Module 7: Python Functions

- Function is group of related statements that perform a specific task.
- Functions make the code organized, reusable and manageable.
- Type of Functions
  - Built-in functions: functions that are built into Python.
  - User-Defined functions: functions defined by users.

return statement:

- We can return a result from a function using return statement.
- Functions in python can return multiple values also.

Parameter Types:

- **Formal Parameters:**
  - These are the parameters that appear in the function definition.
  - They are useful for receiving the value from outside function.
- **Actual Parameter:**
  - These are the parameters that appear in the function call.
- Note: Parameters are also called arguments.

Parameter Passing Techniques:

- **Positional Parameters:**
  - In this the parameters passed to a function correspond to their positions. The number and positions of Actual and Formal parameters should match.
- **Keyword Parameters:**
  - In this the parameters passed to a function correspond to their names. The positions of the arguments can be changed.
- Note: Having positional argument after keyword arguments will result into errors.

Default Parameters:

- With formal parameters we can specify a default value which would be used if that parameter is not passed.
- Note:
  - Any number of arguments in a function can have default values.
  - All arguments to the right of default arguments must have default values.
  - Non-default arguments cannot follow default arguments.

Module 7: Python Functions Test

Q1) What is the result?

```
define welcome():
        print('Welcome All')
welcome()
```

Options:

A. Welcome All
B. All
C. Welcome
D. Error

Solution:

Q2) You are developing a Python application for an online game.
You need to create a function that meets the following criteria:

- The function is named update_score
- The function receives the current score and a value
- The function adds the value to the current score
- The function returns the new score

How should you complete the code?

```
Option1   Option2
        current += value
        Option3
```

Options:

Option1:

A. update_score
B. def update_score
C. return update_score

Option2:

A. (current,value):
B. ():
C.(current,value)
D. ()

Option3:

A. pass current
B. return current
C. return
D. pass

Solution:

Q3) The ABC company is creating a program that allows customers to log the number of miles biked. The program will send messages based on how many miles the customer logs. You create the following code.

```
01.
02.    name = input("what is your name ? ")
03.    return name
04.
05.    calories = miles * calories_per_mile
06.    return calories
07. distance = int(input('how many miles did you bike this week? '))
08. burn_rate = 50
09. biker = get_name()
10. calories_burned = calc_calories(distance, burn_rate)
11.  print(biker, 'you burned about ', calories_burned, 'calories')
```

Which code segments should you use for line 01 and line 04. Select two

Options:

A. 01. def get_name():

B. 01. def get_name(name):

C. 04. def calc_calories():

D. 04. def calc_calories(miles, calories_per_mile):

Solution:

Q4) What is the result?

```
def sum(a, b):
    c = a + b
    return c
    print(c)
res = sum(10, 20)
print(res)
```

Options:

A. 30 30	B. 0 30
C. Error	D. 30

Solution:

Q5) What is the result?

```
def welcome(msg="Hello", name):
    print(msg, name)
    welcome(name='Amit', msg='Hi')
```

Options:

A. Hello Amit
B. Hi Amit
C. Amit
D. Error

Solution:

Q6) What is the result?

```
def msg(msg, times = 2):
    print(msg * times)
    msg(times = 4, 'hello')
```

Options:

A. hellohello
B. hellohellohellohello
C. Error
D. hello4

Solution:

Q7) What is the result?

```
def fun(a, b=5, c = 10):
    print(a, b, c)
fun(c=20, a = 5)
```

Options:

A. 5 10
B. 5 5 10
C. 5 5 20
D. Error

Solution:

Q8) You are writing a function that increments the player score in a game. The function has the following requirements:

- If no value is specified for points, then points start at one
- If bonus **IS True,** then points must be doubled

You write the following code. Line numbers are included for reference only.

```
01 def increment_score(score, bonus, points):
02     if bonus == True:
03         points = points * 2
04         score = score + points
05         return score
06 points = 5
07 score = 10
08 new_score = increment_score(score, True, points)
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

Question1:

To meet the requirements, line 01 must be changed to the following:
def increment_score(score, bonus, points = 1):
A. yes B. No

Question2:

Once any parameter is defined with a default value, any parameters to the right must also be defined with default values.
A. yes B. No

Question3:

If the function is called with only two parameters, the value of the third parameter will be None.
A. yes B. No

Solution: