Q1

Object-oriented programming is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields, and code, in the form of procedures. A feature of objects is an object's procedures that can access and often modify the data fields of the object with which they are associated.

Q2

* It provides a clear modular structure for programs which makes it good for defining abstract datatypes in which implementation details are hidden.
* Objects can also be reused within an across applications. ...
* It makes software easier to maintain. ...
* Reuse also enables faster development.

Q3

Python method is called on an object, unlike a function. ... Since we call a method on an object, it can access the data within it. A method may alter an object's state, but Python function usually only operates on it, and then prints something or returns a value

In object-oriented programming, a class is an extensible program-code-template for creating objects, providing initial values for state and implementations of behavior.

In object-oriented programming (OOP), objects are the things you think about first in designing a program and they are also the units of code that are eventually derived from the process. ... Each object is an instance of a particular class or subclass with the class's own methods or procedures and data variables.

A Python class attribute is an attribute of the class (circular, I know), rather than an attribute of an instance of a class.

Behavior. A class's behavior determines how an instance of that class operates; for example, how it will "react" if asked to do something by another class or object or if its internal state changes. Behavior is the only way objects can do anything to themselves or have anything done to them.