXAMPLE SOLUTION:

```
# ten numbers from 1 to 10
number = 1

while (number <= 10):
#{
  # determine trailing
  if (number < 10):
  #{
    trailing = " : "
  #}
  else:
  #{
    trailing = ""
  #}

  # display number
  print(number*1.1, end="")

  # display trailing
  print(trailing, end="")

  # update number
  number = number + 1
#}
```

```
10 * 8 * 7 * 6 * 5.
i = 0
while (i < 6):

  if i == 5:
    trailing = '.'
  else:
    trailing = ' * '

  print(10-i, end=trailing)
  i += 1
  if i==1:
    i += 1
```

```
1; 3; 5; 7; 9.

# odd numbers from 1 to 9
odd = 1

while (odd <= 9):
#{
  # determine trailing
  if (odd < 9):
  #{
    trailing = "; "
  #}
  else:
```

```
  #{
    trailing = "."
  #}

  # display odd number
  print(odd, end="")

  # display trailing
  print(trailing, end="")

  # update odd number
  odd = odd + 2
#}
```

3. Write a program that uses the **`while` loop,** asks the user to enter integer numbers and then displays the summary information. The user has to enter q to quit the program. The program should work as follows:

```
Enter an integer or q to quit: 5
Enter an integer or q to quit: 10
Enter an integer or q to quit: -1
Enter an integer or q to quit: 3
Enter an integer or q to quit: 0
Enter an integer or q to quit: -5
Enter an integer or q to quit: q


Summary information:
You have entered 6 integers. The sum of these numbers is 12.
There are 2 even numbers.
There are 4 odd numbers. There are 3 positive numbers. There
are 2 negative numbers.
```

EXAMPLE SOLUTION:

```
# initialise
number_count = 0
number_sum = 0
even_count = 0
odd_count = 0
positive_count = 0
negative_count = 0

# run for ever until user want to quit
while True:
#{
  # ask user enter integer or q for quit
  user_input = input("Enter an integer or q to quit: ")

  # check if user want to quit
  if (user_input == "q"):
  #{
    break
  #}
```

```
  # user does not want to quit
  number = int(user_input)

  # update number count
  number_count = number_count + 1

  # update sum
  number_sum = number_sum + number

  # update even and odd counts
  if (number%2 == 0):
  #{
    even_count = even_count + 1
  #}
  else:
  #{
    odd_count = odd_count + 1
  #}

  # update positive count
  if (number > 0):
  #{
    positive_count = positive_count  + 1
  #}

  # update negative count
  if (number < 0):
  #{
    negative_count = negative_count  + 1
  #}

#}
# display summary
print()
print("Summary information:")
print("You have entered {0} integers.".format(number_count))
print("The sum of these numbers is {0}.".format(number_sum))
print("There are {0} even numbers.".format(even_count))
print("There are {0} odd numbers.".format(odd_count))
print("There are {0} positive numbers.".format(positive_count))
print("There are {0} negative numbers.".format(negative_count))
```