1.

b) 20

In the given example, `func(30, 75)` is called. The initial values of `a` and `b` are 30 and 75, respectively. Since `a` is not 0, the function calls itself with the arguments `(75 % 30, 30)`. The remainder of 75 divided by 30 is 15, so the new values of `a` and `b` become 15 and 30.

Again, `a` is not 0, so the function calls itself with the arguments `(30 % 15, 15)`. The remainder of 30 divided by 15 is 0, so `a` becomes 0. At this point, the function returns `b`, which is 15.

Therefore, the output of `print(func(30, 75))` will be 15.

2. The output will be:

b) Filter

.

In the given code snippet, `even_numbers` is assigned the result of `filter(even, sorted_numbers)`, which means `even_numbers` is an iterator object of filtered even numbers.

Therefore, the output of `print(type(even_numbers))` will be `filter`.

3. When passed into a function, the `*args` parameter collects any number of positional arguments and stores them as a tuple datatype.

a) Tuple: The `*args` parameter gathers the positional arguments and stores them as a tuple. The individual arguments passed to `*args` will be accessible as elements within the tuple.

b) List: If you want to store the arguments as a list, you can explicitly convert the tuple of `*args` to a list using the `list()` function. For example, `args_list = list(args)` would convert the `args` tuple into a list.

c) Dictionary: The `*args` parameter is not suitable for collecting arguments to be stored as a dictionary. If you want to pass key-value pairs, it's more appropriate to use the `**kwargs` parameter, which collects keyword arguments and stores them as a dictionary.

d) None: `None` is not a datatype for storing `*args` itself. However, if no arguments are passed to `*args`, it will be an empty tuple (`()`).

5. The given code snippet will result in an error.

The error occurs because the `+` operator cannot be directly used to concatenate sets in Python. The `+` operator is used for set union operation rather than concatenation.

6. The keyword used in Python to raise exceptions is:

a) raise

The `raise` keyword is used to explicitly raise an exception in Python. It is followed by the type of exception to be raised or an instance of an exception class. By using `raise`, you can generate exceptions programmatically based on certain conditions or criteria.

For example, you can raise a `ValueError` exception as follows:

```python
raise ValueError("Invalid value")
```

6. To handle date and time computations in Python, you need to import the `datetime` module.

c) datetime

The `datetime` module in Python provides classes for working with dates, times, and combinations of both. It includes various classes like `datetime`, `date`, `time`, `timedelta`, and more, which allow you to perform various operations and manipulations on dates and times.

7. The output of the code snippet will be:

b) 169

Explanation:

The code snippet performs a mathematical expression involving exponentiation (`**`) and addition (`+`).

8. The output of the code snippet will be:

b) 169

Explanation:

The code snippet performs a mathematical expression involving exponentiation (`**`) and addition (`+`).

8. The correct option is:

b) strftime

9.The python tuple is:

b) immutable

10. The correct option is:

A. range()

The `range()` function in Python is a built-in function that returns a range object, which represents a sequence of numbers. The range object can be used to iterate over a series of integer numbers using a for loop or other iterable operations.

11.

The correct option is:

C. Lambda function

A lambda function in Python is an anonymous function that does not have a name. It is also known as an anonymous function because it can be defined without using the `def` keyword and without assigning it a name.

12. The correct option is:

C. Both A and B

The module `pickle` in Python is used for serializing (pickling) and de-serializing (unpickling) Python object structures. Serialization refers to the process of converting a Python object into a byte stream, and deserialization is the reverse process of reconstructing the Python object from the byte stream.

The `pickle` module provides functions such as `pickle.dump()` and `pickle.load()` to serialize and deserialize Python objects. With these functions, you can convert complex Python data structures, including custom objects, into a byte stream that can be stored or transmitted, and then reconstruct them back into Python objects.

13. The correct option is:

B. dump() method

The `dump()` method is a method provided by the `pickle` module in Python, which is used to convert Python objects into a binary format and write them to a file.

14. The correct option is:

A. load()

The `load()` method is a method provided by the `pickle` module in Python, which is used to unpickle (deserialize) data from a binary file.

15.

The correct answer is:

D. All of the mentioned above

A text file can contain a variety of textual information, including alphabets, numbers, and special symbols. Text files are commonly used for storing and exchanging textual data, such as documents, code files, configuration files, logs, and more.

16. The correct answer is:

d) both a and b

Both options a) and b) can be used to achieve the desired output. Let's analyze them:

```
a) for ship, captain in captains.items():
   print(ship, captain)
```

This code iterates over the items of the `captains` dictionary using the `items()` method, which returns a sequence of key-value pairs. In each iteration, it assigns the key to the variable `ship` and the corresponding value to the variable `captain`. Then, it prints the values of `ship` and `captain`, which will result in the desired output.

17. The correct line of code to create an empty dictionary named "captains" is:

d) captains = {}

18.

b).

19.

b)

20.

c)