1. How does Object Oriented Programming differ from Process Oriented Programming?

In Object Oriented Programming code is divided into smaller parts - objects. Data is more secure and adding new data is easy. Data is more important than functions and is in use with large and complex programs.

In Process Oriented Programming code is divided into smaller parts - functions. Data is less secure and adding new data is not easy. Functions are more important than data is in use medium-sized programs.

2. What's polymorphism in OOP?
   The literal meaning of polymorphism is the condition of occurrence in different forms. Polymorphism is a very important concept in programming. It refers to the use of a single type entity (method, operator or object) to represent different types in different scenarios. While creating class methods as Python allows different classes to have methods with the same name.

3. What's inheritance in OOP?
   Inheritance enables us to define a class that takes all the functionality from a parent class and allows us to add more.It refers to defining a new class with little or no modification to an existing class. The new class is called derived (or child) class and the one from which it inherits is called the base (or parent) class.

4. If you had to make a program that could vote for the top three funniest people in the office, how would you do that? How would you make it possible to vote on those people?
   the program will show list of all people in the office, everyone must vote on 3 persons form the list. 3 with max total votes is winning.

5. What's the software development cycle?

   Planning, defining the requirements, prototyping and design, development, testing, deployments, operations and maintenance.

6. What's the difference between agile and waterfall?
   Agile is an iterative methodology, Waterfall is a sequential methodology that can also be collaborative, but tasks are generally handled in a more linear process.

7. What is a reduced function used for?
   reduce() **r**eturns the value that is returned from the callback function on the final iteration of the array. reduce() is a central concept in functional programming, where it's not possible to mutate any value, so in order to accumulate all values in an array, one must return a new accumulator value on every iteration.

8. How does merge sort work
   Merge sort works by splitting the input list into two halves, repeating the process on those halves, and finally merging the two sorted halves together. The algorithm first moves from top to bottom, dividing the list into smaller and smaller parts until only the separate elements remain.

9. Generators - Generator functions allow you to declare a function that behaves like an iterator, i.e. it can be used in a for loop. What is the use case?
   Can be used to iterate over a large number of values, without creating a massive list.

10. Decorators - A page for useful (or potentially abusive?) decorator ideas. What is the return type of the decorator?
    A decorator is a callable (usually a function though sometimes a class) that accepts either a function or a class and returns a new function or class that wraps around the original one.