1. The given code implements the Euclidean algorithm to find the greatest common divisor (GCD) of two numbers `a` and `b`. The algorithm repeatedly computes the remainder of `b` divided by `a`, and swaps `a` and `b` until `a` becomes zero. At this point, the GCD is the value of `b`.

The first call to `func` checks if `a` (which is 30) is equal to zero. Since it is not, the function calls itself with the arguments `(75 % 30, 30)`, which simplifies to `(15, 30)`.

In the second call to `func`, `a` is not zero, so the function calls itself again with the arguments `(30 % 15, 15)`, which simplifies to `(0, 15)`. Now `a` is zero, so the function returns the value of `b`, which is 15.

Therefore, the output of the given code snippet is:

c) 15

2. The given code creates a tuple of numbers and then sorts them in ascending order using the `sorted()` function. It then defines a lambda function `even` that returns `True` if a number is even (i.e., divisible by 2) and `False` otherwise. It then applies the `filter()` function to the sorted numbers using the lambda function `even` as the filter criterion. This creates a filter object that contains only the even numbers from the sorted tuple.

Finally, the code prints the type of the filter object using the `type()` function.

The output of the code will be:

b) Filter

This is because `filter()` returns a filter object, which is an iterator that contains the elements from the original iterable that satisfy the filter criterion. The filter object is not a list, tuple, or integer, but it can be converted to a list or tuple using the `list()` or `tuple()` functions, respectively.

3. When using `*args` in a function parameter, it allows the function to accept an arbitrary number of arguments. The arguments are collected into a tuple.

So, the answer is:

a) Tuple

4. (d)error
5. (a)raise
6. (c)datetime
7. c) 208

8. The function that converts a date to the corresponding time in Python is `strftime`.

`strftime` stands for "string format time" and is used to convert a `datetime` object to a string representation of the date and time in a specified format.

On the other hand, `strptime` is used to parse a string representation of a date and time into a `datetime` object.

Therefore, the correct answer is b) `strftime`.

9. b)immutable

10. range()

11. C. Lambda function.
12. The correct answer is C. Both A and B.

The `pickle` module in Python is used to serialize and de-serialize Python object structures.

Serialization is the process of converting an object into a format that can be stored or transmitted, such as a byte stream or a file. De-serialization is the process of converting the serialized object back into its original form.

The `pickle` module provides two main functions: `pickle.dump()` and `pickle.load()`. The `pickle.dump()` function is used to serialize a Python object structure and save it to a file or a byte stream. The `pickle.load()` function is used to read a serialized object structure from a file or a byte stream and de-serialize it back into its original form.

13. The correct answer is B. `dump()` method.

14. a load()

15. A text file can contain all of the mentioned above: alphabets, numbers, and special symbols.

In general, a text file is a file that contains only human-readable characters encoded in some character set, such as ASCII, UTF-8, or ISO-8859. The characters can include letters, digits, punctuation marks, whitespace, and other printable characters.

In Python, we can read and write to text files using the built-in `open()` function, with the file mode set to `"r"` for reading or `"w"` for writing. For example, to open a text file called "data.txt" for reading, we can use the following code:

```
with open("data.txt", "r") as f:

    contents = f.read()

    print(contents)
```

This code opens the file "data.txt" in read mode, reads the entire contents of the file using the `read()` method, and then prints the contents to the console.

17. The correct answer is d) `captains = {}`.

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

In Python, we can iterate over the key-value pairs in a dictionary using the `items()` method. This method returns a sequence of tuples, where each tuple contains a key-value pair from the dictionary. We can then use tuple unpacking to assign the key and value to separate variables.

Option a) `for item in captains.items(): print(f"The [ship] is captained by [captain].")` is not correct because it uses square brackets instead of curly braces to format the output string, and the variables `ship` and `captain` are not defined within the loop.

Option c) `for captain, ship in captains.items(): print(f"The {ship} is captained by {captain}.")` is not correct because it swaps the variable names `captain` and `ship`, which would result in incorrect output.

Therefore, option b) `for ship, captain in captains.items(): print(f"The {ship} is captained by {captain}.")` correctly iterates over the key-value pairs in the `captains` dictionary, assigns the ship name to the variable `ship` and the captain name to the variable `captain`, and then uses string interpolation to format the output string with the correct ship and captain names.

20. c) del captains["Discovery"]

16. d) Both a and b