**Tutorial 04**

**Simple Questions to Try**

1. **Experiment with for loops**: Write simple for loops to print numbers from 1 to 5.

2. **Play with while loops**: Create a while loop that counts down from 5 to 1.

3. **Explore endless while loops**: Write an endless loop and safely exit it with a break statement.

4. **Use continue**: Implement a while loop that skips even numbers using the continue statement.

**Guided Practice Questions**

1. **Sum of Numbers**
   - **Question**: Write a program that uses a **for** loop to sum numbers from 1 to 10.

   - **Pseudo Code**:

     ```
     Initialize sum to 0
     For each number from 1 to 10
       Add the number to sum
     Print sum
     ```

2. **Guessing Game**
   - **Question**: Implement a guessing game using a **while** loop, where the user has to guess a randomly selected number between 1 and 10.

   - **Pseudo Code**:

     ```
     Generate a random number between 1 and 10
     While user's guess is not equal to the generated number
       Ask user for their guess
       If guess is correct
         Print a congratulatory message
         Break the loop
       Else
         Provide a hint and continue asking
     ```

3. **Number Skipper**
   - **Question**: Use a **for** loop with **continue** to print all odd numbers between 1 and 20.

   - **Pseudo Code**:

     ```
     For each number from 1 to 20
       If the number is even
         Continue to the next iteration
     ```

Else
Print the number

4. **Endless Input**
   - **Question**: Create an endless **while** loop that takes user inputs until the user types "exit".

   - **Pseudo Code**:

     While True
     Ask user for input
     If input is "exit"
     Break the loop
     Else
     Print the input or perform an action

**Unguided Practice Questions**

1. **Reverse Counter**
   - **Question**: Write a program that counts backwards from 10 to 1 using a **while** loop.

   - **Pseudo Code**:

     Set counter to 10
     While counter is greater than 0
       Print counter
       Decrease counter by 1

2. **Multiplication Table**
   - **Question**: Generate a multiplication table for a number provided by the user up to 10.

   - **Pseudo Code**:

     Ask user for a number
     For each number from 1 to 10
       Calculate multiplication of the user's number with the current number in the loop
       Print the result

3. **Number Accumulator**
   - **Question**: Use a **while** loop to accumulate the sum of user-entered numbers until the user enters 0.
   - **Pseudo Code**:

     Initialize sum to 0
     While True
       Ask user for a number
       If the number is 0
         Break the loop
       Add the number to sum
     Print sum

4. **Pattern Printer**
   - **Question**: Utilize nested **for** loops to print a pyramid pattern of stars (*) for a given number of levels.

   - **Pseudo Code**:

     Ask user for the number of levels
     For each level from 1 to the user's number
       Print spaces decreasingly as per level number
       Print stars (*) increasingly as per level number
       Move to the next line

**MCQs**

1. What does the following code print?

```
for i in range(5):
    print(i)
```

- A) 0 1 2 3 4
- B) 1 2 3 4 5
- C) 0 1 2 3
- D) Error

2. How many times does the following loop execute?

```
i = 5
while i > 0:
    i -= 1
```

- A) 4
- B) 5
- C) 6
- D) Infinite loop

3. What is the output of the following code?

```
for i in range(1, 10, 2):
    print(i)
```

- A) 1 3 5 7 9
- B) 2 4 6 8 10
- C) 1 4 7
- D) 2 5 8

4. Consider the following code snippet. What will it print?

```
count = 0
for i in range(10):
    if i % 2 == 0:
        continue
    count += 1
print(count)
```

- A) 5
- B) 10
- C) 4
- D) 6

5. What does the following code result in?

```
while True:
    print("Hello")
    break
```

- A) Prints "Hello" forever
- B) Prints "Hello" once
- C) Syntax error
- D) None of the above

6. How does `break` work in a loop?
- A) Stops the execution of the loop and resumes outside the loop
- B) Skips the current iteration and moves to the next
- C) Causes the loop to execute indefinitely
- D) None of the above

7. What is the result of using `continue` in a loop?
- A) Immediately exits the loop
- B) Skips the rest of the code inside the loop for the current iteration
- C) Restarts the loop from the beginning
- D) Causes an error

8. Which of the following code snippets will print numbers 1 to 5?

A) for i in range(1, 6): print(i)
B) for i in range(5): print(i + 1)
C) i = 1 while i <= 5: print(i) i += 1
D) All of the above

9. What will the following code print?

```
for i in range(3):
    for j in range(2):
        print(i, j)
```

- A) 0 0, 0 1, 1 0, 1 1, 2 0, 2 1
- B) 0 1, 1 2, 2 3
- C) 1 1, 2 2, 3 3
- D) None of the above

10. What does the following code do?

```
import random
n = random.randint(1, 10)
guess = int(input("Enter a guess: "))
while guess != n:
    guess = int(input("Try again: "))
print("Correct!")
```

