1. What is the output of the following code? **1 / 1 point**

```
x="Go"

if(x=="Go"):

  print('Go ')

else:

  print('Stop')

print('Mike')
```

- (●) Go Mike
- ( ) Mike
- ( ) Stop Mike

Correct

2. What is the result of the following lines of code? **1 / 1 point**

```
x=1
x>5
```

---

3. What is the output of the following few lines of code? **1 / 1 point**

```
x=0
while(x<2):
    print(x)
    x=x+1
```

- (●) 0
  1
- ( ) 0
  1
  2
- ( ) 0
  1
  3
  4

Correct
Correct

4. What is the result of running the following lines of code ?

1 / 1 point

```
class Points(object):
  def __init__(self,x,y):

    self.x=x
    self.y=y

  def print_point(self):

    print('x=',self.x,' y=',self.y)

p1=Points("A","B")
p1.print_point()
```

- ○ x= A
- ○ y= B
- ◉ x= A y= B

Correct
correct

5. What is the output of the following few lines of code?

1 / 1 point

```
for i,x in enumerate(['A','B','C']):
    print(i+1,x)
```

- ◉ 1 A
  2 B
  3 C
- ○ 0 A
  1 B
  2 C
- ○ 0 AA
  1 BB
  2 CC

Correct
Correct

6. What is the result of running the following lines of code ?

1 / 1 point

```
class Points(object):

  def __init__(self,x,y):

    self.x=x
    self.y=y

  def print_point(self):

    print('x=',self.x,' y=',self.y)

p2=Points(1,2)

p2.x='A'

p2.print_point()
```

- ○ x= 1 y=2
- ◉ x= A y=2
- ○ x=A, y=B

✓ **Correct**
correct

7. Consider the function step, when will the function return a value of 1? **1 / 1 point**

```
def step(x):
    if x>0:
        y=1
    else:
        y=0
    return y
```

- (●) if x is larger than 0
- ( ) if x is equal to or less then zero
- ( ) if x is less than zero

Correct
correct, the value of y is 1 only if x is larger than 0

8. What is the output of the following lines of code? **1 / 1 point**

```
a=1

def do(x):
    a=100
    return(x+a)

print(do(1))

```

- ( ) 2
- (●) 101
- ( ) 102

Correct
Correct, the value of a=100 exists in the local scope of the function. Therefore the value of a=1 in the global scope is not used.

9. Write a function name **add** that takes two parameter **a** and **b,** then return the output of **a + b** (Do not use any other variable! You do not need to run it. Only write the code about how you define it.) **1 / 1 point**

```
1  def add(a, b):
2      return a + b
```

Run

Reset

Correct

Good job!

10. Why is it best practice to have multiple except statements with each type of error labeled correctly? **1 / 1 point**

- ○ Ensure the error is caught so the program will terminate
- ◉ In order to know what type of error was thrown and the location within the program
- ○ To skip over certain blocks of code during execution
- ○ It is not necessary to label errors

Correct