Q1. \*\*Why do we call Python as a general purpose and high-level programming language?\*\*

- Python is considered a general-purpose programming language because it can be used for a wide variety of applications, including web development, scientific computing, data analysis, artificial intelligence, automation, and more.

- It is called a high-level language because it abstracts away many low-level details, making it easier to write and understand code.

Q2. \*\*Why is Python called a dynamically typed language?\*\*

- Python is dynamically typed because the type of a variable is determined at runtime. You don't need to explicitly declare the type of a variable; it is inferred from the value it holds.

Q3. \*\*Pros and Cons of Python programming language:\*\*

\*\*Pros: \*\*

- Readable and expressive syntax.

- Extensive standard library.

- Strong support for scientific computing and data analysis.

- Platform-independent.

- Large and active community.

- Versatile (used in web development, automation, scientific computing, etc.).

\*\*Cons: \*\*

- Slower execution speed compared to languages like C or C++.

- Global interpreter lock (GIL) can limit multithreading performance in some cases.

- Not as suitable for low-level system programming.

Q4. \*\*Domains where Python can be used:\*\*

- Web Development

- Data Analysis and Visualization

- Artificial Intelligence and Machine Learning

- Scientific Computing

- Automation and Scripting

- Game Development

- Network Servers

- Finance and Trading

- Education and Research

Q5. \*\*Variables and Declaration:\*\*

- Variables are used to store data in a program. They are like containers for holding values.

- In Python, you can declare a variable by simply assigning a value to it. For example: `x = 10`.

Q6. \*\*Taking Input from the User:\*\*

- You can use the `input()` function to take input from the user. For example:

```python

name = input("Enter your name: ")

```

Q7. \*\*Default datatype of input() function:\*\*

- The `input()` function always returns a string. If you want to use the input as a different type (e.g., integer or float), you need to explicitly convert it.

Q8. \*\*Type Casting:\*\*

- Type casting is the process of converting one data type into another. For example, converting a string to an integer.

Q9. \*\*Taking Multiple Inputs with a Single input() Function:\*\*

- Yes, you can take multiple inputs in a single line using `split()` and `map()` functions. For example:

```python

a, b, c = map(int, input("Enter three numbers: ").split())

```

Q10. \*\*Keywords:\*\*

- Keywords are reserved words in Python that have special meaning and cannot be used as variable names.

Q11. \*\*Using Keywords as Variables:\*\*

- No, you cannot use keywords as variable names because they are reserved for specific purposes in Python.

For the remaining questions (Q12-Q25), I'll provide code snippets:

```python

# Q12. Indentation and its use

# Indentation is the spacing at the beginning of a line that determines the grouping of statements in Python.

# It is crucial for defining blocks of code (like in loops, conditional statements, functions, etc.).

# Q13. Output in Python

print("Hello, World!")

# Q14. Operators in Python

# Arithmetic operators: +, -, \*, /, %, \*\*

# Comparison operators: ==, !=, >, <, >=, <=

# Logical operators: and, or, not

# Q15. Difference between / and // operators

# / performs floating-point division, while // performs floor division (rounds down to the nearest whole number).

# Q16. Code output

print("iNeuro" \* 4)

# Q17. Check if a number is odd or even

num = int(input("Enter a number: "))

if num % 2 == 0:

print("Even")

else:

print("Odd")

# Q18. Boolean operators

# and, or, not

# Q19. Output of given expressions

print(1 or 0) # Output: 1

print(0 and 0) # Output: 0

print(True and False and True) # Output: False

print(1 or 0 or 0) # Output: 1

# Q20. Conditional statements

# Conditional statements allow you to execute different code based on different conditions.

# Q21. 'if', 'elif', and 'else' keywords

# 'if': Checks a condition and executes a block of code if the condition is true.

# 'elif' (else if): Provides an alternate condition to check if the first condition is false.

# 'else': Executes a block of code if none of the conditions are true.

# Q22. Voting eligibility code

age = int(input("Enter your age: "))

if age >= 18:

print("I can vote")

else:

print("I can't vote")

# Q23. Sum of even numbers

numbers = [12, 75, 150, 180, 145, 525, 50]

sum\_even = sum(num for num in numbers if num % 2 == 0)

print("Sum of even numbers:", sum\_even)

# Q24. Find the greatest number

a, b, c = map(int, input("Enter three numbers: ").split())

max\_num = max(a, b, c)

print("The greatest number is:", max\_num)

# Q25. Display numbers based on conditions

numbers = [12, 75, 150, 180, 145, 525, 50]

for num in numbers:

if num > 500:

break

if num % 5 == 0:

print(num)

```

Feel free to ask if you have any further questions!