- A) Asks the user to guess a number between 1 and 10 until they get it right
- B) Generates a random number and prints it
- C) Causes an infinite loop
- D) None of the above

11. Given the following code, what will be the final value of `sum`?

```
sum = 0
for i in range(5, 0, -1):
    sum += i
```

- A) 10
- B) 15
- C) 5
- D) 0

12. How many iterations will this loop perform?

```
for i in range(1, 10):
    if i % 3 == 0:
        break
```

- A) 3
- B) 2
- C) 9
- D) 0

13. What is the output of the following code?

```
for i in range(-2, -5, -1):
    print(i)
```

- A) -2, -3, -4
- B) -3, -4, -5
- C) No output
- D) Error

14. Consider the following code. How many times will "" be printed?

```
count = 5
while count < 8:
    print("")
    count -= 1
    if count == 2:
        break
```

- A) 3
- B) Infinite
- C) 2
- D) 1

15. What does the following code print?

```
x = 1
while x < 5:
    x *= 2
    if x == 4:
        continue
    print(x)
```

- A) 2, 4, 8
- B) 2, 8
- C) 4
- D) 2, 4

16. What will be the output of the following snippet?

```
nums = [1, 2, 3, 4, 5]
for n in nums:
    if n % 2 == 0:
        nums.remove(n)
print(nums)
```

- A) [1, 2, 3, 4, 5]
- B) [1, 3, 5]
- C) [2, 4]
- D) [1, 3, 4]

17. After executing the following code, what will `numbers` contain?

```
numbers = [x for x in range(10) if x % 2 == 0]
for i in range(len(numbers)):
    if i % 3 == 0:
        del numbers[i]
print(numbers)
```

- A) [0, 2, 4, 6, 8]
- B) [2, 4, 6, 8]
- C) [1, 3, 5, 7, 9]
- D) Error

18. How many elements will the following list comprehension have?

```
lst = [i for i in range(-5, 5) if i < 0]
```

- A) 5
- B) 10
- C) 0
- D) 4

19. What will the following code output?

```
i = 0
while i < 3:
    i += 2
    print("A")
    if i == 4:
        break
    else:
        continue
    print("B")
```

- A) A B A
- B) A A
- C) A A B
- D) A

20. What is the result of the following code?

```
for i in range(1, 4):
    for j in range(i):
        if j == 2:
            break
        print(i, j)
```

- A) 1 0, 2 0, 2 1, 3 0, 3 1
- B) 1 0, 2 0, 3 0, 3 1
- C) 1 0, 2 1, 3 2
- D) 2 0, 2 1, 3 0, 3 1

21. What does this code print?

```
num = 0
while num < 10:
    num += 3
    print("Tick")
    if num > 5:
        break
```

- A) Tick Tick
- B) Tick

- C) Tick Tick Tick

22. Given the following nested loop, what will be printed?

```
for i in range(3):
    for j in range(2, -1, -1):
        if i == j:
            print(f"{i} equals {j}")
            break
```

- A) 0 equals 0, 1 equals 1, 2 equals 2
- B) 2 equals 2
- C) 0 equals 0
- D) No output

23. What will be the output of the following code using `random.randint`?

```
import random
print(random.randint(1,1))
```

- A) A random number between 1 and 1
- B) 1
- C) Error
- D) None of the above

24. How many times will "Hello" be printed?

```
i = 10
while i > 0:
    i -= 3
    print("Hello")
    if i <= 3:
        break
```

- A) 3
- B) 4
- C) 2
- D) 1

25. What is the output of the following code?

```
for i in range(4):
    if i == 2:
        break
    else:
        print(i)
else:
    print("Done")
```

- A) 0 1 Done
- B) 0 1
- C) 0 1 2 Done
- D) 0 1 3

26. What will the following code output?

```
sum = 0
for i in range(5):
    if i % 2 == 0:
        continue
    sum += i
print(sum)
```

- A) 4
- B) 6
- C) 5
- D) 9

27. Considering `random.randint(0,5)` generates a random number each time it's called, what is a possible output of the following code?

```
import random
for i in range(3):
    num = random.randint(0, 5)
    print(num, end=" ")
```

- A) 0 0 0
- B) 5 5 5
- C) 1 2 3
- D) All of the above

28. What does the following code snippet output?

```
i = 2
while True:
    if i % 3 == 0:
        break
    print(i)
    i += 2
```

- A) 2 4 6
- B) 2 4
- C) 2
- D) An infinite number of 2s

29. What is the final value of `x` after this code runs?

```
x = 1
for _ in range(5):
    x *= 2
```

- A) 10
- B) 16
- C) 32
- D) 64

30. How many "X" will be printed?

```
for i in range(6):
    if i == 3:
        continue
    if i > 4:
        break
    print("X", end="")
```

- A) 4
- B) 5
- C) 6
- D) 3