1. Fill in the blank: Automation is _____. **1 / 1 point**

- (●) the use of technology to reduce human and manual effort to perform common and repetitive tasks
- ( ) the combination of technology and manual effort to complete a task
- ( ) the replacement of existing technology
- ( ) the use of human and manual effort to reduce technological power consumption

✓ **Correct**

2. The purpose of the following code is to print an `"Attempting connection"` message while the value of the count variable is less than `10`. The value of count should increase by `1` with each iteration of the loop. What is wrong with the code? Select all that apply. **1 / 1 point**

```
count = 1

while count < 10:

print("Attempting connection")

count = count + 1
```

- [x] The line with `print("Attempting connection")` is not indented.

✓ **Correct**

- [x] The line with `count = count + 1` is not indented.

✓ **Correct**

- [ ] The line with `count = 1` is not indented
- [ ] The line with `while count < 10:` is not indented.

3. Which of these are string data? Select all that apply. **1 / 1 point**

- [x] `"user1"`

✓ **Correct**

- [x] `"100"`

✓ **Correct**

- [ ] `100`
- [ ] `[100, 200, 300]`

4. What are possible values for the Boolean data type? Select all that apply. **1 / 1 point**

- [ ] `!=`
- [x] `False`

✓ **Correct**

- [x] `True`

✓ **Correct**

- [ ] `>`

5. What are the variables in the following code? Select all that apply. **1 / 1 point**

```
username = "kcarter"

attempts = 5

print(username)

print(attempts)

print("locked")
```

- [ ] `"kcarter"`
- [x] `attempts`

✓ **Correct**

- [ ] `"locked"`
- [x] `username`

✓ **Correct**

6. What will this code do when you run it? **1 / 1 point**

```
var2 = ["a","b","c"]

var2_type = type(var2)

print(var2_type)
```

- ○ Print the string `"var2_type"` to the screen
- ◉ Indicate that `var2` contains list data
- ○ Change the data type of `var2`
- ○ Output the characters `"a"`, `"b"`, and `"c"` to the screen

✓ **Correct**

7. You are implementing security measures on a server. If a user has more than `3` failed login attempts, the program should print `"locked out"`. The number of failed login attempts is stored in a variable called `failed_attempts`. Which conditional statement has the correct syntax needed to do this? **1 / 1 point**

- ○
```
if failed_attempts < 3
    print("locked out")
```
- ◉
```
if failed_attempts > 3:
    print("locked out")
```
- ○
```
if failed_attempts >= 3
    print("locked out")
```
- ○
```
if failed_attempts <= 3:
    print("locked out")
```

✓ **Correct**

8. You wrote the following code: **1 / 1 point**

```
if attempts >= 5:
    print("locked")
else:
    print("try again")
```

If the value in the `attempts` variable is `3`, what will Python do?

- ○ First output the message `"try again"` and then output the message `"locked"`
- ◉ Output the message `"try again"`
- ○ First output the message `"locked"` and then output the message `"try again"`
- ○ Output the message `"locked"`

✓ **Correct**

9. What iterative statement can you use if you want to print `"Security alert"` five times? **1 / 1 point**

- ○
```
for i in [0, 5]:
    print("Security alert")
```
- ○
```
for i in range(6):
    print("Security alert")
```
- ◉
```
for i in range(5):
    print("Security alert")
```
- ○
```
for i in range(1,5):
    print("Security alert")
```

✓ **Correct**

10. You want to print all even numbers between `0` and `10` (in other words, `0`, `2`, `4`, `6`, `8`, and `10`). What should your next line of code be? **0 / 1 point**

```
count = 0
while count <= 10:
    print(count)
```

- ○ `count = count + 2`
- ◉ `count = count + 1`
- ○ `count = 1`
- ○ `if count < 10:`

✗ **Incorrect**

Please review the video on while loops ↗