**Understanding input(), text formatting, operations on data in Python Multiple Choice Questions (MCQs)**

1. What does the `input()` function in Python do?

a. Prints output to the screen

b. Calculates mathematical operations

c. Takes user input as a string

d. Creates a new file

2. Which operator is used for exponentiation in Python?

a. %

b.

c. //

d. ^

3. What is the output of `10 / 3` in Python?

a. 3

b. 3.33

c. 3.0

d. "10 / 3"

4. Which of the following is NOT a valid Python data type?

a. int

b. string

c. boolean

d. perform

5. Which operator checks for equality between two values?

a. ==

b. !=

c. <=

d. >

**Fill in the Blanks**

1. The `print()` function is used to print something on the console in Python.

2. In Python, `7 // 3` results in `2`.

3. A `boolean` data type can only have the values

4. The symbol `%` is used for the modulus operation in Python.

5. To concatenate two strings in Python, the `+` operator is used.

**True or False**

1. The `input()` function always returns an integer value. False

2. Python uses the `=` symbol for both assignment and equality checking. False

3. The expression `"Python" == "python"` evaluates to True. False

4. In Python, `8 \* 3` results in `24`. True

5. The expression `5 > 3 and 5 < 10` will return False. True

**Match the Columns Column A Column B**

a. `input()` i. Division

b. `==` ii. User input

c. `!=` iii. Equality

d. `/` iv. Inequal

Ans:

a. >> ii. User input

b. >> iii. Equality

c. >> iv. Inequality

d. >> i. Division

**Theory Questions**

1. Explain the difference between `=` and `==` in Python.

Ans: = (Assignment Operator):

The = symbol in Python is an assignment operator.

It is used to assign a value to a variable.

For example, x = 5 assigns the value 5 to the variable x.

It does not test for equality; instead, it assigns the value on the right-hand side to the variable on the left-hand side.

== (Equality Operator):

The == symbol in Python is an equality operator.

It is used to compare two values to determine if they are equal.

For example, x == 5 checks if the value of the variable x is equal to 5.

It returns True if the values are equal, and False otherwise.

It does not perform assignment; it only checks for equality.

1. Describe how a Python program executes on a computer.

Following are the steps:

Lexical Analysis (Tokenization)

Parsing (Syntax Analysis)

Semantic Analysis

Intermediate Code Generation (Optional)

Execution

Runtime Environment

Output

Termination

1. What are the benefits of using functions in Python?

Code Reusability, Functions allow code to be reused multiple times within a program or across different programs. Modularity, Functions allow you to break down a program into smaller, manageable pieces of code. Abstraction ,Functions provide a level of abstraction, allowing you to focus on the functionality of each function without worrying about its implementation details.

Practical Coding Questions

1. Daily Steps Tracker: Write a Python program that asks the user to enter the number of steps they walked each day for a week. After entering seven numbers, the program should calculate and display the total number of steps walked in the week.

#adding code from my pycharm IDE

def daily\_steps\_tracker():  
 total\_steps = 0  
 for day in range(1, 8):  
 steps = int(input(f"Enter the number of steps walked on day {day}: "))  
 total\_steps += steps  
 print(f"Total steps walked in the week: {total\_steps}")  
  
  
  
daily\_steps\_tracker()

1. Grade Calculator: Create a Python program that asks for a student's marks in three subjects (out of 100). The program should calculate the average mark and print the corresponding grade based on the average (A for 90 and above, B for 80-89, C for 70-79, D for 60-69, and F for below 60).

def grade\_calculator():  
 marks = []  
 for subject in range(1, 4):  
 mark = int(input(f"Enter the mark for subject {subject} (out of 100): "))  
 marks.append(mark)  
 average\_mark = sum(marks) / len(marks)  
 if average\_mark >= 90:  
 grade = "A"  
 elif average\_mark >= 80:  
 grade = "B"  
 elif average\_mark >= 70:  
 grade = "C"  
 elif average\_mark >= 60:  
 grade = "D"  
 else:  
 grade = "F"  
 print(f"Average mark: {average\_mark}")  
 print(f"Grade: {grade}")  
  
grade\_calculator()

1. Currency Converter: Write a Python code that converts US dollars to Euros. The program should ask the user to input the amount in US dollars and use a conversion rate of 1 USD = 0.85 Euros to convert and display the amount in Euros.

def currency\_converter():  
 usd\_amount = float(input("Enter the amount in US dollars: "))  
 conversion\_rate = 0.85  
 euros\_amount = usd\_amount \* conversion\_rate  
 print(f"{usd\_amount} US dollars is equivalent to {euros\_amount} Euros.")  
  
  
currency\_converter()

1. Time Converter: Create a program in Python that asks the user to input time in seconds and then converts it into hours, minutes, and seconds. For example, 3665 seconds should be converted to 1 hour, 1 minute, and 5 seconds.

def time\_converter():  
 seconds = int(input("Enter the time in seconds: "))  
 hours = seconds // 3600  
 remaining\_seconds = seconds % 3600  
 minutes = remaining\_seconds // 60  
 remaining\_seconds = remaining\_seconds % 60  
 print(f"{seconds} seconds is equivalent to {hours} hours, {minutes} minutes, and {remaining\_seconds} seconds.")  
  
  
  
time\_converter()