**CSI 160 Python Programming**

Activity 3: Loops

True/False

|  |  |  |
| --- | --- | --- |
| **No** | **Statement** | **T/F** |
| 1 | The **break** statement can be used to exit a loop prematurely. |  |
| 2 | It is possible to use the **break** statement to exit multiple nested loops at once. |  |
| 3 | The **continue** statement is used to skip the remaining code within the current iteration of a loop and move to the next iteration. |  |
| 4 | The sequence generated from **range(n)**, where **n** is an integer, starts at 1 and goes up to (but does not include) **n**. |  |
| 5 | Nested loops require using both for and while loops. |  |
| 6 | Infinite loops occur when the condition controlling the loop never becomes false or when there is no exit statement within the loop body. |  |
| 7 | The body of a while loop in Python is always guaranteed to execute at least once. |  |
| 8 | A **while True** loop will run indefinitely until a **break** statement is encountered or an error occurs within the body. |  |
| 9 | The loop condition in a while loop is evaluated before each iteration. |  |
| 10 | The loop control variable in loops like **for x in range(2)** can be accessed outside the loop. |  |

What is the output of the following code?

|  |
| --- |
| num = 4  result = 1  i = 1  while (i <= num):  result \*= i  i += 1  print("The result is:", result) |
| start\_num = 6  while (start\_num > 0):  print(start\_num)  start\_num -= 1 |
| limit = 10  result = 0  for i in range(2, limit + 1, 2):  result += i  print("The result is:", result) |
| num = 10  while (num > 10):  if ((num % 2) == 0):  print("Hello from if!")  else:  print("Hello from else!") |
| count = 1  while (count <= 10):  if (count % 3 == 0):  count += 2  print(count, end=" ")  count += 1 |
| value\_x = 0  while value\_x < 5:  value\_x += 1  if value\_x == 3:  continue  print(value\_x) |
| passwords= ["1234", "admin", "guest", "letmein"]  for pwd in passwords: # **pwd** is a variable  print("Checking:", pwd)  if pwd == "guest":  print("Password found!")  break |
| scores = [89, -1, 76, 92, -5, 83]  for s in scores:  if s < 0:  continue  print("Valid score:", s) |
| for i in range(4):  for j in range(i):  print(i + j, end=" ") |
| for i in range(1, 4):  for j in range(i):  print("\*", end="")  print() |