1. a) 15

   The code defines a recursive function func that calculates the greatest common divisor (GCD) of two numbers using the Euclidean algorithm.

   In the func function, the base case is when a becomes 0. In that case, the function returns b. Otherwise, it calls itself recursively with the arguments b % a and a. This process continues until a becomes 0, at which point the GCD is found and returned.

   When func(30, 75) is called, it calculates the GCD of 30 and 75, which is 15. Therefore, the output of print(func(30, 75)) will be 15.

2. c) List (because the filtered numbers will be stored as a list when you convert the even_numbers object to a list explicitly)

   Option a) Int is incorrect because even_numbers is not a single integer but a filtered sequence of numbers.

   Option b) Filter is correct because even_numbers is indeed an instance of the filter class.

   Option d) Tuple is incorrect because the output will not be a tuple.

3. a) Tuple

   The *args syntax in Python allows a function to accept a variable number of arguments. It collects all the positional arguments passed to the function into a tuple.

4. d) Error

   The given code will result in an error. The error is due to the use of the "+" operator to concatenate sets. In Python, the "+" operator is not used to concatenate sets. Instead, it is used to perform set union.

5. a) raise

   The raise keyword is used to raise exceptions in Python. When an error or exceptional condition occurs in a program, we can use the raise keyword to

explicitly raise an exception.

6. c) datetime

   To work with date and time in Python, we need to import the datetime module. The datetime module provides classes and functions for manipulating dates, times, and combinations of both.

7. b) 169

   43 = 4 raised to the power of 3 = 64
   (7 + 5)(1 + 1) = 12 raised to the power of 2 = 144

   Therefore, the overall expression becomes:
   64 + 144 = 208
   Hence, the output will be 169.

8. b) strftime

   The strftime function in Python is used to format a datetime object or a date object into a string representation based on a specified format. It stands for "string format time." It allows us to convert a date object into a string representation that includes the time information.

9. b) Immutable

   Tuples are immutable, which means that once created, their elements cannot be modified. Immutable objects in Python cannot be changed after they are created.

10. A) range()

    The range() function in Python generates a sequence of numbers based on the specified start, stop, and step values. It is commonly used in for loops to iterate over a specific range of values.

11. C) Lambda function

    Lambda functions in Python are anonymous functions that do not have a specific name associated with them. They are defined using the lambda keyword,

followed by the function's input arguments and a single expression.

12. C) Both A and B

The module pickle in Python is used for object serialization and deserialization. It allows you to convert Python objects into a byte stream (serialization) and vice versa (deserialization). The byte stream can be saved to a file or sent across a network, and later reconstructed to create a copy of the original object.

13. B) dump() method

Python's pickle module contains the dump() method, which is used to serialise Python objects and write them to a binary file.

14. A) load()

The load() method is a part of the pickle module in Python and is used to deserialize data from a binary file and reconstruct the original Python objects.

15. D) All of the mentioned above

The alphabet, numbers, and special symbols can all be found in text files. It is a typical file format for storing and sending text-based data. Text files can contain any printable character, including letters, numbers, punctuation, spaces, and other symbols. As a result, a text file can contain all of the previously described possibilities (alphabets, numbers, and special symbols).

16. d) both a and b

Both of these options iterate over the captains dictionary and print each ship and its corresponding captain. They will produce the desired output:

17. d) captains = {}

Curly brackets can be used in Python to define dictionary literals. We make an empty dictionary by giving the variable captains a series of empty curly braces. This line assigns an empty dictionary to the variable captains. After executing this line, the captains variable will be an empty dictionary that can be used to store key-value pairs.

18. b) captains["Enterprise"] = "Picard" captains["Voyager"] = "Janeway" captains["Defiant"] = "Sisko"

In option a), the syntax used to add key-value pairs to the dictionary is incorrect. The correct syntax to add key-value pairs to a dictionary is using square brackets [].

Option c) is not applicable to adding key-value pairs to an existing dictionary. It is used to create a new dictionary with the specified key-value pairs.

19. b) for ship, captain in captains.items(): print(f"The {ship} is captained by {captain}.")

The captain's dictionary's entries are iterated over in this code using a for loop. The key-value pairs of the dictionary are included in a series of tuples that are returned by the items() method. We can obtain the names of the ship and captain by decomposing each tuple into the variables ship and captain. The print() statement then uses f-string formatting to display the ship and captain names together with the additional context.

20. c) del captains["Discovery"]

We can remove a particular key-value pair from a dictionary by using the del keyword in combination with the name of the dictionary and the key enclosed in square brackets. In this instance, the dictionary entry for the key "Discovery" is deleted.