Q1.

1. Abstraction is the hiding methods that we do not need to see them, and encapsulation that an idea to hide the method inside one object for example when we make a Class for something we can access it and modify it
2. A class is a data structure built-in like class str or dict or list or any other data type has methods, everything in python is an object
3. Operator overload
4. Codes###
5. Dunder methods are methods has double underscore or we call them dunder or magic that methods for
6. Inheritance that a class inherit attribute from another class and we can add more attribute for the child class and we call the class we inherit from a parent class and the child is that class would inherit from their parents
7. Function is a function outside a class and when we call that we only call the name of the function, methods live inside a class and when we need to call that method or use it we need to use dote to access that methods, similarity both can take input and they can return something sometimes nothing
8. Namespace is like dictionary it has key and values, each key has value so the keys like object name and the value are objects global and local built in and enclosing
9. OOP has polymorphism, instances, inheritance that we can make an object or a class and make another class that would inherit the same attributes from their parents and we could say parents and child and the child can have more attribute, but functional programing does not have them ……#####
10. Accessor and getter are methods one to set things for example to set positions using that method and that method does not return anything, and getter does not set anything but it does get what we want so it returns things and there are two types for them privet and public .

Q2

a)

1. Machine code is a language that hard to code and it takes time and effort to code with that language and that was the first language like binary code 0010101010101
2. Assembly language is that easier than machine code and easier to read and it has more things human can understand when they read it like LOAD, ADD
3. High level languages it is faster than the other two languages and much easier to read and to code on it like c++, python, java so that has libraries we can import them to help us and it has classes and methods and it is more flexible
4. compiler is faster in executing, show errors in the check it needs more memory. Interpreter shows error in the running time, takes one line of code, slower in executing, it needs less memory.

c) algorithms is a sequence of steps that we follow to reach the goal. We can do that as an English word like instructions to follow and we call that pseudocode or we draw the flowchart and that has some shapes with meaning

Q3) platform is the hardware plus the software and the application run on top of them, the platform independent can be run on different operation systems without difficulty like python

Q4)

1. data structure is instruction we built or it is built in like dictionary or lists or string tuple (classes) and that has method and each method can work on one class or more like append in python can not be used on dict or tuple but some they can work like length or count methods depends mutable is a data structure that we can not use methods like append like dict list string and immutable is a data structure we can use methods like append or extend…
2. Modularisation is important because we can divide one module to many and make them into separate parts and then connect them together to make it working that would be more flexible easy to follow better than all the code in one thing.

Q5)

Class diagram