Q.1. What are keywords in python? Using the keyword library, print all the python keywords.

Keywords in Python: Keywords are reserved words that have a specific meaning and purpose in Python. They cannot be used as identifiers (variable names, function names, etc.) because they are part of the language syntax. Python provides a keyword library that allows us to access and print all the keywords.

import keyword

print(keyword.kwlist)

Q.2. What are the rules to create variables in python?

Variable names can contain letters (a-z, A-Z), digits (0-9), and underscores (\_).

Variable names must start with a letter or an underscore.

Variable names are case-sensitive (e.g., myVar, MyVar, and myvar are different variables).

Python keywords cannot be used as variable names.

Variable names should be descriptive and follow a meaningful naming convention.

Q.3. What are the standards and conventions followed for the nomenclature of variables in

python to improve code readability and maintainability?

Use lowercase letters for variable names (e.g., my\_variable).

Separate words in variable names using underscores (snake\_case) for better readability.

Avoid using single-character names unless they have specific meanings (e.g., i for loop counters).

Use meaningful names that reflect the purpose or content of the variable.

Avoid using reserved words, built-in functions, or module names as variable names.

Q.4. What will happen if a keyword is used as a variable name?

it will result in a syntax error. Python will raise an error because keywords cannot be used as identifiers.

Q.5. For what purpose def keyword is used?

The def keyword is used to define a function in Python. It allows you to create custom functions that encapsulate a set of instructions to perform a specific task.

Q.6. What is the operation of this special character ‘\’?

The backslash (\) is used as an escape character in Python. It is used to indicate that the character following it has a special meaning. For example, \n represents a newline, \t represents a tab, etc.

Q.7. Give an example of the following conditions:

(i) Homogeneous list

(ii) Heterogeneous set

(iii) Homogeneous tuple

A homogeneous list contains elements of the same data type.

A heterogeneous set contains elements of different data types.

A homogeneous tuple contains elements of the same data type. It is a tuple

Q.8. Explain the mutable and immutable data types with proper explanation & examples.

Mutable data types allow modifications after creation. Changes to a mutable object do not create a new object in memory. Examples of mutable data types: lists, dictionaries, and sets.

For example : # Example of a mutable list

my\_list = [1, 2, 3]

my\_list.append(4)

print(my\_list) # Output: [1, 2, 3, 4]

Immutable data types cannot be changed after creation. Any modification to an immutable object results in creating a new object in memory. Examples of immutable data types: integers, floats, strings, and tuples.

# Example of an immutable string

my\_string = "Hello"

new\_string = my\_string + " World"

print(new\_string) # Output: "Hello World"

Q.9. Write a code to create the given structure using only for loop.

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

rows = 5

for i in range(1, rows + 1):

for j in range(rows - i):

print(' ', end='')

for k in range(2 \* i - 1):

print('\*', end='')

print()

Q.10. Write a code to create the given structure using while loop.

|||||||||

|||||||

|||||

|||

|

rows = 5

i = rows

while i >= 1:

spaces = rows - i

while spaces > 0:

print(' ', end='')

spaces -= 1

stars = 2 \* i - 1

while stars > 0:

print('|', end='')

stars -= 1

print()

i -= 1