**Module 4. Control Structures**

**TRUE/FALSE**

1. The if statement allows for conditional execution of code blocks.

Answer: T

1. Indentation is optional in Python for defining the scope of code blocks.

Answer: F

1. The ***elif*** statement can be used multiple times to check multiple conditions.

Answer: T

1. Python’s **if** statement syntax requires parentheses around the condition.

Answer: F

1. An ***else*** statement must follow an if statement.

Answer: F

1. Logical operators such as ***and***, ***or***, and ***not*** can be used within ***if*** conditions.

Answer: T

1. The else block executes only if the if condition is false.

Answer: T

1. In Python, == is used to assign a value to a variable.

Answer: F

1. Python uses the colon (:) to indicate the start of an indented code block.

Answer: T

1. It is not possible to nest if statements within another if statement.

Answer: F

1. The condition in an if statement must always be a Boolean expression.

Answer: T

1. Python supports the switch-case statement similar to other languages.

Answer: F

1. Python’s match-case statement is used for pattern matching.

Answer: T

1. The wildcard character in match-case is represented by an asterisk (\*).

Answer: F

1. The elif statement is an abbreviation for “else if”.

Answer: T

1. The if-elif-else structure ensures that only one block of code executes.

Answer: T

1. Using tabs and spaces interchangeably for indentation in Python will cause errors.

Answer: T

1. The else block is optional in an if-elif-else structure.

Answer: T

1. Python uses curly braces to define the scope of if-else blocks.

Answer: F

1. The print statement must be indented properly to belong to the if block.

Answer: T

1. The if statement can only check for equality using the == operator.

Answer: F

1. Python allows you to use arithmetic operations within if conditions.

Answer: T

1. The condition in an if statement can include function calls.

Answer: T

1. Python does not allow for comparison of different data types in an if condition.

Answer: F

1. You can have multiple else blocks following an if statement.

Answer: F

**MULTIPLE CHOICE**

1. What is used to define the scope of code blocks in Python?

a. Curly braces { }

b. Indentation

c. Parentheses ( )

d. Square brackets [ ]

Answer: B

1. Which of the following logical operators means "both conditions must be true"?

a. or

b. and

c. not

d. xor

Answer: B

1. What keyword is used to check an additional condition if the previous conditions are false?

a. if

b. else

c. elif

d. end

Answer: C

1. What will be printed by the following code?

*a = 10*

*b = 20*

*if a < b:*

*print("a is less than b")*

*else:*

*print("a is not less than b")*

a. a is less than b

b. a is not less than b

c. No output

d. Syntax error

Answer: A

1. How do you check if a variable x is not equal to 10 in Python?

a. x = 10

b. x == 10

c. x != 10

d. x >= 10

Answer: C

1. What will be the output of the following code?

*x = 5*

*if x > 10:*

*print("Greater than 10")*

*elif x > 0:*

*print("Greater than 0 but less than or equal to 10")*

*else:*

*print("Less than or equal to 0")*

a. Greater than 10

b. Greater than 0 but less than or equal to 10

c. Less than or equal to 0

d. No output

Answer: B

1. Which of the following is a correct elif condition to check if x is equal to 5?

a. elif x == 5:

b. elif x = 5:

c. elif x != 5:

d. elif x < 5:

Answer: A

1. What is the output of the following code?

*number = 15*

*if number % 2 == 0:*

*print("Even")*

*else:*

*print("Odd")*

a. Even

b. Odd

c. Syntax error

d. No output

Answer: B

1. How do you start an indented code block in Python?

a. Using a colon (:)

b. Using curly braces { }

c. Using parentheses ( )

d. Using a semicolon (;)

Answer: A

1. What is the purpose of the else statement in an if-elif-else structure?

a. To handle an exception

b. To execute code if none of the previous conditions are true

c. To check another condition

d. To terminate the program

Answer: B

1. Which statement will correctly determine if a number x is positive?

a. if x = 0:

b. if x > 0:

c. if x < 0:

d. if x != 0:

Answer: B

1. What will be the output of the following code?

*value = 7*

*match value:*

*case 1:*

*print("One")*

*case 2:*

*print("Two")*

*case \_:*

*print("Other")*

a. One

b. Two

c. Other

d. No output

Answer: C

1. What is the syntax to check if a variable a is greater than or equal to b?

a. if a < b:

b. if a <= b:

c. if a >= b:

d. if a > b:

Answer: C

1. What will be printed by the following code?

*grade = 85*

*if grade >= 90:*

*letter = 'A'*

*elif grade >= 80:*

*letter = 'B'*

*elif grade >= 70:*

*letter = 'C'*

*else:*

*letter = 'D'*

*print(letter)*

a. A

b. B

c. C

d. D

Answer: B

1. Which statement is used to end the execution of the current iteration in a loop?

a. break

b. continue

c. pass

d. return

Answer: B

1. What is the result of the following code?

*x = 10*

*if x % 2 == 0:*

*print("Even")*

*else:*

*print("Odd")*

a. Even

b. Odd

c. Error

d. No output

Answer: A

1. What is the output of the following code?

*a = 5*

*b = 10*

*if a == b:*

*print("Equal")*

*else:*

*print("Not equal")*

a. Equal

b. Not equal

c. No output

d. Syntax error

Answer: B

1. What is the correct syntax to use the else statement?

a. else (condition):

b. else:

c. else if:

d. if else:

Answer: B

1. What is the output of the following code?

*a = 20*

*b = 20*

*if a > b:*

*print("a is greater")*

*elif a < b:*

*print("b is greater")*

*else:*

*print("a and b are equal")*

a. a is greater

b. b is greater

c. a and b are equal

d. No output

Answer: C

1. Which of the following can be used to check multiple conditions?

a. if-elif-else

b. if-if-else

c. switch-case

d. for-while

Answer: A

1. What is the correct way to check if a number is positive or negative in Python?

a. if number == 0:

b. if number != 0:

c. if number > 0:

d. if number < 0:

Answer: C

1. How do you print "Hello" if x is equal to 5?

a. if x == 5 print("Hello")

b. if x = 5: print("Hello")

c. if x == 5: print("Hello")

d. if x = 5 print("Hello")

Answer: C

1. Which of the following is a correct match-case statement in Python 3.10?

a.

switch value:

case 1:

print("One")

b.

match value:

case 1:

print("One")

c.

case value:

match 1:

print("One")

d.

if value:

case 1:

print("One")

Answer: B

1. What is the output of the following code?

*number = 0*

*if number:*

*print("Non-zero")*

*else:*

*print("Zero")*

a. Non-zero

b. Zero

c. Error

d. No output

Answer: B

1. What does the match-case statement introduced in Python 3.10 provide?

a. Function definition

b. Looping mechanism

c. Pattern matching

d. Variable assignment

Answer: C